Efficient and robust partitioned solution schemes for fluid-structure interactions

by

Alfred E.J. Bogaers

A dissertation submitted in partial fulfilment
of the requirements for the degree

Doctor of Philosophy

in the

Faculty of Engineering and the Built Environment

University of Cape Town

January 2015

Abstract

Title:	Efficient and robust partitioned solution schemes for fluid-structure interactions
Author:	Alfred Edward Jules Bogaers
Supervisors:	Prof. B.D. Reddy
	Prof. S. Kok
	Prof. T. Franz
Department:	Department of Mechanical Engineering
Degree	Doctor of Philosophy
Keywords:	fluid structure interactions, partitioned solvers, black-box, Quasi-Newton methods, aritificial compressibility, radial basis function interpolation

In this thesis, the development of a strongly coupled, partitioned fluid-structure interactions (FSI) solver is outlined. Well established methods are analysed and new methods are proposed to provide robust, accurate and efficient FSI solutions. All the methods introduced and analysed are primarily geared towards the solution of incompressible, transient FSI problems, which facilitate the use of black-box sub-domain field solvers.

In the first part of the thesis, radial basis function (RBF) interpolation is introduced for interface information transfer. RBF interpolation requires no grid connectivity information, and therefore presents an elegant means by which to transfer information across a non-matching and non-conforming interface to couple finite element to finite volume based discretisation schemes. The transfer scheme is analysed, with particular emphasis on a comparison between consistent and conservative formulations. The primary aim is to demonstrate that the widely used conservative formulation is a zero-

order method. Furthermore, while the consistent formulation is not provably conservative, it yields errors well within acceptable levels and converges within the limit of mesh refinement.

A newly developed multi-vector update quasi-Newton (MVQN) method for implicit coupling of black-box partitioned solvers is proposed. The new coupling scheme, under certain conditions, can be demonstrated to provide near Newton-like convergence behaviour. The superior convergence properties and robust nature of the MVQN method are shown in comparison to other well known quasi-Newton coupling schemes, including the least squares reduced order modelling (IBQN-LS) scheme, the classical rank-1 update Broyden's method, and fixed point iterations with dynamic relaxation.

Partitioned, incompressible FSI, based on Dirichlet-Neumann domain decomposition solution schemes, cannot be applied to problems where the fluid domain is fully enclosed. A simple example often provided in the literature is that of balloon inflation with a prescribed inflow velocity. In this context, artificial compressibility (AC) will be shown to be a useful method to relax the incompressibility constraint, by including a source term within the fluid continuity equation. The attractiveness of AC stems from the fact that this source term can readily be added to almost any fluid field solver, including most commercial solvers. AC/FSI is however limited in the range of problems it can effectively be applied to. To this end, the combination of the newly developed MVQN method with AC/FSI is proposed. In so doing, the AC modified fluid field solver can continue to be treated as a black-box solver, while the overall robustness and performance are significantly improved.

The study concludes with a demonstration of the modularity offered by partitioned FSI solvers. The analysis of the coupled environment is extended to include steady state FSI, FSI with free surfaces and an FSI problem with solid-body contact.

Acknowledgements

First and foremost, I would like express my most sincere gratitude to my supervisors, Prof. Thomas Franz, Prof. Schalk Kok and Prof. B. Daya Reddy, for their guidance and support throughout this study. It has been an enormous pleasure and privilege to have worked with you over these past few years.

I would like to thank Modelling and Digital Sciences, (CSIR, South Africa), for their financial support without whom this study would not have been possible. To Dr. Onno Ubbink, thank you for providing and fostering such a pleasant working environment.

A special thanks to my parents for always inspiring the best in me.

Lastly, I want to thank Antoinique, for her endless support and tireless patience throughout this period. I am truly grateful.

Contents

Abstract iii
Acknowledgments v
Nomenclature xi
List of Figures xv
List of Tables xix

1 Introduction **1**

2 Brief Description of the Structural and Fluid Sub-problems **9**
2.1 Introduction 9
2.2 Solid Sub-domain 10
2.3 Fluid Sub-domain 13
2.4 Conclusion 18

3 FSI Preliminaries **19**
3.1 FSI Multi-field Problem Description 19
3.2 Monolithic vs. Partitioned Solution Schemes 20
3.3 Partitioned FSI Solver 23
3.3.1 Fluid and Solid Solvers 23
3.3.2 Dirichlet-Neumann Interface Conditions 24
3.3.3 Implicit Iterative Coupling 25
3.3.4 Information Transfer and Inter-solver Communication 27
3.3.5 Mesh Movement 28
3.4 Conclusion 29

4 Interface Information Transfer **31**
4.1 Background . . . 31
4.2 Conservative and Consistent Coupling Approaches . . . 33
4.2.1 Interface Conditions . . . 33
4.2.2 Conservative Information Transfer . . . 34
4.2.3 Consistent Information Transfer . . . 36
4.3 Brief Overview of Interface Information Transfer Methods . . . 37
4.4 Multivariate Interpolation . . . 41
4.4.1 Radial Basis Functions . . . 41
4.4.2 Basis Functions . . . 44
4.4.3 Control Points . . . 45
4.5 Numerical Analysis . . . 47
4.5.1 Patch Test . . . 47
4.5.2 Analytical Test Function . . . 49
4.5.2.1 Displacement Transfer . . . 50
4.5.2.2 Pressure Transfer Error . . . 51
4.5.2.3 Energy Conservation . . . 53
4.5.2.4 RBF matrices condition number . . . 54
4.5.3 Mesh Mismatch . . . 54
4.5.4 1D Test Problem . . . 57
4.6 Conclusion . . . 60

5 Quasi-Newton Methods for Implicit Black-Box FSI Coupling **61**
5.1 Introduction . . . 61
5.2 Coupled FSI Problem . . . 62
5.3 Block-Newton Equations . . . 63
5.4 Quasi-Newton Coupling Algorithms for FSI . . . 65
5.4.1 Multi-vector update method . . . 65
5.4.2 Broyden's Method . . . 69
5.4.3 Least squares, reduced order modelling quasi-Newton method . . . 70
5.4.4 Aitken's dynamic relaxation . . . 73
5.5 Test Problems . . . 74
5.5.1 Flow Induced Oscillating Flexible Beam . . . 75

5.5.2 1D dynamic piston-channel problem 78
5.5.3 Flexible Tube, $\rho_s = 1.2\text{g/cm}^3$ and $\rho_f = 1.0\text{g/cm}^3$ 83
5.5.4 Flexible Tube, $\rho_s = 0.12\text{g/cm}^3$ and $\rho_f = 1.0\text{g/cm}^3$ 84
5.6 Conclusion 86

6 Proposed POD Augmentation of IBQN-LS **89**
6.1 Introduction 89
6.2 POD and the Proposed IBQN-LS Augmentation 90
6.3 Analysis via Flexible Tube Problem 92
6.4 Critical analysis of the POD augmentation 94

7 Fully Enclosed Partitioned FSI **97**
7.1 Introduction 97
7.2 Artificial Compressibility (AC) 98
7.3 IAC/FSI + quasi-Newton 101
7.4 Test Problems 105
7.4.1 2D Flexible Beam 105
7.4.2 Balloon Inflation 108
7.4.3 Damped Structural Instability Fully Enclosed Test Problem . . 111
7.5 Conclusion 117

8 Partitioned Coupling of Steady State, Free Surfaces, and Solid Body Contact **119**
8.1 Introduction 119
8.2 Steady State 119
8.2.1 Steady State Flow Past Elastic Structure 120
8.3 FSI with Free Surfaces 122
8.3.1 Dam Break with Elastic Obstruction 123
8.3.2 Submerged Square 126
8.4 Solid Contact 129
8.4.1 Problem Description 129
8.4.2 Contact Formulation via Springs 131
8.4.3 Including Contact Forces in QN Approximations 132
8.4.4 Flexible Beam with Solid Body Contact 133

8.5 Conclusion 135

9 Conclusion **139**

A MVQN Algorithm **143**

B IAC/FSI + MVQN Algorithm **145**

Nomenclature

Lowercase letters

$\boldsymbol{a}$	eigenvectors
$\boldsymbol{b}$	body force
d_p	contact penetration distance
$\boldsymbol{d}$	displacement vector
$\boldsymbol{f}$	force vector
$\boldsymbol{g}$	arbitrary field
h	notional element height
k	iteration counter
m	number of interface degrees of freedom
$\boldsymbol{n}$	unit normal outward pointing vector
p	pressure
q	number of retained histories for IBQN-LS method
r	support radius
$\boldsymbol{r}$	residual vector
$\boldsymbol{s}$	interpolation function
t	time
$\boldsymbol{t}$	traction field
$\boldsymbol{u}$	fluid velocity field
$\boldsymbol{u}_m$	ALE-coordinate system velocity
$\boldsymbol{w}$	weighting field
$\boldsymbol{x}$	spatial coordinates

Uppercase letters

$\Delta \boldsymbol{D}$	displacement observation matrix
$\boldsymbol{D}$	fluid rate of deformation tensor
E	Young's modulus
$\boldsymbol{E}$	Green-Lagrange strain
$\boldsymbol{F}$	material deformation gradient
	discrete force vector
$\boldsymbol{F}(\cdot)$	fluid solver mapping operator
$\Delta \boldsymbol{F}$	force/traction observation matrix
H_{AB}	interface information transfer interpolation matrix from interface mesh A to B
$\boldsymbol{I}$	identity matrix
$\boldsymbol{J}$	Jacobian matrix
K	contact spring stiffness
$\boldsymbol{K}$	stiffness matrix
L	length
$\boldsymbol{M}$	mass matrix
N	local polynomial interpolation shape function
$\boldsymbol{P}$	Piola-Kirchhoff stress tensor
$\boldsymbol{R}$	residual matrix
$\boldsymbol{\mathcal{R}}$	autocorrelation matrix
$\boldsymbol{S}$	second Piola-Kirchoff stress tensor
$\boldsymbol{S}(\cdot)$	structural solver mapping operator
$\boldsymbol{T}$	discrete traction vector
$\boldsymbol{U}$	discrete velocity vector
V	volume
$\boldsymbol{X}$	material point in reference position

Greek letters

α	generalised-α coefficients
	RBF interpolation coefficients
$\boldsymbol{\alpha}$	vector of expansion coefficients
β	RBF coefficients for polynomial term
	AC coefficient

Operators

$\|\|\cdot\|\|$	minimum norm

Subscripts

f	fluid domain
k	iteration counter
n	time step
s	structural domain
D	Dirichlet
F	with regards to fluid operator $\boldsymbol{F}$
N	Neumann
S	with regards to structural operator $\boldsymbol{S}$

Superscripts

0	starting/initial vector/matrix
-1	matrix inverse
f	fluid domain
n	time step
s	structural domain
T	transpose

Abbreviations

1D	one dimensional
2D	two dimensional
3D	three dimensional
AC	artificial compressibility
DN	Dirichlet-Neumann
DOF	degrees of freedom
FEM	finite element method
FSI	fluid-structure interactions
FVM	finite volume method

IAC	interface artificial compressibility
IBQN-LS	"interface block quasi-Newton with approximation of interface Jacobians via least squares reduced order model"
MVQN	"multi-vector iteratively updated quasi-Newton"
MQ	multi-quadratic
ND	Neumann-Dirichlet
NN	nearest neighbour
POD	proper orthogonal decomposition
RBF	radial basis function
SVD	singular value decomposition
TPS	thin plate spline
WR	weighted residual
QN	quasi-Newton

List of Figures

3.1 Notional FSI problem, depicting the fluid and solid domains, sharing a common interface Γ. 20
3.2 Summary of main coupling approaches, reproduced from [68] 22
3.3 Flow charts illustrating (a) weak (or explicit) and (b) strong (or implicit) coupling of partitioned sub-field solvers. 23
3.4 Illustration of the staggered in time implicit, partitioned solution scheme. 27

4.1 Illustrative example of a non-matching mesh along a curved interface. . 32
4.2 One dimensional representation of (a) the node-projection scheme and (b) the quadrature-projection scheme. 38
4.3 Simple 1D example of an overlay mesh. 40
4.4 Constant pressure problem description. 47
4.5 Constant pressure patch test, for (a) conservative information transfer and (b) consistent information transfer. 48
4.6 Displacement field for constant pressure patch test, for (a) conservative information transfer and (b) consistent information transfer. 48
4.7 Illustrative example of the typical surface mesh mismatch using triangular fluid and four-noded solid surface elements (red: fluid mesh, black: solid mesh). 50
4.8 Displacement error for the simultaneous refinement of the fluid and solid meshes, comparing (a) different basis functions and illustrating (b) the effect of support radius for the compactly supported C^2 basis function. 52
4.9 Comparison of pressure error for the simultaneous refinement of fluid and solid interface discretisation. Solid lines (—) represent consistent transfer and (- · -) dashed lines represent conservative transfer. 53

4.10 Difference in work done across the interface. Only consistent transfer is shown, as the conservative transfer is exactly matching. 54
4.11 Comparison of the condition number as a function of number of grid points for (a) all the basis functions and the (b) C^2 basis function for different support radiuses. 55
4.12 Illustrative example of the mesh mismatch using four noded surface elements to analyse allowable mesh mismatch (red: fluid mesh, black: solid mesh). 56
4.13 Plot of L^2 relative pressure error as a function of the number of fluid to solid element ratio. 57
4.14 Relative L^2 norm displacement error as a function of the fluid to solid element ratios. The errors are compared with the interpolation error of the linear FVM to interpolate nodal velocities from nodes to face centres. (a) Linear FEM surface elements and (b) quadratic FEM surface elements. 58
4.15 1D piston-channel problem setup. 59
4.16 Plots of (a) displacement and (b) velocity for the pseudo-1D piston channel problem, comparing the conservative and consistent RBF interpolation using both linear and quadratic structural elements. 60

5.1 Flexible tail behind rigid square test problem set-up. 76
5.2 Domain discretisation for the oscillating flexible tail test problem. Total of 3104 linear, triangular fluid elements with 20 quadratic, full integration solid elements. 77
5.3 Beam tip displacement for the flexible tail problem shown for 3 different time step sizes. 77
5.4 Typical MVQN convergence rates for flexible tail problem at different times for $\Delta t = 0.02$s, $\epsilon = 10^{-9}$. 79
5.5 1D piston-channel problem set-up. 79
5.6 Domain deformation at simulation time (a) 0s, (b) 8.85s and (c) 10.0s. 80
5.7 Typical MVQN convergence rates for 1D piston-channel problem at different times, $\Delta t = 0.02$s, $\epsilon = 10^{-9}$. 82
5.8 Pressure pulse propagation at 0.003 and 0.005 seconds. Wall displacement amplified 10 times. 85

5.9 Typical MVQN convergence rates for the flexible tube problem with $\rho_S = 1.2\text{g/cm}^3$ and $\rho_f = 1.0\text{g/cm}^3$ for $\Delta t = 10^{-4}$s and $\epsilon = 10^{-9}$ at different times. 86
5.10 Typical MVQN convergence rates for the flexible tube problem with $\rho_S = 0.12\text{g/cm}^3$ and $\rho_f = 1.0\text{g/cm}^3$ for $\Delta t = 10^{-4}$s and $\epsilon = 10^{-9}$ at different times. 87

6.1 (a) Comparison of the number of solver iterations, included here is the coupling performance of Aitken's method for comparison, with (b) the associated interface matrix, $[\boldsymbol{J}_S\boldsymbol{J}_F - \boldsymbol{I}]$, condition number. 93
6.2 Displacement convergence rates for the IBQN-POD coupling scheme at different time steps. Notice the comparatively rapid "learning" with increasing retention of additional time observations. 93
6.3 A comparison of the total number of discrete differencing vectors accumulated over the course of the simulation to the number of retained solid and fluid space basis modes in the construction of $\boldsymbol{J}_S$ and $\boldsymbol{J}_F$. 94

7.1 2D Flexible beam problem description 105
7.2 (a) Beam tip displacement of point A for the flexible beam problem shown for different time steps, along with (b) a snapshot of the beam deformation with pressure contours shown here for time step 12.58s. . . 106
7.3 Fully enclosed balloon inflation test problem setup. 108
7.4 Balloon inflation results for solid test 1 showing (a) $\boldsymbol{u}_x$ contour plot at 10 seconds and (b) pressure level at point A. 110
7.5 Boundary flux error made in each of the time steps for the balloon inflation test problem. The cumulative error across the entire simulation is 2.73×10^{-6} with a maximum error of 1.83×10^{-7}. 110
7.6 Performance plots illustrating the number of coupling iterations required for each of the time steps and the typical convergence behaviour for each of the methods (shown for time step 150) for the balloon inflation test problem for (a) Test 1 ($E = 7 \times 10^5\text{kN/m}^2$, $\rho_s = 1000\text{kg/m}^3$, $\rho_f = 1.1\text{kg/m}^3$) and (b) Test 2 ($E = 7 \times 10^5\text{kN/m}^2$, $\rho_s = 1.1\text{kg/m}^3$, $\rho_f = 1.1\text{kg/m}^3$). 112

7.7 Effect of $\boldsymbol{\beta}$ on the number of coupling iterations, shown for three different time steps. On the left are the results for test 1 with test 2 results on the right. $\boldsymbol{\beta}$ for the exhaustive search is computed as a constant field value. Notice the fine bandwidth of acceptable values of $\boldsymbol{\beta}$ for the IAC/FSI implementation and the smoothing effect of the additional QN terms. 113
7.8 Problem description for the structural instability fully enclosed domain problem. 114
7.9 Velocity fields with structural displacements shown at different time steps for the curved domain fully enclosed test problem. 115
7.10 Plot of the pressure level at point A, for the curved domain fully enclosed problem. 115
7.11 Total flux error made within each of the time steps for the curved domain fully enclosed problem. 116
7.12 Comparative performance plots for the curved domain, fully enclosed test problem showing the number of coupling iterations within each time step and the typical convergence behaviour (shown here for time step 150).116

8.1 Steady state domain description. 121
8.2 Steady state simulation results of a flexible membrane for inlet velocities of $U_{\text{in}} = 0.1\text{m/s}$ (left) and $U_{\text{in}} = 1.0\text{m/s}$ (right). 121
8.3 Convergence rates for steady state simulation for inlet velocities (a) $U_{\text{in}} = 0.1\text{m/s}$ and (b) $U_{\text{in}} = 1.0\text{m/s}$. 122
8.4 Dam break with elastic structure problem description. 124
8.5 Wave interaction with elastic wall at different time steps. 125
8.6 Comparative plots of the (a) number of coupling iterations required in each time step and (b) the typical convergence behaviour shown here for time step 130 (directly following the wave striking the elastic structure). 126
8.7 Submerged square test problem description. 127
8.8 Square displacement shown at different time steps. 128
8.9 Square y-displacement for $\rho_f = 1000\text{km/m}^3$ with $\rho_s = 10\text{kg/m}^3$ and $\rho_s = 1000\text{kg/m}^3$. 129
8.10 Plot of the number of required iterations for the rising square problem for $\Delta t = 0.01\text{s}$ and $\rho_s = 10\text{kg/m}^3$. 130

8.11 Simple depiction of slave nodes to master surface contact. 132
8.12 Flexible beam with solid body contact problem description. 134
8.13 Contact penetration difference between hard contact, $K = 1 \times 10^4$ (left), and soft contact, $K = 2 \times 10^3$ (right). 135
8.14 Pressure and valve displacement plots at different time steps, for $\Delta t =$ 0.0001s and $K = 10^4$. 136
8.15 Typical convergence behaviour for valve with contact problem, shown here for time step 0.11s for (a) soft and (b) hard contact. 137

List of Tables

4.1 Comparison of the sum of the transferred forces for the constant pressure patch test. 49

5.1 Comparison of the number of iterations and relative computational time for the flexible tail problem, for different time steps sizes and convergence criterion ϵ. The notation IBQN-LS(q) indicates information from q preceding time steps are retained. If a scheme is non-convergent, the diverging time step as a ratio of the total required time steps is indicated in the CPU column. 78

5.2 Comparison of the number of iterations and relative computational time for the 1D piston-channel problem, for different time steps and convergence criterion ϵ. The notation IBQN-LS(q) indicates information from q preceding time steps are retained. If a scheme is non-convergent, the diverging time step as a ratio of the total required time steps is indicated in the CPU column. 82

5.3 Comparison of the number of iterations and relative computational time for pressure propagation through flexible tube where $\rho_s = 1.2\text{g/cm}^3$ and $\rho_f = 1.0\text{g/cm}^3$, for different convergence criterion ϵ. The notation IBQN-LS(q) indicates information from q preceding time steps are retained. If a scheme is non-convergent, the diverging time step as a ratio of the total required time steps is indicated in the CPU column. 85

5.4 Comparison of the number of iterations and relative computational time for pressure propagation through flexible tube where $\rho_s = 0.12\text{g/cm}^3$ and $\rho_f = 1.0\text{g/cm}^3$, for different convergence criterion ϵ. The notation IBQN-LS(q) indicates information from q preceding time steps are retained. If a scheme is non-convergent, the diverging time step as a ratio of the total required time steps is indicated in the CPU column. 87

7.1 Comparison of the minimum, maximum and mean number of iterations required for a convergence tolerance of $\epsilon = 10^{-8}$ for the flexible beam benchmark problem. (-) indicates non-convergence and the beam tip displacement at point A (u_y) is provided in the form of mean±amplitude[frequency] and compared to the reference results of Turek *et al.* [105]. 107

7.2 Comparison of the number of iterations and relative computational time for the balloon inflation test problem. 111

7.3 Comparison of the number of iterations and relative computational time for the damped structural instability problem. 115

8.1 Comparison of the number of iterations required and comparative CPU time for the dam break benchmark problem. 126

8.2 Comparison of the required number of iterations for the rising square test problem for $\rho_s = 10\text{kg/m}^3$. 129

8.3 Comparison of the number of iterations required for the valve with contact problem. 137

Chapter 1

Introduction

With an ever-increasing availability and size of computational resources at the disposal of researchers, multi-physics simulations are becoming an ever-present reality. Fluid-structure interactions (FSI), a subclass of multi-physics problems, have in particular received much attention. FSI can broadly be defined as the multi-field simulation or modelling of the interaction between fluid flow and deformable structures. The large majority of these problems are two-way coupled, where fluid flow induces deformations of the structure, which in turn influences the fluid domain velocities and internal stress states. FSI finds application in a wide range of important problems within the fields of engineering and life sciences, and to mention but a few, include flutter prediction in aero-elasticity [38, 89], parachute dynamics [97, 99] and blood flow through the vascular system [103, 117, 119].

The field of FSI is slowly moving towards a state of maturity. Many of the properties and complexities involved are in general well understood, with clear emerging trends. Essentially, there are two main approaches to solving the FSI problem, namely, the monolithic approach or the partitioned approach. The monolithic approach solves all equations, including those related to the interface and computational sub-domains, in a single unified solver, typically involving some variant of Newton's method [9, 55]. Alternatively, a partitioned solver utilises separate field solvers for the fluid and solid domains, with these separate domains then coupled along the interface. Partitioned schemes can either be solved explicitly or allow for sub-iterations (or implicit coupling) in order to satisfy the governing equations along the interface.

One particular subclass of FSI problems, which has been the focus of large bodies

of research, is the simulation of incompressible flows. Incompressible FSI present a number of numerical challenges, particularly when the solid and fluid densities are of equal orders of magnitude, or the geometrical aspect ratio of the problem is large (for example the simulation of blood flow through arteries). In these situations, the numerical coupling between the solid and fluid domains are non-trivial, leading to what has been referred to as the "added-mass" effect [15, 42].

One of the emerging trends to obtain a stable and robust solution procedure for these "strongly" coupled problems, has been that monolithic solvers are advised (if not required) [25, 50, 73]. While there are merits to this argument, it does remain difficult to justify the enormous initial investment required in developing a monolithic solver. This is especially relevant considering the large availability of optimised fluid and solid solvers, which monolithic solution procedures completely preclude. Segregated solvers further allow for the solution of smaller systems as opposed to one large unified set of equations, which unless properly preconditioned can be very expensive [50]. Lastly, partitioned schemes allow for each of the sub-domain fields to be solved using field specific discretisation and solution schemes, which is an important property as the length scales often differ significantly.

The design and development of partitioned FSI solvers present three primary difficulties. The first relates to efficiency and robustness of segregated solution schemes, especially when the strength of coupling increases. To guarantee that the interface conditions are satisfied, iterating between the two sub-domain solvers is required. However, unless a suitable coupling scheme is introduced, the number of required iterations can quickly escalate, if convergence is obtained at all.

Several useful and noteworthy contributions have been made with regards to stabilising partitioned incompressible FSI solvers. These include amongst others artificial compressibility [28, 60], Robin transmission conditions [5], and computing the exact interface Jacobians or sensitivities [32, 39]. Computation of the exact system sensitivities of the cross coupling terms (between the fluid and solid domains) has been demonstrated to be particularly useful, with convergence properties directly comparable to monolithic methods [73]. While these contributions represent distinct steps in allowing for the re-use of existing solvers, they are limited in the sense that they require access to the source code of one or both of the domain solvers.

For true modularity and freedom in solver choice, coupling methods that allow

for "black-box" solvers are required. Black-box schemes are designed such that no access to the source code is needed, and typically only requires access to the input and output space of the sub-domain solvers. In so doing, virtually any solver can be used. The general understanding however is that black-box coupling algorithms are not widely applicable, and are often limited in the complexity of problems they can "safely" be applied to. One of the earliest proposed methods (and one still frequently used today) is that of fixed point iterations with dynamic relaxation [112]. While leading to many early promising results, these schemes have proved insufficient for the class of problems often encountered in haemodynamics. A host of methods to improve on this performance emerged, ranging from problem-specific reduced order models [44], space mapping methods [91], using higher order polynomials to estimate or predict interface positions, and approximate quasi-Newton (QN) methods. The QN methods, whereby the interface Jacobians were approximated rather than explicitly computed, have especially shown promise towards providing a robust and efficient black-box alternative.

Most notable of these is the use of finite-differencing (FD) to approximate the matrix-vector product for Newton-Krylov solution methods [77]. The FD based methods proved capable of producing near second-order convergence rates but showed sensitivity to the choice of step sizes. They are also, in comparison to monolithic schemes, very expensive, as they require several additional fluid and solid solver calls to construct the differencing approximations [73]. The current "state of the art" black box coupling method is the "interface block quasi-Newton method for approximation of the Jacobians using Least-Squares" (IBQN-LS and similar IQN-LS) of Vierendeels *et al.* [110] and Degroote *et al.* [27]. These schemes construct approximate Jacobians using observations of the interface results. They have been demonstrated to be capable of solving a wide variety of strongly coupled FSI problems, and perform favourably when compared to monolithic schemes [25].

While the IBQN-LS and IQN-LS methods have shown great promise, their performance is directly linked to a problem-specific choice of how the number of time histories, for the construction of these Jacobians, should be retained. If properly chosen, these QN methods very nearly provide Newton-like behaviour. There is however no *a priori* way to choose this problem-specific heuristic, which if improperly chosen can lead to divergence or severely reduced convergence behaviour. To better explain this beha-

viour, we aim to illustrate that the least-squares reduced order modelling method can in fact be re-derived as a minimum norm solution of the set of secant equations. This sheds some light on the performance behaviour of the method, and directly links the method to other more established QN methods. We then further aim to introduce a new "multi-vector iteratively updated quasi-Newton method" (MVQN). Not only is this new scheme heuristic independent, but it also provides superior convergence behaviour.

This leads us to the second major problem of partitioned solution schemes. The majority of partitioned schemes, including those used in this study, are based on Dirichlet-Neumann (DN) domain decomposition. DN schemes are classified by the fluid domain interface inheriting a displacement/velocity interface condition where interface tractions (Neumann conditions) are imposed along the solid domain interface. The limitation of DN based algorithms arises when considering fully enclosed, incompressible fluid domains, coined by Küttler *et al.* [72] as the "incompressibility dilemma". An incompressible fluid domain with a prescribed inlet flow and fully enclosed by a Dirichlet type interface manifests itself as an ill-posed problem. Common examples of the incompressibility dilemma include balloon inflation problems. Numerical difficulties however are also present when considering quasi-enclosed problems, for example flow through a collapsible tube or the simulation of opening and closing heart valves.

Several potential solutions to the incompressibility dilemma have been proposed. Küttler *et al.* [72] suggest reformulating the solid equations as a volume constraint problem. Robin-Robin (or Robin-Neumann) boundary conditions have been mentioned (though not yet demonstrated) as a plausible solution [5]. Degroote *et al.* [28], and Råback *et al.* [85] have proposed modifying the continuity equation to include an artificial compressibility term. While these methods are certainly reasonable solutions, they require modifications to the source code. To the best of the author's knowledge, reversing the direction of information transfer (i.e. Neumann-Dirichlet boundary conditions) remains as the only viable means by which black-box partitioned solutions schemes can allow for fully enclosed incompressible FSI problems. However, for stiff structures, the sensitivity of interface forces with respect to displacements (and vice versa) becomes too large for standard black-box coupling schemes [72].

Artificial compressibility (AC) is however an interesting option in the context of black-box solvers. The source code modifications for the implementation of AC are minimal, and require the addition of a scalar source term to the continuity equation.

Most commercial solvers, including for example Fluent [1], allow for the addition of source terms via user defined functions; in so doing, these provide a means of introducing the additional AC without direct access to the source code. Similarly, if the source code is available, the AC term can often be included without the need for the user becoming involved with the internal solver specific solution procedures. AC as a method therefore still resides within the "spirit" of black-box coupling, where almost all stand-alone fluid solvers remain available for use.

When considered in the context of FSI, artificial compressibility (AC/FSI) does have some limitations. The additional source term can be based on a linearised approximation of the solid domain's volume change in response to a change in pressure [24, 28, 60]. AC/FSI is therefore well suited to problems where this relationship bears relevance to the nature of the FSI problem; see for example [60], where AC/FSI was applied to solve for blood flow through a bifurcating arterial section. There are however many problems where this relationship is entirely unsuitable. One such example, as mentioned in [24], is the simulation of flow around a flexible beam. A small localised pressure perturbation at the beam tip will result in the entire length of the beam deforming (as opposed to a small localised tip displacement as would be suggested by the linearised pressure-volume approximation). Furthermore, while the proposed pressure-volume approximation often results in the near optimal computation of an appropriate coefficient choice, it is based on a linearised approximation. As such, AC as a method is often incapable of accurately accounting for many of the non-linearities present within an FSI system, especially when there are large inertial forces present in the structural domain sub-problem.

To this end, we will propose the possibility of combining AC with quasi-Newton (QN) methods. The AC/FSI modification would allow for problems with fully/quasi-enclosed domains to be solved. The additional quasi-Newton sensitivities would then serve to further stabilise the method. It does so by continuing to treat the solid domain solver as a complete black-box, requiring only interface displacement information as a function of imposed interface tractions. Similarly, it treats the AC modified fluid equations as a black-box.

The last major difficulty when considering partitioned solvers is the transfer of information between the two sub-domains. One of the merits of partitioned solvers is the allowance of independent discretisation for each of the sub-domains. In addition, more

for historical reasons, these domains are solved using different numerical techniques. Most available, commercial and in-house solvers utilise the finite volume method (FVM) for solving fluid flow, and the finite element method (FEM) for solving the structural domain problem. The accuracy of the overall FSI scheme is dependent on accurate interface information transfer, which is non-trivial for non-conforming, non-matching interfaces.

Early examples of non-matching interface transfer for FSI include the node-projection scheme of Farhat *et al.* [37] and the quadrature-projection scheme of Cebral *et al.* [16]; both however have been found to be zero-order methods. Other well-adopted methods include common-refinement based data transfer [59, 63] and mortar-element methods [22]. While these two methods present favourable transfer properties, they are best suited for FEM-FEM coupling. An interesting alternative, used in this thesis, is the use of multivariate, and (in particular) radial basis function (RBF) interpolation. RBF requires no grid-connectivity information, making it an ideal candidate for FVM-FEM information transfer across non-matching interfaces.

Transferring information across non-matching meshes requires a compromise between being either conservative or consistent. By consistent interface transfer, we adopt the definition from [22], where the transfer scheme can exactly reproduce a constant stress state, and hence pass a patch test. This is a slightly modified definition from [37], where information is interpolated and integrated along a surface using rules consistent with the corresponding solver. Most researchers however argue that interface transfer should be strictly conservative (and is often formulated to satisfy the principle of virtual work) [7, 59, 58, 74, 84]. In Chapter 4 we aim to demonstrate that the conservative interface information transfer in general also leads to a zero-order method with respect to the spatial distribution of the transferred interface stress state. Similarly, while the consistent approach is not strictly conservative, the overall errors introduced are of an acceptable order.

The outline of the remainder of the thesis is as follows. In Chapter 2, we briefly present (for completeness), the fluid and solid sub-problems. We pose both the strong and weak form of the governing equations, including a few notes on the chosen solution methods. Chapter 3 briefly outlines many of the design philosophies and necessary building blocks to construct a partitioned FSI solver. In Chapter 4, we investigate interface information transfer using radial basis function interpolation. We formulate

the interface transfer problem using both the conservative and consistent approaches. We then briefly summarise the various popular interface transfer methods available in literature prior to introducing RBF in the context of interface information transfer. A comparison of the conservative and consistent formulations is performed for a patch test and a series of analytical functions.

Chapters 5, 6 and 7 deal with improving the convergence of partitioned solution schemes via QN methods. In Chapter 5 we pose the FSI problem as a linear set of block-Newton equations of the interface problem. We introduce the newly developed quasi-Newton method alongside Broyden's method, the interface block-Newton method (IBQN-LS) and Aitken's dynamic relaxation. The performance of the four interface acceleration methods is compared and analysed on a host of strongly coupled problems. In Chapter 6, we present a simple, proper orthogonal decomposition augmentation of the IBQN-LS coupling method, and illustrate how this in turn reduces some of the history dependent performance. In Chapter 7, we address the incompressibility dilemma presented by fully-enclosed partitioned FSI. We demonstrate how the incompressibility dilemma can be resolved using artificial compressibility. We then further outline the combination with quasi-Newton methods to improve on the overall coupling performance and the generic applicability of AC/FSI as a method.

In Chapter 8, we deviate from the standard class of transient incompressible FSI problems. The chapter can largely be seen as a defence of the modularity offered by partitioned solution schemes. We demonstrate how the methods developed in the preceding chapters can be used to solve steady-state problems, FSI problems with free surfaces, and a mitral valve-like problem with solid-body contact. In Chapter 9 we present the concluding remarks.

Portions of the work presented in this thesis have been published, and are available in [12, 13].

Chapter 2

Brief Description of the Structural and Fluid Sub-problems

2.1 Introduction

The aim of this chapter is briefly to introduce the fluid and structural sub-problems. The methods introduced and later analysed in this thesis are geared towards black-box field solvers and are unbiased towards any particular formulation or choice of solvers.

The fluid and solid sub-problems both belong to the general field of continuum mechanics. While they share several commonalities, the two sub-problems are naturally defined using different reference frames: a Lagrangian formulation is generally preferred for the solid equations and a Eulerian formulation for the fluid domain. The finite element method (FEM) [120] and finite volume method (FVM) [40], both subsets of the more general class of weighted residual methods, are two popular numerical methods to approximate the solution to a set of partial differential equations. Both the FEM and the FVM are well established numerical methods and find widespread application in both fluid flow problems and structural elastodynamics.

In this thesis, Calculix (introduced in Section 3.3.1) will be used for the structural domain, and OpenFOAM (Section 3.3.1) for the fluid domain. As such, we adopt the FEM for the structural domain and the FVM for fluid domain.

2.2 Solid Sub-domain

In this thesis, FSI problems with large structural deformations are analysed. We consider a homogeneous isotropic elastic solid formulation for large non-linear deformations. Given a structural domain, Ω_s, with displacements $\boldsymbol{d}$ imposed along the Dirichlet boundary $\Omega_{s,D}$, traction forces $\boldsymbol{t}$ imposed along the Neumann boundary $\Omega_{s,N}$ and a body load $\boldsymbol{b}$ imposed on the whole domain Ω_s, the elastodynamics governing equation satisfying the balance of linear momentum (also known as the Cauchy equation) [54] is given by

$$\nabla \cdot \boldsymbol{\sigma} + \boldsymbol{b} = \rho_s \boldsymbol{a}, \quad \text{in}\, \Omega_s. \tag{2.1}$$

Here, $\boldsymbol{\sigma}$ refers to the Cauchy stress, ρ_s the solid density and $\boldsymbol{a}$ the acceleration. Equation (2.1) is given in a current configuration. An equivalent formulation, which is used in Calculix [33], would be to define the equations in the reference (undeformed) configuration. While both formulations are mathematically equivalent, the choice of which formulation to use is often a matter of convenience.

The momentum balance equation transformed to the reference configuration [54] is given by

$$\nabla \cdot \boldsymbol{P} + \boldsymbol{b} = \rho_s \boldsymbol{a} \tag{2.2}$$

where $\boldsymbol{P}$ is the first Piola-Kirchhoff stress tensor, which relates to the Cauchy stress via

$$\boldsymbol{P} = \det(\boldsymbol{F})\, \boldsymbol{\sigma} \boldsymbol{F}^{T}. \tag{2.3}$$

The material deformation gradient $\boldsymbol{F}$ is given by

$$\boldsymbol{F} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} = \begin{bmatrix} \frac{\partial x_1}{\partial X_1} & \frac{\partial x_1}{\partial X_2} & \frac{\partial x_1}{\partial X_3} \\ \frac{\partial x_2}{\partial X_1} & \frac{\partial x_2}{\partial X_2} & \frac{\partial x_2}{\partial X_3} \\ \frac{\partial x_3}{\partial X_1} & \frac{\partial x_3}{\partial X_1} & \frac{\partial x_3}{\partial X_1} \end{bmatrix}, \tag{2.4}$$

where $\boldsymbol{x} = \boldsymbol{X} + \boldsymbol{d}(\boldsymbol{X}, t)$.

The various stress measures are related by

$$\boldsymbol{P} = \boldsymbol{F}\boldsymbol{S}, \tag{2.5}$$

and

$$\boldsymbol{\sigma} = \det(\boldsymbol{F})\,\boldsymbol{F}\boldsymbol{S}\boldsymbol{F}^T, \tag{2.6}$$

where $\boldsymbol{S}$ is the second Piola-Kirchhoff stress tensor (the stress in the undeformed or reference configuration).

Closure of the elastodynamics equations requires a constitutive equation relating stress and displacement, which is obtained indirectly through strain:

$$\boldsymbol{S} = \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{E}}, \tag{2.7}$$

where $\boldsymbol{W}(\boldsymbol{F}) = \hat{\boldsymbol{W}}(\boldsymbol{E})$ is the strain energy density function, which depends on the chosen material model. The Green-Lagrange strain, $\boldsymbol{E}$, is defined by

$$\boldsymbol{E} = \frac{1}{2}\left(\boldsymbol{F}^T\boldsymbol{F} - \boldsymbol{I}\right), \tag{2.8}$$

where $\boldsymbol{I}$ is an identity matrix.

A solution to this problem is generally not possible. With a view to approximating the system using the finite element method, the equations are cast into the weak form via the method of weighted residuals [54]. The equation of motion is expressed in the form of a residual $\boldsymbol{R}$,

$$\boldsymbol{R} = \nabla\boldsymbol{P} + \boldsymbol{b} - \rho_s\boldsymbol{a} = \boldsymbol{0}. \tag{2.9}$$

The inner product is then computed between the residual vector field $\boldsymbol{R}$ and a weighting function field $\boldsymbol{w}$. This scalar field is then integrated over the entire domain to obtain

$$\int_{\Omega_S} \boldsymbol{w}\cdot\boldsymbol{R}d\Omega_s = \boldsymbol{0}. \tag{2.10}$$

The weighting function $\boldsymbol{w}$ satisfies the homogeneous boundary condition $\boldsymbol{w} = \boldsymbol{0}$ along $\partial\Omega_{s,D}$. Using the Gauss divergence theorem [54], the (useful) general weak form of the solid mechanics problem is then given by

$$\int_{\Omega_S} \boldsymbol{P}\cdot\frac{\partial \boldsymbol{w}}{\partial \boldsymbol{X}}d\Omega_s = \int_{\Omega_{S,N}} \boldsymbol{t}\cdot\boldsymbol{w}d\Omega_{s,N} + \int_{\Omega_S}\boldsymbol{w}\cdot\boldsymbol{b}d\Omega_s - \int_{\Omega_S}\rho_s\boldsymbol{w}\cdot\boldsymbol{a}d\Omega_s. \tag{2.11}$$

In the classical finite element method, the domain is discretised into a finite number

of non-intersecting finite elements. Both the displacement field and the weighting functions $\boldsymbol{w}$ are chosen to be piecewise-continuous polynomial functions defined over each of the elements [19]. The nodal weighting values $\{W\}$ and displacements $\{D\}$ are then interpolated within an element using the chosen polynomial shape (or basis) function $[N]$ such that

$$\begin{aligned} \boldsymbol{w} &= [N]\{W\}, \\ \boldsymbol{d} &= [N]\{D\}. \end{aligned} \tag{2.12}$$

Since the weighting functions are arbitrary, the boundary value problem can be simplified to the semi-discrete form

$$[\boldsymbol{M}_s]\left\{\ddot{D}\right\} + [\boldsymbol{K}_s(\boldsymbol{d})]\{D\} = \{\boldsymbol{f}_s\}, \tag{2.13}$$

with $[\boldsymbol{M}_s]$ the mass matrix, $[\boldsymbol{K}_s(\boldsymbol{d})]$ the stiffness matrix (associated with a non-linear problem) and $\{\boldsymbol{f}_s\}$ the loading forces given in terms of locally constructed matrices over each element, e,

$$\begin{aligned} \boldsymbol{M}_s &= \int_{\Omega_S} \rho_s [N]^T [N]\, dV_e, \\ \boldsymbol{K}_s(\boldsymbol{d})\,\boldsymbol{d} &= \int_{\Omega_S} \nabla [N] : \{P\}\, dV_e, \\ \boldsymbol{f}_s &= \int_{\Omega_S} [N]^T \{\boldsymbol{b}\}\, dV_e + \int_{\Omega_{S,N}} [N]^T \boldsymbol{t} d\Gamma. \end{aligned} \tag{2.14}$$

In this thesis, the solid equations are integrated in time using the Generalised-α method [18]. The problem time interval $t \in [0, T]$, is divided into a finite number of intervals given a time step size $\Delta t = t_{n+1} - t_n$. The time derivatives are then approximated by

$$\boldsymbol{d}_{n+1} = \boldsymbol{d}_n + \Delta t \dot{\boldsymbol{d}}_n + \Delta t^2 \left[\left(\frac{1}{2} - \beta\right)\ddot{\boldsymbol{d}}_n + \beta \ddot{\boldsymbol{d}}_{n+1}\right], \tag{2.15}$$

$$\dot{\boldsymbol{d}}_{n+1} = \dot{\boldsymbol{d}}_n + \Delta t \left[(1-\gamma)\ddot{\boldsymbol{d}}_n + \gamma \ddot{\boldsymbol{d}}_{n+1}\right], \tag{2.16}$$

where the displacement at the intermediate points are given by

$$\boldsymbol{d}_{n+\alpha_f} = (1-\alpha_f)\,\boldsymbol{d}_n + \alpha_f \boldsymbol{d}_{n+1}, \tag{2.17}$$

$$\dot{\boldsymbol{d}}_{n+\alpha_f} = (1-\alpha_f)\,\dot{\boldsymbol{d}}_n + \alpha_f \dot{\boldsymbol{d}}_{n+1}, \tag{2.18}$$

$$\ddot{\boldsymbol{d}}_{n+\alpha_m} = (1-\alpha_m)\,\ddot{\boldsymbol{d}}_n + \alpha_m \ddot{\boldsymbol{d}}_{n+1}. \tag{2.19}$$

The generalised-α method is a one-step implicit scheme, which is unconditionally stable for linear problems. In general though, implicit schemes behave better for non-linear problems than explicit equivalents. The generalised-α scheme attempts to increase the numerical damping within the system, without degrading the order of accuracy. The default parameter values [56], for which in the linear case the method is unconditionally stable and second order accurate, are

$$\gamma = \frac{1}{2} + \alpha_m - \alpha_f, \;\; \beta = \frac{1}{4}\left(1 + \alpha_m - \alpha_f\right)^2, \;\; \alpha_m = \frac{2\rho_\infty - 1}{1+\rho_\infty}, \;\; \alpha_f = \frac{\rho_\infty}{1+\rho_\infty}, \tag{2.20}$$

where $\rho_\infty \in [0,1]$ is the spectral radius, which controls the numerical damping of the scheme ($\rho_\infty = 0$ is for maximal damping). Several other time integration schemes can easily be derived from the generalised-α method. For example, choosing $\alpha_m = 0$ and $\alpha_f = 0$ produces the Newmark family of time integration schemes, where the classical trapezoidal rule is obtained by setting $\beta = 1/4$ and $\gamma = 1/2$ [62].

2.3 Fluid Sub-domain

The fluid flow domain is governed by the Navier-Stokes equations. Typically, these equations are expressed in an Eulerian reference frame, where fluid flows through a fixed spatial discretisation. In FSI however the fluid domain needs to account for the solid domain deformation. While there are numerous ways in which to deal with this, most CFD solvers, including OpenFOAM (see Section 3.3.1) accommodate the deforming boundary by displacing the fluid boundary and internal discretisation (via mesh movement algorithms). This requires casting the Navier-Stokes equations in an arbitrary-Lagrangian-Eulerian (ALE) reference frame. This approach was first described by Hirt *et al.* [52] and is currently used in most FSI fluid formulations. Examples of alternatives to the ALE approach include the immersed boundary method [46], the fictitious

domain method [4] or a Lagrange multiplier method [45] over a fixed solution grid. Unlike the fixed grid methods, the ALE approach does offer the advantage of being able to better capture the fluid flow physics along the boundaries due to its inherent ability to maintain high-quality fluid flow grids [100]. The ALE based methods are however at a disadvantage for very large boundary deformations. In such cases, ALE based methods often require re-meshing or topological changes. This in turn requires mapping solutions from the old to the new mesh, which may introduce conservation errors.

Considering a fluid domain Ω_f, the governing equations for a viscous, isothermal, incompressible and isotropic Newtonian fluid flow in an ALE reference frame are given by

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} - \boldsymbol{u}_m) \cdot \nabla \boldsymbol{u} = \nabla \cdot \boldsymbol{\sigma}_f + \boldsymbol{b}_f, \text{ in } \Omega_f \tag{2.21}$$

$$\nabla \cdot \boldsymbol{u} = \boldsymbol{0}, \text{ in } \Omega_f. \tag{2.22}$$

Here $\boldsymbol{u}$ is the fluid velocity, $\boldsymbol{u}_m$ the ALE-coordinate system velocity at a given reference position, $\boldsymbol{\sigma}_f$ the fluid Cauchy stress tensor and $\boldsymbol{b}_f$ accounts for the body forces acting on Ω_f. For a Newtonian incompressible fluid the constitutive relation is

$$\boldsymbol{\sigma}_f = -p\boldsymbol{I} + 2\mu\boldsymbol{D}, \tag{2.23}$$

where p is the thermodynamic pressure, μ the fluid viscosity and $\boldsymbol{D}$ the rate of deformation tensor. Substituting (2.23) into the momentum equation and dividing through by fluid density yields

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} - \boldsymbol{u}_m) \cdot \nabla \boldsymbol{u} + \nabla p - \nu \nabla^2 \boldsymbol{u} = \boldsymbol{b}_f, \text{ in } \Omega_f \tag{2.24}$$

where ν is the kinematic viscosity and p now refers to the kinematic pressure (note: we divided the momentum equation through by density to pose the same form of the equations as implemented in OpenFOAM).

In order to solve the equations require that the variables be initialised over Ω_f and the definition of appropriate boundary conditions along the Neumann, $\Omega_{f,N}$, and Dirichlet $\Omega_{f,D}$ boundaries. The two primary primitive boundary conditions can be

posed as

$$\boldsymbol{u} = \boldsymbol{u}_0 \quad \text{on } \Omega_{f,D} \tag{2.25}$$

$$\boldsymbol{n} \cdot \boldsymbol{\sigma} = \boldsymbol{t} \quad \text{on } \Omega_{f,N}, \tag{2.26}$$

where $\boldsymbol{n}$ is the unit normal outward pointing vector. From these two primary boundary conditions several derived boundary conditions can be constructed, such as the commonly used slip or no-slip conditions. The outflow boundary condition remains much debated in literature, with several different practices to choose from. The primary complication relates to having to limit the overall computational domain to a finite size, which often requires imposing a "non-physical" boundary condition. In our case, we make use of a "do nothing" outflow boundary condition. The do nothing condition has been theoretically analysed and numerically validated for use whenever the Dirichlet outlet boundary conditions are not explicitly known [104]. It allows for the formulation of pressure-drop problems, and has been demonstrated to minimally affect the domain of interest, for single outlet, parallel flow problems.

Overall, the incompressible fluid equations describe only the evolution of pressure. A reference pressure should therefore be prescribed, either at a boundary or at a reference point within the computational domain. For FSI problems, it is important that this reference pressure is chosen to be representative of the problem at hand, as it affects the absolute pressure along the FSI wetted interface, which in turn largely governs the overall interface stress state (to be transferred across to the solid domain).

OpenFOAM solves the Navier-Stokes equations using the finite volume method. Similarly to the finite element method, the fluid domain Ω_f is subdivided into a finite number of non-overlapping control volumes, $\mathcal{V}$. In order to apply a numerical scheme, the equations have to be cast into weak form. As with the FEM, this is done via the method of weighted residuals, where we integrate the weighted residual equation over the domain: $\int_{\Omega_f} \boldsymbol{w} \cdot \boldsymbol{R} d\Omega_f = 0$. The primary difference is that the weighting function for the FVM is chosen as [75]

$$\boldsymbol{w}_m(\boldsymbol{x}) = \begin{cases} 1, & \boldsymbol{x} \in \mathcal{V}_m \\ 0, & \boldsymbol{x} \notin \mathcal{V}_m \end{cases},$$

where $\mathcal{V}_m$ denotes the control volume associated with $\boldsymbol{w}_m(\boldsymbol{x})$. The weak form of the momentum equation then becomes

$$\int_{\mathcal{V}_m} \frac{\partial}{\partial t}\boldsymbol{u}d\mathcal{V} + \int_{\mathcal{V}_m} (\boldsymbol{u}-\boldsymbol{u}_m)\cdot\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}}d\mathcal{V} - \int_{\mathcal{V}_m}\frac{\partial p}{\partial \boldsymbol{x}}d\mathcal{V} - \int_{\mathcal{V}_m}\nu\frac{\partial^2 \boldsymbol{u}}{\partial \boldsymbol{x}^2}d\mathcal{V} = \int_{\mathcal{V}_m}\boldsymbol{b}_f d\mathcal{V}. \quad (2.27)$$

Using Gauss's divergence theorem, the divergence terms can be cast as flux terms at the boundary of the control volumes [75], after which the equation now reads as

$$\frac{\partial}{\partial t}\int_{\mathcal{V}_m}\boldsymbol{u}d\mathcal{V} + \int_{\mathcal{A}_m}(\boldsymbol{u}-\boldsymbol{u}_m)\,\boldsymbol{u}\cdot\boldsymbol{n}d\mathcal{A} - \int_{\mathcal{A}_m}p\boldsymbol{n}d\mathcal{A} - \nu\int_{\mathcal{A}_m}\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}}\cdot\boldsymbol{n}d\mathcal{A} = \int_{\mathcal{V}_m}\boldsymbol{b}_f d\mathcal{V}, \quad (2.28)$$

where $d\mathcal{A}$ is the surface area enclosing the associated control volume $\mathcal{V}_m$. Similarly, the continuity equation becomes

$$\int_{\mathcal{A}_m}\boldsymbol{u}\cdot\boldsymbol{n}d\mathcal{A}. \quad (2.29)$$

The difficulty now is to represent accurately the fluxes at the boundaries to the control volumes. For this there are a large number of numerical techniques to interpolate nodal quantities to face centres, and a number of differencing techniques to construct gradient approximations. These will not be covered here, but we refer to [2] for a detailed list of the various methods available in OpenFOAM .

A final constraint on the system, due to the ALE approach, is that the mesh velocity satisfies the Geometric Conservation Law (GCL) [30, 102]. The GCL defines the relationship between the rate of change of the volume $\mathcal{V}$ to the mesh velocity $\boldsymbol{u}_m$ given in weak form as

$$\frac{\partial}{\partial t}\int_{\mathcal{V}_m} d\mathcal{V}_m - \int_{\mathcal{A}}\boldsymbol{n}\cdot\boldsymbol{u}_m d\mathcal{A} = 0. \quad (2.30)$$

The GCL requires that the numerical scheme produces a constant result, exactly and independently from mesh motion . It has been shown, under certain circumstances, that the numerical results or the associated numerical efficiencies are not strictly influenced by respecting or violating the GCL [79]. Equally, however, it has been proven (see for example [36]) that satisfying the GCL corresponds to a "sufficient and necessary condition for a numerical scheme to preserve the nonlinear stability". Violating the GCL may therefore result in spurious oscillations and/or overshoots. Since including the additional GCL constraint adds little computational effort, its inclusion is advised

[36]. For more detail on dynamic mesh handling in OpenFOAM, we refer the reader to [61].

One popular means by which to deal with the incompressibility constraint and excessive memory requirements is to adopt iterative solution methods such as the SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) algorithm, originally put forth by Patankar and Spalding [82]. Similar methods include the SIMPLER (Simple Revised), SIMPLEC (Simple-consistent) or PISO-like (Pressure Implicit with Splitting of the Operators) algorithms [109]. These methods all rely on a predictor-corrector type iterative solution scheme.

To provide a brief description of the SIMPLE method, we adopt here the semi-discrete form of the fluid flow equations, using typical notation commonly adopted in the CFD community [109]. The semi-discrete form of the steady state momentum equation can be written as

$$a_p \{U\}_p = H(U) - \nabla p, \tag{2.31}$$

where a refers to the matrix coefficients obtained from the FVM, and subscript p refers to the cell centre locations (a_p can therefore be interpreted as a diagonal coefficient matrix). $H(U)$ refers to the matrix of coefficients of neighbouring cells multiplied by their respective velocities and further includes all other source terms. $H(U)$ in semi-discrete form can be given as

$$H(U) = -\sum_{nb} a_{nb} \{U\}_{nb} + S_U. \tag{2.32}$$

The subscript nb refers here to the cells neighbouring cell centres p and S_U the remaining source terms (a very loose clumping term and is usually left to mean all other terms not accounted for already). The velocity at cell centres can then be computed by

$$\{U\}_p = \frac{H(U)}{a_p} - \frac{\nabla p}{a_p}. \tag{2.33}$$

The discretised continuity equation may be posed as

$$\nabla \cdot \boldsymbol{u} = \sum_{fc} A \cdot \{U\}_{fc} = 0, \tag{2.34}$$

where A is the outward pointing normal face area vector and subscript fc refers to quantities at face centres. The idea behind the SIMPLE scheme is to substitute (2.33) into the discrete continuity equation (2.34). To do so, however, requires interpolating $\{U\}_p$ from the cell centres to the cell faces fc. To avoid the possibility of an oscillating "checker-board" pressure field when using co-located grids (where p and $\boldsymbol{u}$ are stored at the same spatial locations) the use of correction methods such as Rhie and Chow's interpolation practise [87, 109] are required. While OpenFOAM does not use Rhie-Chow corrections, it makes use of an interpolation correction which is often referred to as "in the spirit" of Rhie-Chow [66, 67]. By interpolating the velocity equations to face centres we obtain the final pressure correction equation as

$$\nabla \cdot \left(\frac{\nabla p}{a_p} \right) = \nabla \cdot \left(\frac{H\left(U\right)}{a_p} \right) = \sum_{fc} A \left(\frac{H\left(U\right)}{a_p} \right)_{fc} . \tag{2.35}$$

Overall then, the SIMPLE algorithm can be summarised as

1. Guess a pressure value, p^*.
2. Solve for velocity, using the guessed value of p^*.
3. Using (2.35), correct the pressure and solve for $p = \Delta p + p^*$.
4. Correct the velocities and boundary conditions using the updated p.
5. Repeat from step 2, until convergence.

Normally, the SIMPLE method is used in conjunction with under-relaxation to improve the convergence behaviour, as the scheme is conditionally unstable. In OpenFOAM, PISO-like algorithms are also commonly used, where the primary differences are that PISO requires no under-relaxation terms and the momentum corrector step is performed multiple times.

2.4 Conclusion

The focus of this chapter was a brief introduction to the solid and fluid sub-problems. For both continuum problems, we posed the strong form, provided the associated weak form and described the solution methods.

Chapter 3

FSI Preliminaries

In Chapters 4 through 7, we analyse in isolation the three primary difficulties encountered in partitioned FSI solution methods. More precisely, and in respective order, we deal with the question of interface information transfer, strong coupling algorithms and fully enclosed FSI domains.

The focus of this chapter is to outline the main design philosophies and the necessary building blocks for the development of a strongly coupled partitioned FSI solver. Much of the information contained in this chapter is alluded to in the chapters to follow.

3.1 FSI Multi-field Problem Description

The term "fluid-structure interactions" describes the two-field problem of interaction between fluid flow and deformable structural bodies. Most FSI problems define a two-way coupled problem, where the fluid motion induces a stress state along the structural interface, which in turn causes deformations which affect the fluid flow. This class of problems, a particular subset of the larger field of multi-physics simulations, requires the development of sophisticated solution procedures, which account for the strong non-linearities to capture the overall two-field interactions.

The notional FSI problem for an arbitrary volume is depicted in Figure 3.1, with a fluid domain given by Ω_f, solid domain, Ω_s and associated boundaries $\partial\Omega_{f,s}$. The two sub-domains are independent from each other, and interact only along a common and shared interface Γ, where $\Gamma = \Gamma_s = \Gamma_f$. The FSI problem requires that both the kinematic and dynamic continuity be satisfied at all times along the shared interface

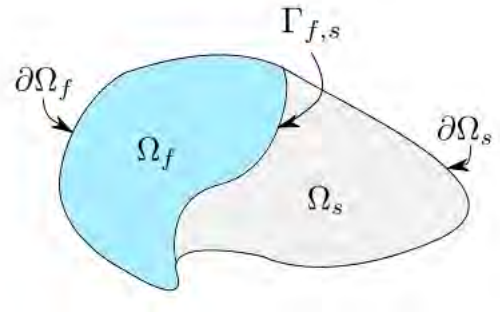


Figure 3.1: Notional FSI problem, depicting the fluid and solid domains, sharing a common interface Γ.

$\Gamma_{f,s}$.

Satisfying kinematic continuity ensures mass conservation, whereas dynamic continuity leads to the conservation of linear momentum. In the case of no-slip boundary conditions on the moving interface, kinematic continuity states that the fluid flow velocity at the interface $\boldsymbol{u}_{f,\Gamma}$ equals the boundary velocity

$$\boldsymbol{u}_{f,\Gamma} = \frac{\partial \boldsymbol{d}_{s,\Gamma}}{\partial t}, \tag{3.1}$$

which can alternatively be stated as $\boldsymbol{d}_{f,\Gamma} = \boldsymbol{d}_{s,\Gamma}$. Dynamic continuity states that the interface stress states are equal at the interface

$$\boldsymbol{\sigma}_{s,\Gamma} \cdot \boldsymbol{n}_{s,\Gamma} = p_{f,\Gamma}\boldsymbol{n}_{f,\Gamma} - \boldsymbol{\tau}_{f,\Gamma} \cdot \boldsymbol{n}_{f,\Gamma}, \tag{3.2}$$

where $\boldsymbol{n}_s$ and $\boldsymbol{n}_f$ are the respective interface normals. Here $\boldsymbol{\sigma}_{s,\Gamma}$ indicates the solid stress state, $\boldsymbol{\tau}_{f,\Gamma}$ the fluid viscous stress tensor and $p_{f,\Gamma}$ the interface pressure.

3.2 Monolithic vs. Partitioned Solution Schemes

The field of FSI can broadly be grouped into two main categories, namely *monolithic* and *partitioned* schemes. The choice of which of these two schemes to apply forms a natural starting point in the discussion on FSI solvers. While both methods aim to solve the same set of equations, the challenges presented by each of the solution schemes require, in general, independent paths of development. Generally speaking, monolithic schemes solve all equations simultaneously, while partitioned solution schemes solve the two sub-domains and interface conditions in a staggered or iterative fashion. Both

schemes have good arguments in their favour, and the specific choice is almost entirely dependent on the necessary outcomes and/or the availability of resources.

Monolithic schemes can in many respects be regarded as the "proper" way to solve the coupled FSI problem. Given the set of equations, the monolithic approach casts the governing equations (including the interface conditions) in terms of the same primitive variables, and discretises the entire domain using the same numerical procedure [9, 55]. Due to the limitations of the FVM when applied to the structural equations [93, 98], this is typically done using the FEM. The equations are then cast into a system of linear equations, and usually solved using some variant of Newton's method. Monolithic schemes allow for seamless coupling between the two sub-domains, and can naturally account for the strong non-linearities present within the system. For FSI problems where the solid to fluid density ratios is close to unity, in which case the cross-coupling terms between the two sub-domains become dominant, the increasing trend in understanding is that monolithic solution schemes are strongly recommended [25, 50, 73]; or alternatively, partitioned schemes with exact computations of the interface Jacobians [32, 39, 73].

The biggest criticism of the monolithic approach is the need to construct, from scratch, highly specialised and very complex software. The solution of flow, and to a lesser extent the structural set of equations requires in most cases several (man) years in development time. Solving for the combined structural, fluid, interface conditions and mesh movement equations simultaneously, can result in linear systems which may quickly become prohibitively large [94]. These combined solution matrices can further lead to poorly conditioned systems, where zero entries along the diagonal are not uncommon, especially when the solid to fluid density ratio increases. These issues require (and are indeed ongoing topics of research) into appropriate matrix pre-conditioners and suitable matrix free solution methods [50, 55, 57].

The overall nature of the two domains is naturally suited to very different requirements in terms of spatial and temporal discretisation. Monolithic schemes however require that the overall domain be discretised using the same computational grid. Constructing such a global grid can be a difficult task, which is exacerbated by complex geometries and the inherent length scale differences [94]. The additional requirement of a single time steps size and unified time integration scheme can lead to further inefficiencies when large time scale differences are present [55].

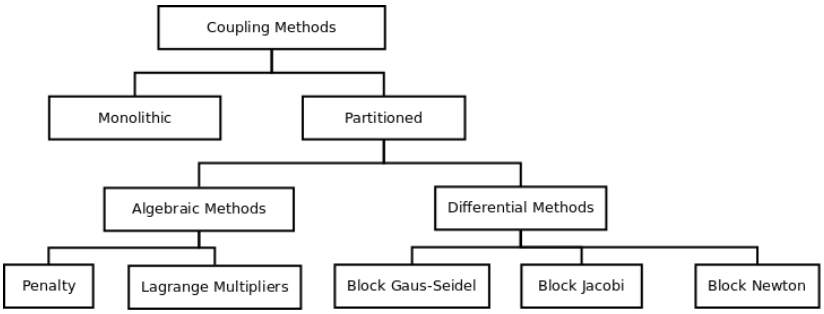


Figure 3.2: Summary of main coupling approaches, reproduced from [68]

For many of these reasons, partitioned solution schemes have been the primary subject in research outputs. In Figure 3.2, a summary of the various coupling methods is provided, reproduced here from [68]. While incomplete, the snapshot of methods in Figure 3.2 does provide an indication of how highly valued the idea of partitioned solution methods are within the research community; equally however, there are a number of unresolved issues, with no single method proving satisfactory for all needs.

Partitioned solution schemes can in general be categorised into explicit or implicit methods (see Figure 3.3). Explicit schemes are those that solve the fluid and solid sub-domains in a staggered fashion with no convergence or residual checks. Explicit, or loosely coupled schemes have been applied with great success to problems in aeroelasticity [38]. In this class of problems, the solid to fluid density ratios are large, leading to a weakly coupled system.

Implicit methods become important when the non-linearity or strength of coupling increases [31, 69]. This can occur for problems with large structural deformations or systems with high "added-mass" [15, 42]. For these kinds of problems, explicit schemes very quickly lead to unstable and inaccurate solutions. Typically, to improve the overall efficiency and robustness of implicit schemes, additional relaxation or acceleration schemes are necessary.

Despite these well known issues, the benefits of black-box solution methods are attractive enough to warrant additional investigation.

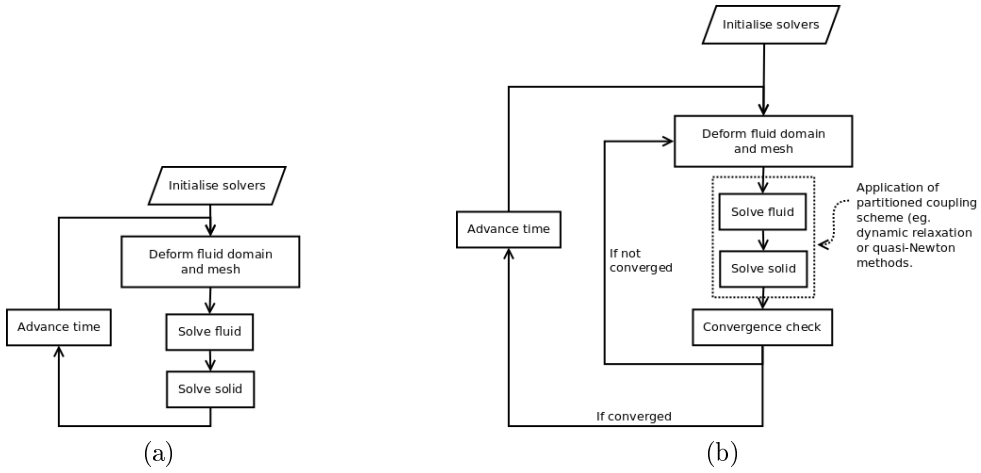


Figure 3.3: Flow charts illustrating (a) weak (or explicit) and (b) strong (or implicit) coupling of partitioned sub-field solvers.

3.3 Partitioned FSI Solver

3.3.1 Fluid and Solid Solvers

In this thesis, OpenFOAM is used to solve the fluid domain problem, and Calculix is used for the structural domain. The associated governing equations were outlined in Chapter 2.

OpenFOAM [2] is an open-source object oriented library for the numerical solution of partial differential equations. While the development branches of OpenFOAM have some rudimentary finite element capabilities, OpenFOAM is more generally regarded as a finite volume based solver; more specifically, OpenFOAM is a co-located, cell-centred FV general partial differential equations solver. OpenFOAM, through the use of high level C++ functionality, comes standard with a host of objects and utilities which can be used to construct solvers at near symbolic level, with easily interchangeable solution specific methods. OpenFOAM includes a large set of pre-existing fluid flow solvers. These include but are not limited to, steady state, multiphase, combustion, transient incompressible and compressible solvers. Despite being an open-source initiative, and being available under a GNU license, it is in many respects a state of the art CFD code, and has comparable capabilities to most high-end commercial CFD packages.

OpenFOAM is released as a set of individual solvers, which are categorised based on the methods and intended application problems. In most of this work we make use

of the *pimpleDyMFoam* solver. *pimpleDyMFoam* is a transient flow solver which uses a PISO-SIMPLE-variant pressure-velocity iterative solution procedure. It is suitable for both incompressible and turbulent flow, and natively supports dynamic or moving computational grids. OpenFOAM supports three-dimensional (3D) elements only. While two-dimensional (2D) simulations are possible, to do so requires constructing a 3D computational grid, with a single element in the contracted dimension. To remain truly generic, we apply the same restriction to the solid domain, where we solve 2D on a 3D grid, with a single element through the thickness, and applying appropriate displacement boundary conditions (to simulate either plane strain or plane stress).

The structural domain is solved here using Calculix, an open-source FEM based solver. Calculix is written in Fortran, and is available under a GPL license. Calculix makes use of an easy to understand Abaqus style input, and is generally regarded as a mature solver. The solver supports both linear and non-linear material models and geometric problems, with many supported element types. In addition, Calculix handles a wide variety of mechanical, thermal, thermo-mechanical and contact field problems. Calculix further allows for easily implemented minor modifications to the functionality via user defined functions and user material models.

3.3.2 Dirichlet-Neumann Interface Conditions

In this thesis, we make use of the Dirichlet-Neumann (DN) interface boundary conditions. DN partitioned algorithms are classified by a Dirichlet boundary (displacement) imposed along the fluid interface Γ_f, where a Neumann interface is applied to the solid domain interface Γ_s in the form of pressures and shear stresses.

DN interface conditions, in conjunction with the incompressibility of the fluid domain and the associated complexities of the added-mass, are responsible for many of the difficulties in staggered solution schemes; including the inability to solve for fully-enclosed domains. This being the case, the question can then be posed: why not reverse the direction and apply Neumann-Dirichlet (ND) boundary conditions? While it is often mentioned that the coupling performance of ND partitioned algorithms is worse, these claims are not always fully substantiated by supporting evidence (see for example [5, 72]). In [15] it was shown that ND partitioned algorithms become unstable when the spatial discretisation tends towards zero. This observation is linked to the use of

relaxation algorithms, but does cast some doubt on the overall usefulness of ND partitioned algorithms. On the other hand, there are contrasting studies, see for example the work of of Hansong [49], in which ND conditions are applied to 1D problems, and were found to provide superior coupling properties.

For academic problems, ND partitioned algorithms can produce superior convergence behaviour. For an example of such a problem, we refer the reader to the 1D problem analysed in Sections 4.5.4 and 5.5.2. For realistic problems however, in our experience, ND partitioned algorithms struggle to converge. The primary reason relates to the elliptic nature of the solid mechanics BVP. The elliptic equations have a very useful smoothing effect when a stress state is imposed along $\Omega_{s,N}$, even in most cases, if this stress state is highly oscillatory. In contrast, applying a displacement field, even when smooth and continuous, can result in a discontinuous or sharply varying stress field. The sensitivity of this displacement-stress relationship is further amplified by increasing the material stiffness (Young's modulus), or by refining the spatial discretisation (supported by the findings in [15]).

A promising alternative to ND interface conditions can be found in the work of Badia *et al.* [5] (and several others, to list but a few [43, 80]) into Robin boundary conditions. Robin-Robin and Robin-Neumann interface transmission conditions have particularly shown promising results in removing much of the instabilities in incompressible FSI, and can further be used to solve for fully-enclosed domains. A Robin boundary is defined as a linear combination of both Dirichlet and Neumann conditions, and allows for the inclusion of the solid domain's stress-displacement response along Γ_f. This is done either via analytical functions or numerical approximations of the solid domain's stress-displacement response. In many regards, the same can be achieved by including artificial compressibility (discussed in greater detail in Chapter 7), which relaxes the incompressibility constraint along the fluid interface. This relaxation is then formulated so that it is also directly linked (in an approximate sense) to the solid domain's stress-displacement response.

3.3.3 Implicit Iterative Coupling

In order to improve the accuracy of the partitioned solvers, we make use of implicit coupling, whereby several iterations are performed in a given time step. The general

outline of the staggered, implicit approach is outlined in Figure 3.3(b). The serial staggering of the two domains implies that there is a natural time lag between the two solutions; a simple illustration of this is shown in Figure 3.4. Iterating between the two solvers until the convergence of the primary variables provides a way to limit the numerically introduced errors. Throughout the study, we make use of the convergence criterion

$$\frac{|\boldsymbol{d}_{\Gamma,k+1} - \boldsymbol{d}_{\Gamma,k}|}{\sqrt{m}} \leq \epsilon, \tag{3.3}$$

where subscript k indicates the iteration counter. m indicates the total number of DOFs along the discrete interface, and is included to remove mesh size dependencies from the stopping criterion.

The partitioned solution methodology allows for different time step sizes to be used for the fluid and solid solvers. Doing so would require interpolation of the results, not only across non-matching spatial discretisation, but also in time. Rather than introducing an additional source of errors, we ensure that the start and final time for both solvers are equal. This does not mean that the solvers themselves cannot be solved using additional time sub-iterations, only that information is transferred between the two domains at equal, finite time intervals $t \in [t_n, t_{n+1}]$, where $t_{n+1} = t_n + \Delta t$.

Additionally, the two sub-domains are solved using implicit time integration schemes. While not strictly necessary, we do so here to guarantee a unique input-output mapping. By this we mean that each of the sub-domain solvers can be thought of, in a black-box sense, as an interface mapping operator. The fluid solver is thus an interface operator which, when given an input displacement, provides an output interface stress state, and vice-versa for the solid domain. The coupling procedures introduced later, require that these mapping operators be unique, which can be obtained (within the limit of the independent solver tolerances) if an implicit time integration scheme is used. We make use of implicit backward-Euler time integration for the flow domain and generalised-α for the solid domain. The implicit-Euler is only first order accurate, and therefore we expect the overall temporal solution order to be limited by the fluid solver. OpenFOAM does allow for higher-order time integration schemes (eg. Crank-Nicholson). The structure and way in which we communicate with OpenFOAM is simplified through the use of implicit-Euler, which only requires information from the previous and current time steps.

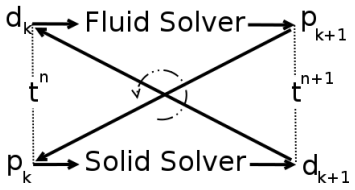


Figure 3.4: Illustration of the staggered in time implicit, partitioned solution scheme.

One of the mild constraints placed on the choice of sub-domain solvers is that they need to be capable of resetting the primary variables to the converged solutions from the previous time step.

3.3.4 Information Transfer and Inter-solver Communication

The communication of stress and displacement between the fluid and solid interface can be broken into two separate questions. The first refers to the mathematical or discrete question of how the information is to be transferred across potentially mismatched interfaces. The second relates to implementation or a programming question, of how the two sub-domain solvers communicate with one another.

Interface information transfer is a critical component for the accuracy of partitioned solution strategies. Generally speaking, along the FSI interface, the computational domains are non-conforming and non-matching. Transferring information between the two domains is therefore non-trivial. In this study, information transfer is performed via Radial Basis Function (RBF) interpolation, using a consistent formulation. This topic is the focus of Chapter 4.

In terms of the physical inter-solver communication, our implementation uses input and output files. There are far more elegant and efficient ways of doing this. From an efficiency perspective, an integrated solution environment would be preferred. Since the source code for both the fluid and solid solvers are available, creating a co-compiled code is possible. This would make all the memory resident primary variables visible to all routines. Since the focus of the study is the development and analysis of numerical methods that are efficient, robust and accurate, we decided to avoid the additional complexities and rather make use of the simplest approach.

As reading and writing files repeatedly to a permanent storage device constitutes an enormous overhead (in terms of computational cycles), we make use of *ramdisk*s

[70]. A ramdisk is a temporary "hard-drive like" storage space allocated in RAM. All input and output files for the two sub-domains are piped through to RAM, and only files that are wanted/needed for post-processing are diverted to the permanent storage device. Since all information remains resident in quick-to-access memory, much of the overhead of using input/output files is removed.

3.3.5 Mesh Movement

Fluid-structure interaction problems involve flow-induced moving boundaries, and in order to accurately complete these simulations requires the computational grid of the fluid domain to conform to the newly displaced domain. This can be done by regenerating the mesh for each sub-iteration. For complex geometries this is an expensive task, especially for three-dimensional simulations. It furthermore requires that solutions be projected from the old to the new computational domains, which typically introduces convective errors. To this end, several mesh movement strategies have been proposed in literature, whereby the computational grid is displaced rather than recomputed, without changing the grid connectivity. Popular examples of mesh movement algorithms include the spring analogy [35], solving a set of Laplacian or biharmonic equations [51], radial basis function (RBF) interpolation [20, 86], solving a pseudo-structure problem [10, 96] or through mesh optimisation [10].

While OpenFOAM natively includes several mesh movement methods, in general they produce poor quality meshes for large deformations. Mesh movement in this thesis is therefore performed external to OpenFOAM, using RBF interpolation. RBF in general offers a mesh movement strategy which is computationally efficient and retains much of the originally generated mesh quality (though not explicitly controlled for). While not introduced here for mesh movement, RBF as a method is outlined in detail in Section 4.4.1 for application to interface information transfer, which can be naturally extended to grid deformation. For a more detailed description and analysis of RBF applied to mesh movement we refer the interested reader to [10, 20].

3.4 Conclusion

We aim to develop a partitioned FSI solver. The coupled environment is designed for use with black-box solvers, which requires the ability to impose and access information along the interface. The flow domain is to be solved using OpenFOAM, and Calculix is used to solve the structure equilibrium equations. The coupled environment is implicit in time, with DN boundary conditions. The transfer of information across the distinct interface and the internal fluid mesh motion is computed through RBF interpolation.

Chapter 4

Interface Information Transfer

4.1 Background

Due to the choice of pursuing a staggered or partitioned solution scheme, we have two independent solvers acting on two independent domains. To couple the two domains requires the transfer of information across the shared interface. The information transferred typically include the interface displacements transferred from the solid domain to the fluid domain, and interface stresses to be transferred from the fluid domain in the form of interface pressures and wall shear stresses.

There are however a number of challenges when considering how information should be transferred between the two independent domains. The numerical properties between the fluid domain and the solid domain differ sufficiently that they are naturally solved with very different geometric discretisation requirements. For example, the fluid solver typically requires far more degrees of freedom than the shared structural problem. Or conversely, the structural domain, either around sharp edges or in vicinities of high stresses, might require localised mesh refinement. This does however mean that information must now be transferred between two non-matching meshes which may contain several geometric inconsistencies.

Consider for example Figure 4.1, illustrating the potential mismatch between a curved fluid and solid interface. The meshes are both mismatched and non-conforming; furthermore, due to the curvature there exists both gaps and overlaps. The problem is further complicated by the choice of using differing numerical schemes on each of the sub-problems. The solid domain problem is solved using the FEM. As such, interface

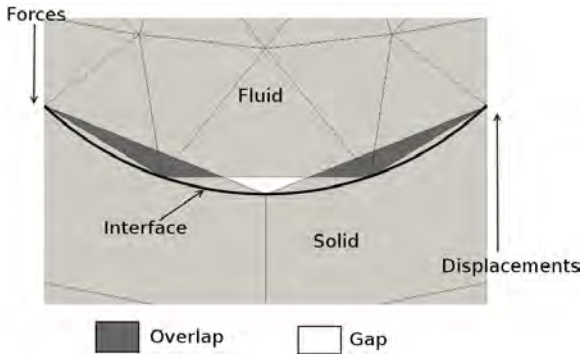


Figure 4.1: Illustrative example of a non-matching mesh along a curved interface.

information is located at the element nodal coordinates with well defined internal interpolation shape functions. The fluid field solver on the other hand, based on the FVM, defines quantities at face centres where the internal interpolation function can at best be described as a face-constant step function.

Accurate information transfer is critical to the accuracy and stability of partitioned FSI solution schemes. There are a large number of interface information transfer schemes available in literature; many are based either on physical arguments [16, 37, 63] or approaches based on mathematical arguments [7, 74, 84, 95]. In general however, when transferring information across a non-conforming, non-matching interface, there is a trade-off between either transferring the information in a conservative or in a consistent sense. Due to the geometric mismatch, it is not possible to satisfy both. By conservative transfer, we imply that the sum of forces are transferred exactly, or that equal work is done along the interface. For consistent information transfer, we adopt the definition to imply that information is transferred in such a manner that a constant stress state can be transferred (or patch test is satisfied).

In this thesis, we have opted to transfer interface information using radial basis functions (RBF) interpolation. Multivariate interface transfer methods have become very popular, primarily because they require no mesh connectivity information. RBF is therefore well suited to coupling FVM-FEM discretisation schemes, and negates any geometric discrepancy.

Many researchers stipulate that conservative information transfer is crucial to the overall stability and accuracy of FSI simulations [74, 83]. To enforce conservatism using multivariate transfer, the interpolation/projection functions are often constructed on

the basis of satisfying virtual work along the interface. In this chapter, we aim to demonstrate that this in essence results in a zero-order scheme, with the possibility of transferring unrealistic and oscillatory stress states. Similar observations have also been noted in [22]. The conservative formulation can also not be used to transfer information between discretisation schemes of different orders. While a consistent formulation is not provably conservative, we will demonstrate that accurate interface transfer remains possible. Furthermore, we aim to demonstrate via numerical examples, that a consistent formulation is convergent, and hence conservative, within the limit of mesh refinement.

The outline for the remainder of the chapter is as follows. We start by outlining the mathematical notation and formulation for consistent and conservative interface information transfer. We provide a brief overview of some of the more common interface transfer schemes before providing a detailed description of RBF interface information transfer. Lastly, we analyse and compare conservative and consistent interface transfer via a patch test, transfer of an analytical function across a curved interface, and a pseudo-1D transient FSI problem.

4.2 Conservative and Consistent Coupling Approaches

4.2.1 Interface Conditions

The coupled FSI problem is a two-field problem, with a fluid domain Ω_f and a solid domain Ω_s. The two domains share a common interface Γ_{FSI}. Due to the discretisation errors, each of the two domains have distinct interfaces, namely Γ_f and Γ_s, where $\Gamma_f \neq \Gamma_s$. The partitioned FSI problem can be viewed as a two-field problem with jump conditions along the interface which need to be satisfied in the form of the kinematic and dynamic continuity conditions. Each FSI cycle requires satisfying the equilibrium of interface tractions $\boldsymbol{t}_f$ and $\boldsymbol{t}_s$, and compatibility of interface velocities, i.e.

$$\boldsymbol{t}_s = \boldsymbol{t}_f, \quad \frac{\partial \boldsymbol{d}_s}{\partial t} = \boldsymbol{u}_f \quad \text{along}\, \Gamma_{\text{FSI}}, \tag{4.1}$$

where $\boldsymbol{t}_f = p_f \boldsymbol{n}_f - \boldsymbol{\sigma}_f \cdot \boldsymbol{n}_f$ and $\boldsymbol{t}_s = \boldsymbol{\sigma}_s \cdot \boldsymbol{n}_s$. p_f denotes the fluid pressure along the interface, $\boldsymbol{\sigma}_f$ the fluid viscous stress tensor and $\boldsymbol{\sigma}_s$ the solid stress tensor; the outward pointing normals along Γ_s and Γ_f are $\boldsymbol{n}_s$ and $\boldsymbol{n}_f$ respectively.

In the event that the meshes are both matching, and the numerical schemes and

order of shape functions are the same, then the discrete information can simply be transferred between the two meshes, i.e. $\boldsymbol{U}_f = \boldsymbol{U}_s$ and $\boldsymbol{T}_s = \boldsymbol{T}_f$. For non-matching meshes, however, an intermediate projection or interpolation step is required:

$$\boldsymbol{U}_f = H_{fs}\boldsymbol{U}_s, \tag{4.2}$$

$$\boldsymbol{T}_s = H_{sf}\boldsymbol{T}_f. \tag{4.3}$$

Here H_{AB} represents the transformation matrix to transfer information from mesh B to mesh A. $\boldsymbol{U}$ and $\boldsymbol{T}$ are the discrete values at the interface points and can be defined by the approximations

$$\boldsymbol{u}(\boldsymbol{x}) \approx \sum_{i=1}^{n^u} N_u^i(\boldsymbol{x})\,\boldsymbol{U}^i, \quad \text{and} \quad \boldsymbol{t}(\boldsymbol{x}) \approx \sum_{j=1}^{n^t} N_t(\boldsymbol{x})_t^j\,\boldsymbol{T}^j. \tag{4.4}$$

$N_{u,t}$ here depends on the spatial discretisation scheme used for the displacement and tractions respectively. Typically for the FEM, $N(\boldsymbol{x})$ is the internal basis/shape functions and a step function for the FVM method; n^u and n^t are the number of degrees of freedom (DOF) along the interface where the discrete displacement and traction quantities are known.

4.2.2 Conservative Information Transfer

The general consensus in literature is that the information transfer should be conservative [7, 37, 59, 58, 74, 84]. By the strictest of definitions this would imply that the integrated quantities should be equal on both Γ_f and Γ_s. The concentrated loads, typically located at the nodal coordinates for the FEM and at face centres for the FVM method, can be defined as

$$\boldsymbol{F}_f = \int_{\Gamma_f} \boldsymbol{t}_f d\Gamma, \tag{4.5}$$

$$\boldsymbol{F}_s = \int_{\Gamma_s} \boldsymbol{t}_s d\Gamma. \tag{4.6}$$

Conservation of forces can then be expressed as

$$\boldsymbol{F}_s = \sum_{i=1}^{n_s} \boldsymbol{F}_s^i = \sum_{j=1}^{n_f} \boldsymbol{F}_f^j = \boldsymbol{F}_f. \tag{4.7}$$

There are however an infinite number of nodal load vectors which will satisfy conservation. A convenient argument often used is the definition of virtual work. For steady state (or for very small time steps), energy can be stated to be globally conserved over the interface if

$$\int_{\Gamma_f} \boldsymbol{u}_f \cdot \boldsymbol{t}_f \boldsymbol{n}_f d\Gamma_f = \int_{\Gamma_s} \boldsymbol{u}_s \cdot \boldsymbol{t}_s \boldsymbol{n}_s d\Gamma_s. \tag{4.8}$$

Using the approximations in (4.4), the semi-discrete form of equation (4.8) becomes

$$\left[M_{ff}\boldsymbol{U}_f\right]^T \boldsymbol{T}_f = \left[M_{ss}\boldsymbol{U}_s\right]^T \boldsymbol{T}_s, \tag{4.9}$$

where matrices M_{ff} and M_{ss} are defined as

$$M_{ff}^{ij} = \int_{\Gamma_f} N_f^i N_f^j d\Gamma, \quad \text{and} \quad M_{ss}^{ij} = \int_{\Gamma_s} N_s^i N_s^j d\Gamma. \tag{4.10}$$

While not strictly correct, matrices M_{ss} and M_{ff} are often referred to in this context as mass matrices [63].

Given some displacement transformation matrix H_{fs} such that $\boldsymbol{U}_f = H_{fs}\boldsymbol{U}_s$, and substituting this into equation (4.9), it is possible to construct a global traction transformation matrix,

$$\boldsymbol{T}_s = \left[M_{ff}H_{fs}M_{ss}^{-1}\right]^T \boldsymbol{T}_f. \tag{4.11}$$

Choosing $H_{sf} = \left[M_{ff}H_{fs}M_{ss}^{-1}\right]^T$, will result in global conservation of interface stress states. If we further recognise that the discrete concentrated forces are defined by $\boldsymbol{F}_s = M_{ss}^T\boldsymbol{T}_s$ and $\boldsymbol{F}_f = M_{ff}^T\boldsymbol{T}_f$, then (4.11) can be re-written as

$$\boldsymbol{F}_s = H_{fs}^T \boldsymbol{F}_f. \tag{4.12}$$

Energy along the interface will therefore provably be conserved if the transpose of the displacement interpolation matrix is used to project the concentrated nodal forces. The force projection in (4.12) is attractive from a black-box coupling perspective. Be-

cause H_{sf} can be constructed using any interpolation method, the projection matrices can be constructed without requiring information regarding the solid domain's internal interpolation and integration schemes. This does however immediately impose a limitation, as it restricts passing information between equal order schemes only (i.e. linear FVM to linear FEM). While the conservative scheme can be used to transfer information between different order schemes, the resulting simulation results will be inaccurate (see Section 4.5.4 for an illustrative example). Ideally, a transfer scheme should at the very least allow for accurate transfer to quadratic FEM internal shape functions.

Furthermore, because $H_{sf} = H_{fs}^T$, it is not possible that the row-sum of both H_{sf} and H_{fs} be equal to 1. The scheme therefore cannot proveably transfer a constant stress state exactly.

4.2.3 Consistent Information Transfer

In this thesis the definition of consistent transfer is adopted from [22], which states that a constant stress state can be exactly transferred. The terminology of consistency stems from the requirement that the interpolation and integration of quantities along the interface should be consistent with the sub-domain schemes. In order to transfer a constant stress state, it is necessary that the row-sum of both H_{sf} and H_{fs} be equal to 1. Consistent information transfer therefore requires that the displacement and force transformation matrices be constructed independently of each other. In other words

$$\boldsymbol{U}_f = H_{fs}\boldsymbol{U}_s, \;\; \boldsymbol{T}_s = H_{sf}\boldsymbol{T}_f, \quad H_{fs} \neq H_{sf}^T. \tag{4.13}$$

For arbitrary non-linear interface fields, there is no guarantee that a consistent scheme will be energy conserving. The fact that the consistent scheme is not provably conservative, does not, as mentioned in [38], imply that the transfer of information will lead to inaccurate or unstable FSI simulations. Transient partitioned FSI solvers are by construction non-conservative due to the time lag between the solid and flow solvers within a given time step. While these errors are reduced with additional sub-iterations within a given time step, the primary concern with the consistent formulation is whether the accuracy and errors introduced by the information transfer are less than those already present. In the limit case of a constant stress state (patch test), provided the constant is exactly transferred, the system will be in static equilibrium, and hence

energy conserving.

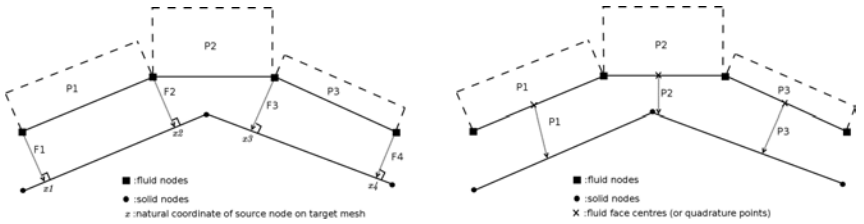
The consistent formulation is slightly less attractive for black-box coupling. Traction fields, and not concentrated nodal forces are transferred from the fluid to solid domain. Some solvers do allow for traction fields (in the form of pressures and shear stresses) to be imposed along a boundary. Usually however, the concentrated nodal forces to be applied along $\partial\Omega_s$ will require a user-implemented integration routine to compute $\boldsymbol{F}_s = M_{ss}^T \boldsymbol{T}_s$.

4.3 Brief Overview of Interface Information Transfer Methods

In this section a brief overview of some of the more common interface transfer methods is given, with the noteworthy exclusion of multivariate interpolation, as this will be covered in greater detail in the sections to follow. The importance of interface information transfer cannot be overstated. It is therefore unsurprising that many techniques have been proposed in literature. The aim here is not to provide a comprehensive overview, but rather highlight a few of the more commonly adopted methodologies.

Probably the simplest form of interface transfer between non-matching meshes is the nearest-neighbour (NN) interpolation method [21, 101]. The NN algorithm consists of finding the matching points on the target mesh which are closest to a given point on the source mesh. Information is then simply transferred from one set of nodes to the other, where the target mesh nodes take on the same values as their closest neighbouring points in the source mesh. Translating the method into the transformation matrix equation (4.3), would imply a Boolean matrix, with the row-column diagonal corresponding to the paired nodes being unity. The NN scheme can be implemented using either the conservative or consistent description, but in general yields poor results when applied to meshes with large mismatch.

One of the earlier interface transfer schemes was a conservative node-projection scheme proposed by Farhat *et al.* [37]. In general the method consists of integrating the interface tractions along the source/fluid mesh. The concentrated loads are then orthogonally transferred to the target/solid mesh (illustrated in Figure 4.2(a)). The nodal loads on the solid surface can then be computed by



(a) (b)

Figure 4.2: One dimensional representation of (a) the node-projection scheme and (b) the quadrature-projection scheme.

$$\boldsymbol{F}_s^j = \sum_{i=1}^{n_f} N_s^j\left(\chi_i\right) \boldsymbol{F}_f^i, \tag{4.14}$$

where χ_i represents the natural co-ordinate location on the solid mesh of node i from the fluid mesh. The method is well suited to FVM-FEM coupling, and based on the summation property of the structural shape functions, the method can be classified as conservative. It is however possible for solid elements to be missed, or receive no information. This may occur in regions where the solid mesh is finer than the fluid mesh. The method can therefore easily be demonstrated to be a zero-order method, where the errors made in surface stresses do not reduce with grid refinement.

A method similar to the node-projection scheme is the quadrature-projection scheme proposed by Cebral *et al.* [16], illustrated in Figure 4.2(b). Similar to the node-projection scheme, Cebral *et al.* proposed projecting the fluid quadrature points onto the solid mesh. $\boldsymbol{F}_s$ is then computed by integrating the surface tractions via numerical quadrature

$$\boldsymbol{F}_s^j = \sum_{i=1}^{m_f} \int_{\Gamma^i} N^j \boldsymbol{t}_f d\Gamma. \tag{4.15}$$

Here m_f is the total number of fluid elements, and Γ^i denotes the i^{th} element. In general, the quadrature-projection method is conservative following the definition in (4.7). The scheme however violates the regularity assumption of quadrature rules [58] which may result in large errors. Furthermore, unlike the FEM, the FVM method does not have a directly related notion to quadrature points save for face centres, where FVM information is typically located. In order to avoid situations where a solid node or

element receives no loads, Cebral *et al.* suggest inserting additional quadrature points in necessary regions within the source mesh; an augmentation which again is poorly suited to the FVM.

An alternative, but commonly used interface transfer method is the method of weighted residuals (WR) or Lagrange multiplier method [22, 63]. Both the node-projection and quadrature projection methods can be viewed as special cases of the WR method; all three methods belong to the general class of Rayleigh-Ritz-Galerkin methods [19, 63]. Suppose we wish to transfer a continuous function $\boldsymbol{g}$ from mesh A to B, the condition that $\boldsymbol{g}_B = \boldsymbol{g}_A$ can then be written in the weak form along the interface as

$$\int_\Gamma w\left(x\right) \boldsymbol{g}_B dx = \int_\Gamma w\left(x\right) \boldsymbol{g}_A dx, \tag{4.16}$$

where $w\left(x\right)$ are the weight functions, or the Lagrange multipliers. Let us assume that $\boldsymbol{g}_{A,B}$ can be given in the discrete form

$$\boldsymbol{g}_A = \sum_{i=1}^{n_A} N_A\left(x\right) \boldsymbol{G}_A, \quad \boldsymbol{g}_B = \sum_{i=1}^{n_B} N_B\left(x\right) \boldsymbol{G}_B, \tag{4.17}$$

where $\boldsymbol{G}_{A,B}$ contains the values of $\boldsymbol{g}_{A,B}$ at the points along the interface. If the mortar element method is used [8, 34, 116], then the Lagrange multiplier is chosen to be a piecewise polynomial and of equal order as the target mesh. In doing so, the approximation can be shown to be optimal for non-conforming meshes. Therefore, transferring from mesh A to B, and substituting the choice of w into (4.16) results in

$$\sum_{i=1}^{n_B} \int_\Gamma N_B^k N_B^i \boldsymbol{G}_B^i dx = \sum_{j=1}^{n_A} \int_\Gamma N_B^k N_A^j \boldsymbol{G}_A^j dx, \quad \text{for } k = 1, ..., n_B, \tag{4.18}$$

which leads to a linear system to be solved for the information transfer. The same result can be found by minimising the L^2 norm of $\boldsymbol{g}_B - \boldsymbol{g}_A$ [63].

While the mortar element method is optimal for non-conforming meshes, some form of interpolation/projection is still required for non-matching meshes, especially along a curved interface. Exact integration of the integrals in (4.18) is required for the WR/mortar method to provide accurate information transfer. To do so for non-matching meshes requires the construction of an overlay mesh, also referred to as the

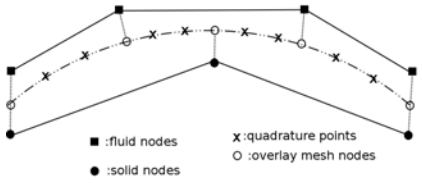


Figure 4.3: Simple 1D example of an overlay mesh.

common refinement based discretisation [59, 58, 63]. The overlay mesh is typically defined by the intersections of the two input meshes (Figure 4.3 shows a simple 1D example). The integration is then performed over the sub-elements of the overlay mesh. The resulting integrals over the sub-elements can in turn be computed exactly, provided that the degrees of the quadrature rules are sufficiently high [63].

For arbitrary three dimensional surfaces, the computation of the element intersections, and hence the construction of the overlay mesh is non-trivial [64, 65]. Furthermore, WR interface information transfer is biased towards FEM-FEM coupling, due to the strong reliance on well defined inter-element interpolation functions and quadrature rules .

Area weighted averaging is an alternative to the WR method for cell-centred data, and can be considered a special case of the common refinement method [63]. Area weighted averaging, also known as conservative remapping or the intersection method [21, 76], is a method whereby the values on the target mesh are based on the area intersections of target and source mesh elements. The nodes and element edges are projected from the source to the target mesh, where the area of intersection is then a weight indicating the magnitude of the quantity of interest to be transferred. As with constructing an overlay mesh, computing element intersections can be computationally demanding. It is furthermore possible that certain regions of curved interfaces may receive no information [21].

4.4 Multivariate Interpolation

Based on our need to couple FVM-FEM, the methods outlined in Section 4.3 are largely ill-suited; an interesting alternative is using multivariate interpolation. The class of multivariate interpolation is based on the idea of using a global interpolation function to transfer information. Multivariate interpolation requires no connectivity information and is therefore well suited to the transfer of information between interface meshes with arbitrary geometric mismatches.

Radial basis functions (RBF) have in particular gained large popularity in the field of multivariate approximation theory. RBF is comparatively simple to implement, the underlying mathematical properties are well understood and provides good interpolation properties (for a numerical comparison to Kriging and the moving least-squares method, see [71]). RBF consists of constructing a global interpolation function where the transfer of information is based on interpolating information from one point cloud to another. The interpolation function is based on fitting a series of splines, or basis functions and the point clouds in question would be the locations, or "centres" where information is known.

4.4.1 Radial Basis Functions

Let us assume we wish to transfer information, $\boldsymbol{s}(\boldsymbol{x})$, from mesh A to mesh B, where $\boldsymbol{s}(\boldsymbol{x}) = \{\boldsymbol{u}, \boldsymbol{t}\}$. At the centres, $x_{A_1}, x_{A_2}, ..., x_{A_n}$ (either nodal coordinates or integration points) of mesh A, we know the discrete values of $\boldsymbol{s}(\boldsymbol{x})$, which we denote here as $g_{A_1}, g_{A_2}, ..., g_{A_n}$, where n_A is the number of DOFs at which information is known along the interface of mesh A. We now wish to construct a continuous function which allows us to interpolate the known values from A to the centres of B. The interpolation function, using RBF, has the following form:

$$\boldsymbol{s}_j = \boldsymbol{s}(\boldsymbol{x}_j) = \sum_{i=1}^{n_A} \alpha_i \phi\left(||\boldsymbol{x}_j - \boldsymbol{x}_{A_i}||\right), \quad \text{for } j = 1, ..., n_A, \tag{4.19}$$

where ϕ is the chosen basis function, $||\cdot||$ refers to the Euclidean distance (in 3 dimensions, $||\boldsymbol{x} - \boldsymbol{x}_{A_i}|| = \sqrt{(x - x_{A_i})^2 + (y - y_{A_i})^2 + (z - z_{A_i})^2}$) and the coefficients α_i are to be solved so that the condition

$$\boldsymbol{s}\left(x_{A_i}\right) = g_{A_i}, \quad \text{for } i = 1, ..., n_A \tag{4.20}$$

is satisfied.

The classical RBF given by equation (4.19) cannot (provably) reproduce constants or linear polynomial terms. This is a rather important requirement for interface information transfer. We would like the interpolating function to be able to transfer constant stress states (for a constant pressure patch test) as well as be able to exactly transfer rigid body motion. To this end, RBF can be augmented by including a polynomial term such that

$$\boldsymbol{s}\left(\boldsymbol{x}\right) = \sum_{i=1}^{n_A} \alpha_i \phi\left(||\boldsymbol{x} - \boldsymbol{x}_{A_i}||\right) + \sum_{j=1}^{m} p_j\left(\boldsymbol{x}\right) \beta_j, \tag{4.21}$$

where $p_j\left(\boldsymbol{x}\right)$ are the monomial terms of the polynomial of degree m, and β_j are m additional constants introduced due to the additional polynomial terms. The m additional constants are obtained by including an additional m constraints in the form

$$\sum_{i=1}^{n_A} p_i\left(\boldsymbol{x}\right) \alpha_i = 0. \tag{4.22}$$

The inclusion of the polynomial has an important consequence. If the function $\boldsymbol{s}\left(\boldsymbol{x}\right)$ can be described exactly by a linear polynomial, and the included polynomial $p\left(\boldsymbol{x}\right)$ is linear, then the polynomial will be exactly reproduced. This follows from, given that ϕ is a conditionally positive definite function, there provably exists a unique function $\boldsymbol{s}\left(\boldsymbol{x}\right)$ which satisfies both conditions (4.20) and (4.22) (see [115] for a proof of this, and [7] for a more detailed discussion).

As a result, the RBF interpolant will exactly match this polynomial. In this thesis we will make use of a linear polynomial of the form $p\left(x\right) = \beta_0 + \beta_1 x + \beta_2 y + \beta_3 z$. By using a linear polynomial we can therefore provably transfer constant information as well as rigid body motion (when transferring displacement). The inclusion of the linear polynomial does place some mild restrictions on our centres. While no additional sampling points are necessary, it does mean that at least 4 points must not fall along a plane. If Γ_{FSI} is in fact a flat face, then a linear polynomial cannot be used.

Using condition (4.20) and (4.22) the following matrix problem can be defined to

solve for the RBF coefficients:

$$\begin{bmatrix} \boldsymbol{g}_A \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{M}_{AA} & \boldsymbol{P}_A \\ \boldsymbol{P}_A^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix}. \tag{4.23}$$

Here $\boldsymbol{\alpha}$ is the vector containing the coefficient sets $\boldsymbol{\alpha}_i$, and $\boldsymbol{\beta}$ is the vector containing the polynomial constants, both to be solved for. For a linear polynomial $p(\boldsymbol{x})$, $\boldsymbol{P}_A$ is an $n_A \times 4$ matrix where each row i is given by $\{1, x_{A_i}, y_{A_i}, z_{A_i}\}$, for $i = 1, 2, ..., n_A$. Finally, $\boldsymbol{g}_A$ refers to the matrix of known values to be interpolated to mesh B. $\boldsymbol{M}_{AA}$ is an $n_A \times n_A$ matrix containing the evaluations of the RBF basis functions

$$\boldsymbol{M}_{AA} = \begin{bmatrix} \phi_{A_1A_1} & \phi_{A_1A_2} & \cdots & \phi_{A_1A_{n_A}} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{A_{n_A}A_1} & \phi_{A_{n_A}A_2} & \cdots & \phi_{A_{n_A}A_{n_A}} \end{bmatrix}, \tag{4.24}$$

where $\phi_{A_1A_2} = \phi\left(||\boldsymbol{x}_{A_1} - \boldsymbol{x}_{A_2}||\right)$.

Once the coefficients $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ have been solved, the interpolated quantities on mesh B, $\boldsymbol{g}_B$ can then be found by

$$[\boldsymbol{g}_B] = \begin{bmatrix} \phi_{BA} & \boldsymbol{P}_B \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix}. \tag{4.25}$$

In other words,

$$\boldsymbol{g}_B = \begin{bmatrix} \phi_{BA} & \boldsymbol{P}_B \end{bmatrix} \begin{bmatrix} \boldsymbol{M}_{AA} & \boldsymbol{P}_A \\ \boldsymbol{P}_A^T & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{g}_A \\ \mathbf{0} \end{bmatrix}. \tag{4.26}$$

The transformation matrix H_{BA} is therefore the first n_B rows and n_A columns of the matrix

$$\begin{bmatrix} \phi_{BA} & \boldsymbol{P}_B \end{bmatrix} \begin{bmatrix} \boldsymbol{M}_{AA} & \boldsymbol{P}_A \\ \boldsymbol{P}_A^T & \mathbf{0} \end{bmatrix}^{-1}. \tag{4.27}$$

The matrix inverse in (4.26) is usually not computed explicitly. We are only interested in $\boldsymbol{g}_B$, which can be found by solving (4.23) and performing the dot product in (4.25).

4.4.2 Basis Functions

RBF has gained significant popularity within engineering and science disciplines for the interpolation of scattered data sets. As such, there are many well documented basis functions from which to choose. Primarily, RBF basis functions can be categorised as either global support functions or compactly supported basis functions. Global support functions construct splines based on the interaction of all cloud points, whereas compactly supported functions allow for localised influences only based on a choice of support radius, which is to be chosen by the user.

The compactly supported C^0 and C^2 piecewise polynomial functions introduced by Wendland [114] have been demonstrated to be robust choices for interface information transfer [7, 22]. The C^0 and C^2 function are defined by

$$C^0 : \left(1 - \left(||x|| / r\right)\right)_+^2 , \tag{4.28}$$

$$C^2 : \left(1 - \left(||x|| / r\right)\right)_+^4 \left(4 \left(||x|| / r\right) + 1\right) . \tag{4.29}$$

Here r refers to the choice of support radius and the subscript $+$ indicates that only positive quantities are taken into account. How to choose r is important to the overall behaviour and interpolation quality of compactly supported functions. A larger value of r typically leads to very good interpolation results. Choosing r too large however, leads to ill-conditioned systems. Equally, smaller values of r leads to a sparsely populated banded matrix which is beneficial for efficient linear system solutions. For good interpolation results, r is typically recommended to be set to $r = 2r_{\max}$, where $r_{\max}$ is the radius which includes all points.

Smith *et al.* [95] and de Boer *et al.* [22], investigate several global support functions for FSI simulations, including the thin plate spline (TPS) and multi-quadratic biharmonic (MQ) RBF functions which provide promising results. These functions are defined as

$$\text{TPS: } ||x||^2 \ln ||x|| , \tag{4.30}$$

$$\text{MQ: } \sqrt{||x||^2 + a}. \tag{4.31}$$

For the MQ RBF function, the shape of the spline is controlled via the parameter a. Small choices of a lead to sharp cone-like splines which flatten out as a is increased. De Boer *et al.* [20, 22], suggest values of a in the range $10^{-5} - 10^{-3}$ for a domain of unit length.

Two global functions, which are often overlooked are the cubic and quintic functions [41],

$$\text{Cubic: } ||x||^3 \tag{4.32}$$

$$\text{Quintic: } ||x||^5 . \tag{4.33}$$

To the best of the author's knowledge, these functions have not been analysed for application to FSI interface information transfer. From our experience, especially the quintic function, provides very good interpolation results. However, both these functions should be used with some care, as they may lead to poorly conditioned matrices when interpolating across very large point clouds.

4.4.3 Control Points

In the construction of the RBF matrices, there are control points and interpolation points. The control points are the locations at which information is known, and interpolation points are the locations where information is to be transferred to. For both the conservative and consistent formulations, displacements are transferred from the solid interface nodes to the fluid interface nodes. For the transfer of displacement, the structural nodes then form the control points, with the fluid nodes being the interpolation points.

In order to remain strictly conservative, then, according to (4.12), the concentrated nodal forces are to be transferred from the fluid nodes to solid nodes. In the FVM, integrated fluid forces are not located at fluid nodes, but rather at fluid face centres. In [74], Lombardi *et al.* suggest using the fluid face centres as control points and the solid nodes as the interpolation points. The effect thereof is to render equation (4.12) into

$$\boldsymbol{F}_s = H_{fs}^T H_{fF} \boldsymbol{F}_F, \tag{4.34}$$

where H_{fF} is then an interpolation matrix from fluid face centres $\boldsymbol{F}_F$ to fluid nodal coordinates $\boldsymbol{F}_f$. While Lombardi *et al.* at no stage compute H_{fF}, this invariably remains the effect when choosing the fluid face centres as the control points for the conservative formulation. The proposed interpolation of Lombardi *et al.* is therefore not strictly conservative, and retains the 0-order property which we will elaborate on in Section 4.5.2.2.

In this study, when analysing the conservative formulation, we use the FVM to interpolate the face centred forces $\boldsymbol{F}_F$ to the fluid nodal locations. For a linear FV mesh, this is equivalent to averaging the face centred quantities over element nodes and is only exact for equally spaced grids (and includes the typical FVM errors when applied to non-uniform grids).

For the consistent scheme, since H_{fs} and H_{sf} are constructed independently, the control and interpolation points do not have to be equivalent. To transfer the stress state, we use the fluid face centres as the control points, and interpolate directly to the solid integration points. As we will demonstrate, this results in good interpolation properties. It does however mean that there may still be an interpolation error present, even if the two interface meshes are exactly matching. Consider for example coupling a linear FVM to a linear FEM mesh consisting of quadrilateral surface elements. In such a case, the conservative information transfer scheme would be exact. For the consistent formulation, there remains a geometric discrepancy between the control points and interpolation points, since the solid element typically contains 4 quadrature points compared to the single fluid face centre. We make mention of this potential flaw, as matching meshes are often taken as the benchmark for interface transfer schemes. In other words, the transfer is exact for matching meshes, with an increasing transfer error as the mesh discrepancy increases. In our implementation, the consistent approach is exact for a constant stress state (regardless of the mismatch), and incurs an error when the interface field is non-linear.

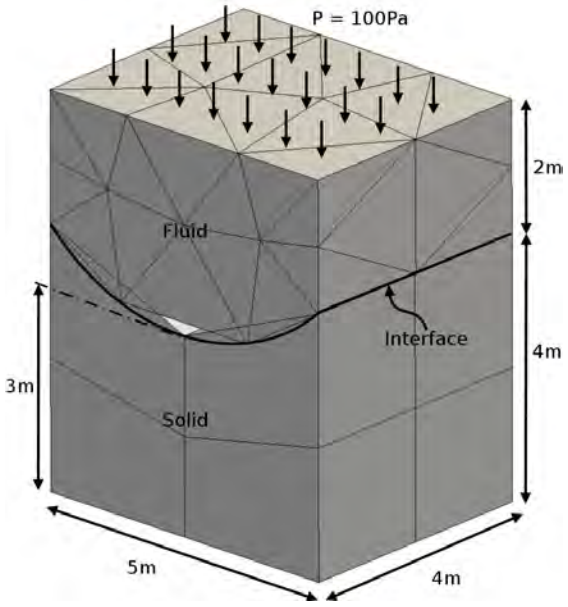
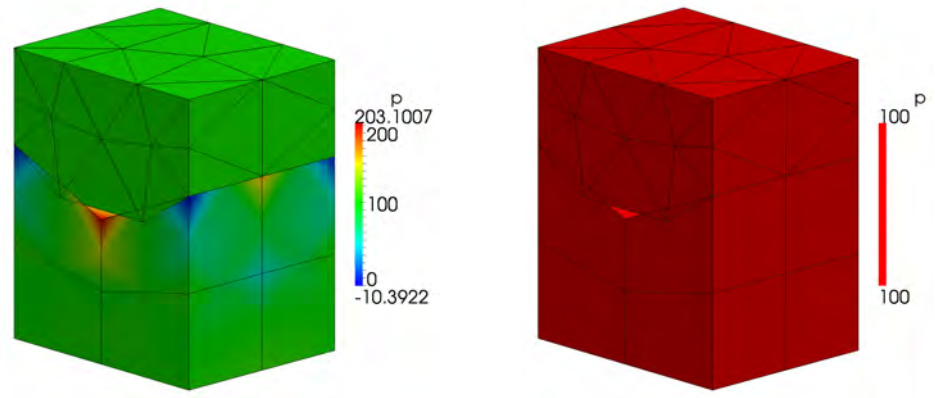


Figure 4.4: Constant pressure problem description.

4.5 Numerical Analysis

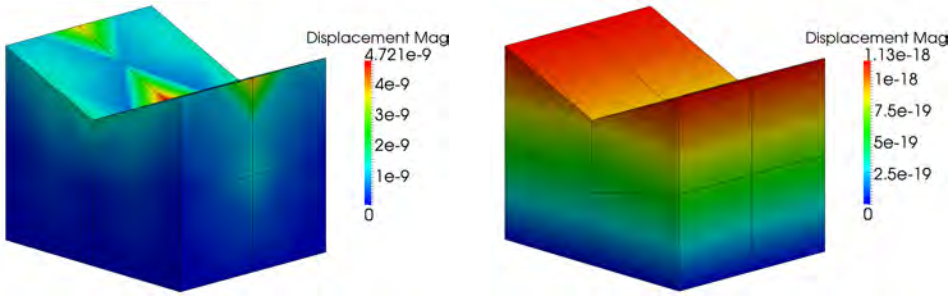
4.5.1 Patch Test

The purpose of the patch test is to determine whether the transfer function can exactly represent a constant stress state. The patch test geometry used in this analysis is shown in Figure 4.4 (see [6] for an alternative proposal of a patch test benchmark problem). We apply a constant, normal pressure along the top of the fluid domain of $p = 100\text{Pa}$. The fluid domain is prescribed with a density and viscosity of $\rho = 1\text{kg/m}^3$ and $\mu = 10\text{kg/ms}$ respectively. For the solid domain, Young's modulus is set to E = 200GPa with a Poisson's ratio of $\nu = 0.49999$ to represent an incompressible material. The fluid domain is discretised with linear tetrahedral elements and the solid domain is described here with quadratic hexahedral elements. The discretised meshes along the curved interface are incompatible, and contain gaps and overlapping regions. The material and problem descriptions are chosen such that the solid domain displacements are very small. The resulting displacements are small enough to be near negligible, but because of the coupled nature of an FSI problem, inaccuracies, even at such small levels remain fundamentally important to the overall solution of the coupled system.



(a) (b)

Figure 4.5: Constant pressure patch test, for (a) conservative information transfer and (b) consistent information transfer.



(a) (b)

Figure 4.6: Displacement field for constant pressure patch test, for (a) conservative information transfer and (b) consistent information transfer.

In Figure 4.5, we show the pressure state for both the conservative and consistent information transfer. The consistent approach exactly transfers the pressure state, and the FSI system is in perfect equilibrium. By contrast, the interface stress state arising from the conservative approach is highly oscillatory. Therefore, despite being exactly conservative, the results deriving from the conservative scheme are incorrect. The inaccuracy of the stress state further manifests itself in an oscillatory displacement field (Figure 4.6).

In Table 4.1 we summarise the fluid forces for the conservative and consistent schemes. The conservative scheme exactly transfers, component wise (x, y and z directions), the fluid interface forces, as it was formulated to provide (via the principle of

Table 4.1: Comparison of the sum of the transferred forces for the constant pressure patch test.

	Conservative		Consistent	
	Fluid	Solid	Fluid	Solid
$\sum F_x$	-1.34e-07	-1.34e-07	-1.07e-10	-8.84e-11
$\sum F_y$	-2000.00	-2000.00	-2000.00	-2000.0 0
$\sum F_z$	-0.332	-0.332	-0.332	1.016e-17
$\lvert\sum F\rvert$	2000.00	2000.00	2000.00	2000.00

virtual work). However, considering that the interfaces have mismatched element areas and interface normals, it does raise the question whether this is what the interface transfer functions should be providing. By comparison, the consistent approach is also exactly conservative; not by the summation definition given in equation (4.7), but by the fact that the integral of the quantities are equal, and hence in static equilibrium.

This is usually where the arguments in favour of consistency fall apart. Typical FSI problems are highly non-linear. What, if any, are the benefits of having a scheme which can only exactly transfer constant stress states? And since it is not possible to marry the ideas of consistency vs. conservatism for non-matching interfaces, is it not better to at least use a scheme which we know will not create or destroy energy? In Section 4.5.2, we will demonstrate via analytical functions, that the conservative scheme, while being exactly conservative, is a 0-order method with regards to the spatial distribution of the transferred forces.

4.5.2 Analytical Test Function

In this section, we compare the accuracy of the consistent and conservative approaches, by transferring an analytical function across a curved interface, for different levels of grid refinement. We make use of the smooth, non-linear function

$$\boldsymbol{s}\left(x,z\right)=\sqrt{\cos\left(x^2+z^2\right)}. \tag{4.35}$$

To measure the accuracy of the transfer schemes we make use of the relative L^2 error defined by

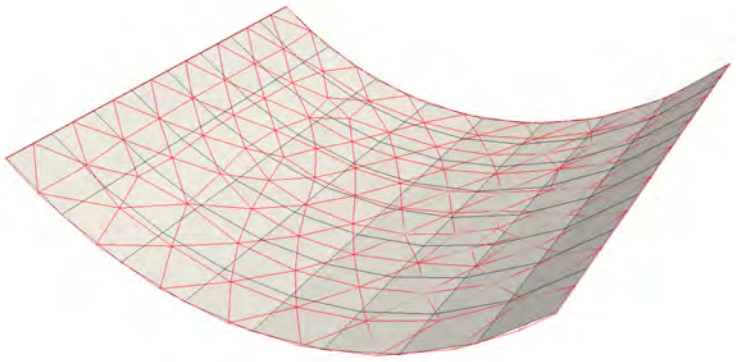


Figure 4.7: Illustrative example of the typical surface mesh mismatch using triangular fluid and four-noded solid surface elements (red: fluid mesh, black: solid mesh).

$$\boldsymbol{s}_{\text{error}} = \sqrt{\frac{\sum_{i=1}^{n} \left(\boldsymbol{s}^i_{\text{exact}} - \boldsymbol{s}^i_{\text{interpolated}}\right)^2}{\sum_{i=1}^{n} \left(\boldsymbol{s}_{\text{exact}}\right)^2}}. \tag{4.36}$$

The interface mesh used for the analysis (depicted in Figure 4.7) is fairly typical of FSI simulations. The fluid interface is described with 2D triangular elements coming from a FVM domain, and the solid interface consists of quadrilateral elements with 4 quadrature points. It is important to mention that we limit our analysis to linear FE meshes for the structural domain. We already mentioned (and will demonstrate more concretely in Section 4.5.4) that the conservative scheme cannot accurately transfer information between linear to quadratic fields. Therefore, rather than biasing our results to the order incompatibility of the internal interpolation and quadrature rules, we wish to demonstrate that the conservative approach is a zero-order method even when transferring between two linear discretisation schemes.

4.5.2.1 Displacement Transfer

We start our accuracy analysis with the error made when transferring displacement. Displacement transfer is the same for both the conservative and consistent formulations, and thus provides an opportunity to analyse the merits of RBF interpolation in isolation.

In Figure 4.8(a), we report the error in displacement as a function of the choice of RBF basis functions for the simultaneous refinement of both the solid and fluid interfaces. These results are further compared to the linear FVM method.

The reported linear FVM convergence rate is the error made when transferring the exact displacement from fluid nodes to the element face centres using the FVM. The FVM error is thus an indication of the error the fluid solver will make when interpolating the known velocities from the nodal co-ordinates to the face centres. Alternatively, it provides an indication of the errors that will be introduced by the subsequent fluid solver step. It is therefore critical that the convergence rates, using RBF to transfer displacement, be lower than FVM convergence rates.

From Figure 4.8(a) it can be seen that all the RBF basis functions do in fact result in displacement errors and in convergence rates that are below the linear FVM error. All the basis functions provide better than linear convergence, and in this case, the quintic basis function results in second order convergence rates. The support radius used for the two compact support functions (C^2 and C^0) is set to $r = 2r_{\max}$. $r_{\max}$ indicates the radius which includes all points (essentially the radius which renders the compactly supported functions into global functions). In Figure 4.8(b) we show the effect of support radius has on the accuracy of the C^2 basis function (the trends are the same for the C^0 function, therefore not shown here). In general, the choice of r only affects the absolute magnitude of the error, and not the associated convergence rate. A large support radius does however mean that the compactly supported functions are computationally more expensive, as the resulting RBF matrices are no longer sparse. A larger support radius also has an impact on the overall condition number of the RBF matrices.

4.5.2.2 Pressure Transfer Error

In Figure 4.9 we show the relative L^2 norm pressure error. The conservative and consistent schemes are indicated with dashed (- · -) and solid lines (–) respectively. For comparison, we include the pressure error for both linear and quadratic FEM interpolation. The FEM interpolation error is computed by providing the exact pressures at the nodal points and interpolating to quadrature points using the internal FEM shape functions. The FEM errors provided here are not indicative of the errors that will be made in any subsequent solid solver step, unlike the FVM errors reported in Section 4.5.2.1.

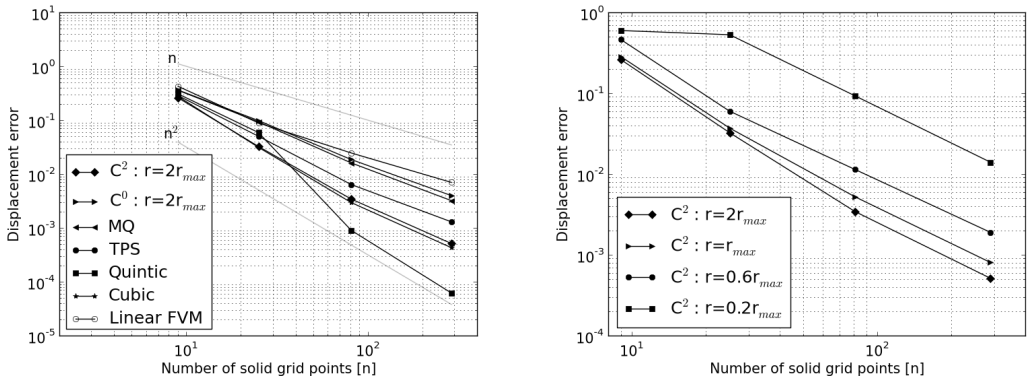


Figure 4.8: Displacement error for the simultaneous refinement of the fluid and solid meshes, comparing (a) different basis functions and illustrating (b) the effect of support radius for the compactly supported C^2 basis function.

It is therefore not *critical* that the interface transfer be more accurate than the cited FEM errors. What it allows however, is to demonstrate the potential accuracy of RBF interface transfer. The consistent approach, transferring information across two distinctly mismatching meshes, compares quite favourably to using the structural shape functions to interpolate quantities internally within the given interface mesh.

The analysis most importantly shows that the conservative approach results in a non-convergent pressure transfer. The reason for this stems from the differences in surface norms and areas, when treating curved surfaces. Despite these local disparities, the conservative formulation exactly transfers, component wise, the forces. Furthermore, should the order of internal shape functions be different, the transferred pressure state will also be highly oscillatory, as was observed in the patch test. This implies that, while the conservative formulation provably transfers the correct magnitude of forces, the spatial distribution and orientation of these forces are incorrect. The same trend has been observed in [22]. This is a fairly concerning result: it implies that no matter how fine the fluid and solid meshes along a curved interface are, the interface stresses will never be correct. As a side note, the conservative pressures are computed from the transferred concentrated forces by $P_s = [M_{ss}]^{-1} H_{fs}^T F_f$.

As outlined in Section 4.2.2, almost all research outputs, including those pertaining to multi-variate interpolation, make use of the conservative formulation. We are there-

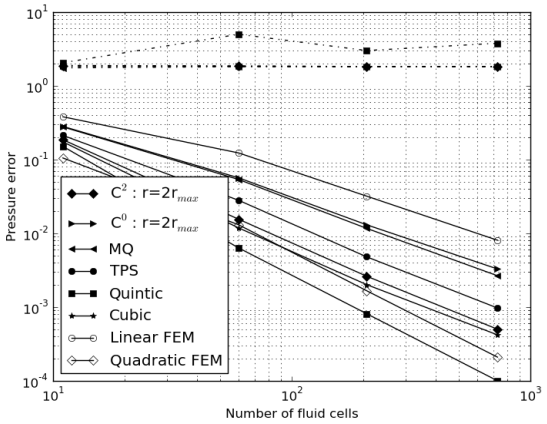


Figure 4.9: Comparison of pressure error for the simultaneous refinement of fluid and solid interface discretisation. Solid lines (—) represent consistent transfer and (- · -) dashed lines represent conservative transfer.

fore hesitant to make the strong claim that the conservative formulation cannot yield accurate FSI simulations; only that the interface stress state will not be correct. It is possible that the stress state along or near the interface is incorrect, while the overall dynamical response of the coupled system is accurate. It should also be mentioned, when applying the conservative formulation to methods such as the mortar element method, that the tractions are applied in a weak setting along the solid interface, and can therefore still be convergent [22] (though at a lower convergence rate than the order of the employed schemes). This is different from multivariate interpolation, where the interface tractions are transferred as concentrated nodal forces, which are then applied explicitly.

4.5.2.3 Energy Conservation

The question that remains is how "un-conservative" is the consistent approach. In Figure 4.10 we depict the difference in work done between the solid and fluid interfaces. For a scheme to be globally conservative, this error should be exactly 0. This is indeed the case when using the conservative approach (by construction), and therefore is not shown. All the basis functions result in an energy error convergence rate of approximately 3.5, almost an order higher than the displacement and pressure convergence rates. Therefore, while the consistent approach is not provably conservative, the

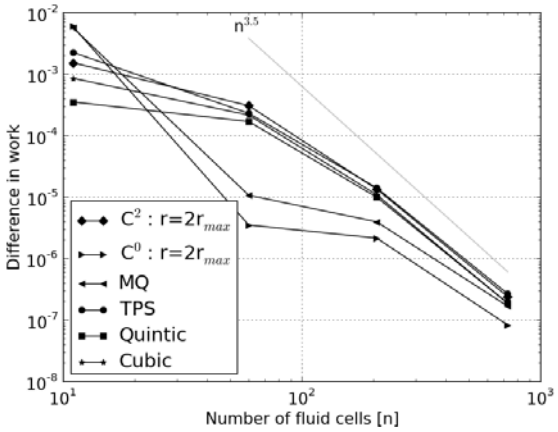


Figure 4.10: Difference in work done across the interface. Only consistent transfer is shown, as the conservative transfer is exactly matching.

incurred energy errors decrease consistently with mesh refinement.

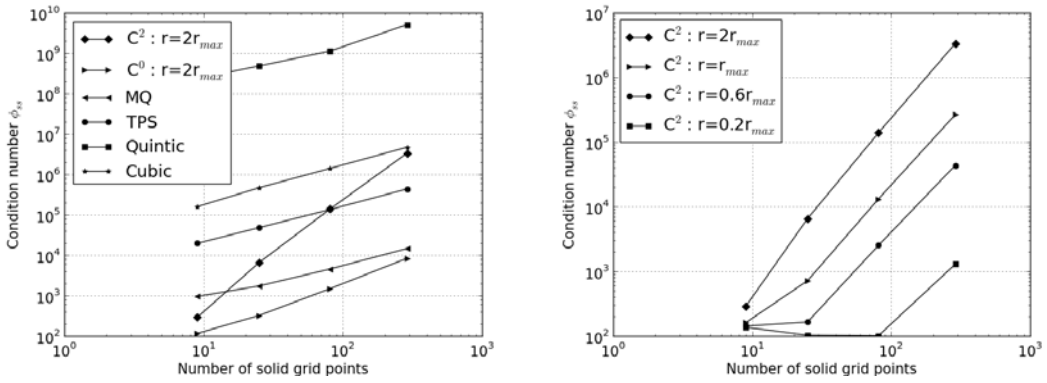
4.5.2.4 RBF matrices condition number

In Figure 4.11 we provide an analysis of the RBF matrix condition number as a function of the point cloud size (Figure 4.11(a)) and as a function of the choice of r for the C^2 function (Figure 4.11(b)).

The condition numbers as a function of the mesh sizes is something which may become an issue for truly large meshes. The quintic basis function for example, provides the most accurate interface information transfer, but also has a condition number which is approximately 4 orders higher than the cubic basis function. Similarly, using a support radius of $r = 2r_{\max}$ results in a condition number almost 3 orders higher than $r = 0.2r_{\max}$. While these condition numbers are not high enough to warrant alarm (at these mesh sizes), it is a factor which may be relevant in the choice of RBF function for very large meshes. It moreover highlights why an inverse computation should be avoided when computing the RBF transfer matrix.

4.5.3 Mesh Mismatch

We have demonstrated RBF interpolation to be a reasonable method by which to transfer information across non-matching, non-conforming meshes. We also demonstrated



(a) (b)

Figure 4.11: Comparison of the condition number as a function of number of grid points for (a) all the basis functions and the (b) C^2 basis function for different support radiuses.

that the conservative approach is a 0-order method and that the consistent approach is convergent within the limit of simultaneous refinement of both the fluid and solid interface meshes. In this section, we aim to provide a basic analysis into how large a mesh-mismatch can be accommodated by the consistent interface transfer scheme. To do so, we freeze one of the surface meshes, and vary the other. To further allow for the case of matching meshes, we use here four-noded structured surface elements, as illustrated in Figure 4.12.

In general, the accuracy of multivariate interpolation improves when there are more control points to interpolation points. Since the fluid mesh is typically finer than the solid mesh, it is therefore expected to be advantageous for interface traction transfer. By the same argument, the larger the ratio of fluid to solid elements, the larger the error made in transferring interface displacements.

We again transfer the analytical displacement and pressure functions used in Section 4.5.2. In Figure 4.13, a plot is provided of the relative L^2 norm pressure error as a function of the mesh mismatch ratio. As mentioned in Section 4.4.3, the consistent formulation cannot exactly transfer pressure between matching meshes, because of the geometric discrepancy between fluid face centres and solid quadrature points. Instead, the pressure error reaches a minimum at roughly four times as many fluid to solid elements (the point at which there are roughly as many fluid elements to solid quadrature points). The pressure error remains relatively constant for fluid-solid element ratios lar-

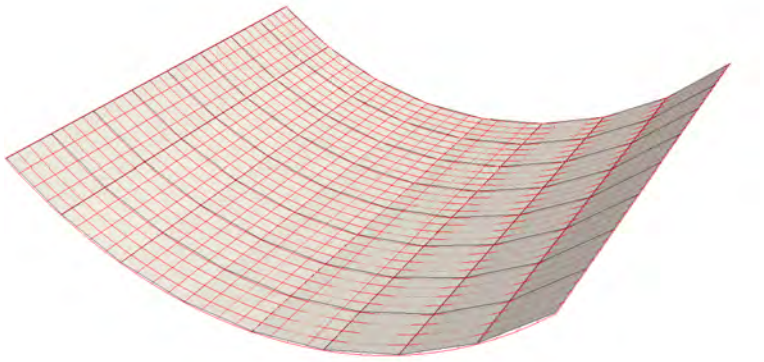


Figure 4.12: Illustrative example of the mesh mismatch using four noded surface elements to analyse allowable mesh mismatch (red: fluid mesh, black: solid mesh).

ger than 1, but rapidly deteriorates when there are many more solid to fluid elements. Fortunately, this is rarely the case in most FSI problems. Overall, the quintic function provides the lowest pressure error for a near unity fluid-solid ratio, but behaves poorly for large mismatches. Overall, the cubic and C^2 functions provide low errors which remain well behaved.

In Figure 4.14, the displacement errors are shown as a function of mesh ratios. We show the displacement errors for both linear and quadratic structural surface meshes. Because of the length scales involved in the two sub-problems, the fluid domain will typically require finer discretisation. This fits in well with the pressure error analysis, where the error remains fairly constant for high fluid-solid element ratios. We can however formulate a mild upper limit on the allowed mismatch based on the linear FVM error as a function of mesh refinement. Ideally, the displacement error should fall below the linear FVM error, which as before is indicative of the error incurred by the fluid solver when interpolating nodal velocities to face centres. In order to avoid confusion, it is important to note that the displacement transfer is exact for fluid-solid ratios 1 and below. This is due to the structured four-noded grids; with fluid-solid ratios 1 and below, the fluid nodes are always coincident with solid nodes, and hence the transfer is exact.

The quadratic surface elements, with more nodes, and hence more control points,

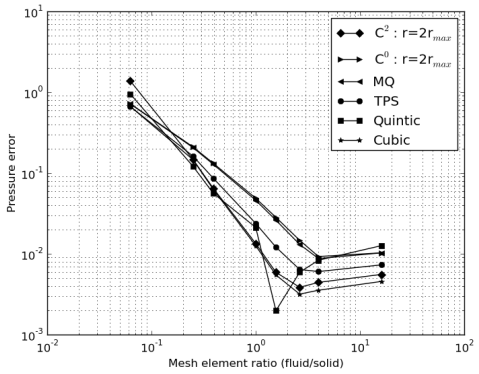


Figure 4.13: Plot of L^2 relative pressure error as a function of the number of fluid to solid element ratio.

naturally produce lower displacement errors. In general, the cubic and C^2 functions again perform well for both the linear and quadratic structural meshes. The behaviour of the quintic function is somewhat more difficult to account for; it provides very good results describing the quadratic displacement field, but performs poorly for a linear field.

Overall though, the conclusions that can be drawn from this analysis are not naturally generalisable. The results shown in Figure 4.14 are directly linked to the discretisation choice (structured, four-noded). Broadly speaking, when coupling a linear (FV) fluid mesh to a quadratic (FE) mesh, the fluid mesh can be approximately an order of magnitude finer than the corresponding solid mesh. Similarly, caution should be used when coupling linear-linear solid and fluid meshes with large mismatches, as the errors introduced by the interface transfer scheme can quickly become a dominant source of errors. These results however, should not supersede those provided in Section 4.5.2.1. Even if the mesh-mismatch is large, the introduced errors will still be reduced with the simultaneous refinement of the fluid and solid interface grids.

4.5.4 1D Test Problem

This section deals with a transient, pseudo-1D piston-channel test problem [6]. The problem layout is described in Figure 4.15, where a 10m long fluid domain is forced out of the domain by the acceleration of a unit by unit solid block. The fluid domain consists of a fluid with density $\rho = 1.0\text{kg/m}^3$, with a viscosity of $\mu = 1.0\text{kg/(m s)}$. No-

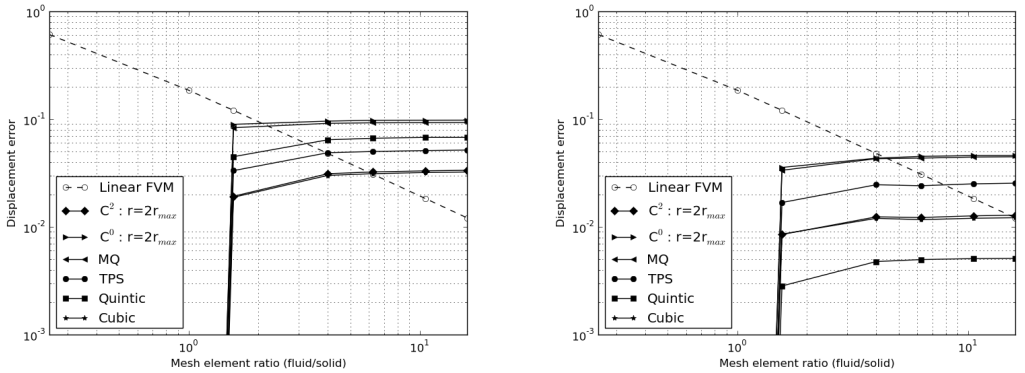


Figure 4.14: Relative L^2 norm displacement error as a function of the fluid to solid element ratios. The errors are compared with the interpolation error of the linear FVM to interpolate nodal velocities from nodes to face centres. (a) Linear FEM surface elements and (b) quadratic FEM surface elements.

slip conditions are applied along all wall boundaries, with an exit pressure of $p = 0$. We make use of a linear elastic, small strain solid with $\rho = 0\text{kg/m}^3$, $E = 10\text{Pa}$, $\nu = 0$, and with a prescribed velocity of $u(t) = 0.2t\,[\text{m/s}]$. The use of small strain linear elasticity only slightly decreases the complexity of the problem as opposed to for example [6] where a hyper-elastic material model is employed. The small strain elasticity does however allow for the construction of a simplified expression for the 1D problem. While the simplified expression does not constitute an analytical expression it is sufficiently cheap to be computed to a high degree of accuracy using any numerical integration scheme with very small time steps.

Following the work of Suliman [98], let us consider the piston-channel problem as a 1D spring-mass system, where the piston acts as a linear spring, and the incompressible fluid as a variable mass system. Based on the balance of forces along the interface, it is possible to construct an expression for the interface displacement d_Γ and interface velocity u_Γ

$$\frac{\partial d_\Gamma}{\partial t} = u_\Gamma, \tag{4.37}$$

$$\frac{\partial u_\Gamma}{\partial t} = \frac{10\,(d_\Gamma - 0.1t^2)}{(d_\Gamma - 10)}. \tag{4.38}$$

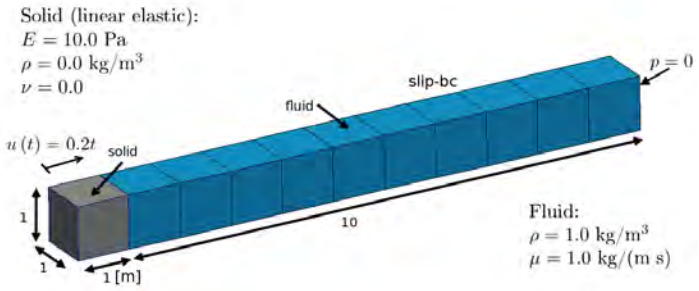


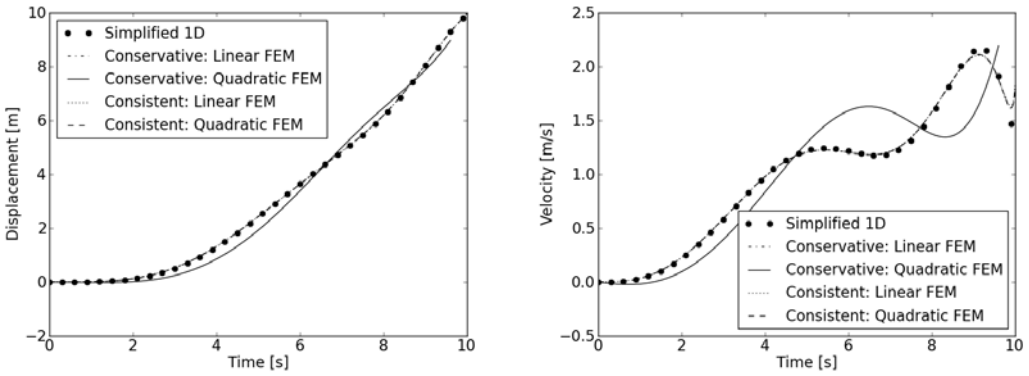
Figure 4.15: 1D piston-channel problem setup.

The force of the solid and fluid domains on the interface can be described by $F_{\Gamma,s} = 10\,(d_\Gamma - 0.1t^2)$ and $F_{\Gamma,f} = (d_\Gamma - 10)$ respectively.

The pseudo-1D problem is used here to analyse the interface transfer methods on a transient problem. While the problem is one dimensional, we solve it here in 3D, where along the interface we have a single, matching four-noded element. Because the interface in 3D is a flat plane, we cannot include the linear polynomial in the RBF approximation, and therefore cannot provably pass a constant (which since the problem is 1D, we in effect are passing across constants). We are concerned here only with the results of the FSI simulation as a function of the interface transfer schemes. The problem is however also very strongly coupled, and in Section 5.5.2 we will re-analyse the problem from a coupling complexity point of view.

The interface displacement and velocities are shown in Figure 4.16, along with the results of the simplified 1D expression (4.38). The consistent transfer for both a linear and quadratic FEM element, as well as the conservative formulation for linear-linear FEM-FVM meshes presents the same displacement and velocities. Because the interface is described by a single coincident element, it may be stated that the conservative transfer between linear-linear FEM-FVM should theoretically be exact. Any deviation from the simplified expression is therefore due to the errors made by each of the sub-domain solvers, and can primarily be attributed to temporal and spatial discretisation errors.

Despite there being no curvature, the conservative scheme produces incorrect results when transferring to a quadratic structural mesh. While this was anticipated, the problem serves as confirmation that the conservative approach, as outlined in Section 4.2.2 cannot be used to couple different orders of spatial discretisation.



(a) (b)

Figure 4.16: Plots of (a) displacement and (b) velocity for the pseudo-1D piston channel problem, comparing the conservative and consistent RBF interpolation using both linear and quadratic structural elements.

4.6 Conclusion

In this chapter, we introduced radial basis function interpolation for interface information transfer. RBF interpolation requires no global connectivity information, and is therefore an elegant means by which to transfer information across a non-matching interface between FVM-FEM schemes. We focused, in particular, on a comparison between conservative and consistent formulations. We demonstrated that the conservative formulation is a zero-order method, where the error made in the spatial distribution of the transferred stress state does not reduce with the simultaneous refinement of the interface meshes. We further demonstrated that the consistent formulation, while not provably conservative, provides consistently decreasing errors in work done along the interface with mesh refinement. Furthermore, the rate of convergence of transfer errors (for both displacement and pressure) are higher than the associated FVM errors. The quintic function provided the highest rates of convergence but should be used with caution, as it rapidly leads to poorly conditioned matrices. Overall, the cubic function appears to provide a good compromise between accuracy and stability, with the C^2 and C^0 basis functions also providing promising results.

Chapter 5

Quasi-Newton Methods for Implicit Black-Box FSI Coupling

5.1 Introduction

In this chapter we introduce a new quasi-Newton method designed for the coupling of transient, black-box FSI solvers. The coupling method is based on a multi-vector iterative updating scheme requiring only observations of the interface tractions and displacements. It allows for the efficient and robust solution of strongly coupled transient FSI problems. The new proposed coupling scheme is heuristic independent and can be applied to a wide range of complex FSI problems, which can be shown under certain circumstances to provide Newton-like convergence behaviour.

We will compare the new proposed scheme to the reduced order modelling (ROM) scheme introduced by Vierendeels *et al.* [110], one of the more promising black-box coupling schemes. We will illustrate how the ROM coupling scheme can be viewed as a minimum norm solution scheme. This provides insights into why the method works comparatively well and provides some explanations as to its behaviour with regards to the number of retained observations in time. We will further compare these methods to the classical Broyden's quasi-Newton method and the popular (though somewhat limited) Aitken's dynamic relaxation.

The outline of the remainder of the chapter is as follows. In Sections 5.2 and 5.3 we describe the general black-box FSI problem within a block-Newton framework.

Section 5.4 introduces the multi-vector Jacobian update scheme and briefly outlines the other quasi-Newton methods investigated in this thesis. Lastly, the performance of the quasi-Newton methods is analysed in Section 5.5 using four popular benchmark problems covering a wide range of complexity, including incompressible flow through a flexible tube where the solid density is an order of magnitude lower than the fluid density.

5.2 Coupled FSI Problem

Since the focus of this thesis is FSI coupling using black-box field solvers, let us review the coupled FSI problem. The choice of discretisation, and numerical methods for black-box solvers are arbitrary, and in this context, the respective field solvers are viewed here as interface operators that map interface displacements and forces. The field solvers operate independently on fluid and structural domains Ω^F and Ω^S which share a common interface Γ. The solid solver is therefore an interface operator $\boldsymbol{S}$ mapping a given interface force vector $\boldsymbol{f}_\Gamma^{n+1}$ to interface displacements

$$\boldsymbol{d}_\Gamma^{S,n+1} = \boldsymbol{S}\left(\boldsymbol{f}_\Gamma^{F,n+1}\right) \tag{5.1}$$

where $\boldsymbol{d}_\Gamma^{n+1}$is the interface displacement vector at time step $n+1$. Similarly, the fluid field solver is represented by a mapping operator $\boldsymbol{F}$ such that

$$\boldsymbol{f}_\Gamma^{F,n+1} = \boldsymbol{F}\left(\boldsymbol{d}_\Gamma^{S,n+1}\right). \tag{5.2}$$

The fluid field operator $\boldsymbol{F}$ denotes both the solution step of the fluid field variables as well as the mesh movement of the fluid domain nodal coordinates.

While we operate under the assumption of "black-box" field solvers, there are some restrictions on the choice of potential solvers. Both solvers need to allow for full access to interface information as well as the ability to prescribe the relevant interface boundary conditions. For strong coupling algorithms, where several coupling iterations are required, it is further necessary that each of the field solvers allow for the primary variables to be reset to the converged solution from the previous time step (while still allowing the user to prescribe the new boundary conditions). Furthermore, since the focus of this chapter is on interface coupling algorithms only, we assume that each of

the field solve mapping operators include the other necessary building blocks required for successful FSI simulations. In this regard, we refer to interface information transfer schemes (especially if non-matching meshes along the interface are employed), mesh deformation schemes and if necessary, depending on the solver choice, that the fluid field solver incorporates an arbitrary Lagrangian-Eulerian (ALE) formulation accounting for convective velocity relative to the domain velocity.

In order to guarantee energy conservation in a partitioned setting, it therefore becomes important to iterate on the two domain solutions (5.1) and (5.2) to within a specified convergence tolerance and require that the interface velocities and stresses are equal along the interface.

5.3 Block-Newton Equations

In order to solve the coupled system defined by (5.1) and (5.2), we rewrite the equations as a root-finding problem, such that

$$\boldsymbol{r}_F = \boldsymbol{F}\left(\boldsymbol{d}\left(\boldsymbol{f}\right)\right) - \boldsymbol{f} = \boldsymbol{0} \tag{5.3}$$
$$\boldsymbol{r}_S = \boldsymbol{S}\left(\boldsymbol{f}\left(\boldsymbol{d}\right)\right) - \boldsymbol{d} = \boldsymbol{0} \tag{5.4}$$

where the functional dependency of forces on displacements and vice versa is indicated by $\boldsymbol{d}\left(\boldsymbol{f}\right)$ and $\boldsymbol{f}\left(\boldsymbol{d}\right)$.

The coupled system defined by equations (5.3) and (5.4) can then be solved by computing the system Jacobian and solving for an update in the Newton direction:

$$\begin{bmatrix} \dfrac{\partial \boldsymbol{F}}{\partial \boldsymbol{d}}\dfrac{\partial \boldsymbol{d}}{\partial \boldsymbol{f}} - \boldsymbol{I} & \dfrac{\partial \boldsymbol{F}}{\partial \boldsymbol{d}} \\ \dfrac{\partial \boldsymbol{S}}{\partial \boldsymbol{f}} & \dfrac{\partial \boldsymbol{S}}{\partial \boldsymbol{f}}\dfrac{\partial \boldsymbol{f}}{\partial \boldsymbol{d}} - \boldsymbol{I} \end{bmatrix} \begin{Bmatrix} \Delta \boldsymbol{f} \\ \Delta \boldsymbol{d} \end{Bmatrix} = \begin{bmatrix} -\boldsymbol{F}\left(\boldsymbol{d}\left(\boldsymbol{f}\right)\right) + \boldsymbol{f}\left(\boldsymbol{d}\right) \\ -\boldsymbol{S}\left(\boldsymbol{f}\left(\boldsymbol{d}\right)\right) + \boldsymbol{d}\left(\boldsymbol{f}\right) \end{bmatrix} \tag{5.5}$$

where the Newton update for coupling iteration $k+1$ is then computed by

$$\boldsymbol{f}_{k+1}^{n+1} = \boldsymbol{f}_{k}^{n+1} + \Delta \boldsymbol{f} \tag{5.6}$$
$$\boldsymbol{d}_{k+1}^{n+1} = \boldsymbol{d}_{k}^{n+1} + \Delta \boldsymbol{d}. \tag{5.7}$$

However, since we are using partitioned solvers, executed in a staggered fashion, we

will solve the system of linear equations in a block-Newton fashion. Following a call to the fluid solver (that returns an interface traction (or integrated forces) f), we update the interface traction iterate as follows:

$$\left(\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{d}}\frac{\partial \boldsymbol{d}}{\partial \boldsymbol{f}} - \boldsymbol{I}\right)\Delta \boldsymbol{f} = -\left(\boldsymbol{F}\left(\boldsymbol{d}\left(\boldsymbol{f}\right)\right) - \boldsymbol{f}\left(\boldsymbol{d}\right)\right) - \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{d}}\Delta \boldsymbol{d}. \tag{5.8}$$

Similarly, following a call to the solid solver (that returns a displacement d), we update this displacement iterate

$$\left(\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{f}}\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{d}} - \boldsymbol{I}\right)\Delta \boldsymbol{d} = -\left(\boldsymbol{S}\left(\boldsymbol{f}\left(\boldsymbol{d}\right)\right) - \boldsymbol{d}\left(\boldsymbol{f}\right)\right) - \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{f}}\Delta \boldsymbol{f}. \tag{5.9}$$

It is important to note here, by recalling equations (5.3) and (5.4), that $\frac{\partial \boldsymbol{d}}{\partial \boldsymbol{f}} = \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{f}}$ and $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{d}} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{d}}$.The main purpose of the quasi-Newton methods discussed in the latter part of this chapter is thus to obtain approximations to the system Jacobians $\boldsymbol{J_s} = \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{f}} = \frac{\partial \boldsymbol{d}}{\partial \boldsymbol{f}}$ and $\boldsymbol{J}_F = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{d}} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{d}}$.

One can alternately set-up a Newton linearised system for the residual equation defined by the difference in displacement between the current and previous iterate:

$$\boldsymbol{r} = \boldsymbol{d}_{k+1} - \boldsymbol{d}_k. \tag{5.10}$$

A Newton system based on the residual equation then becomes

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{d}}\Delta \boldsymbol{d} = -\boldsymbol{r}. \tag{5.11}$$

The benefit of the linearised system (5.11) is that the inverse approximation of $\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{d}}$ can be constructed directly, thereby negating the necessity for a linear system solve step. The quasi-Newton methods discussed in this chapter are equally applicable to the residual system, and in general the performance in terms of number of coupling iterations are comparable, while being somewhat less robust. This can be confirmed by studies by Degroote *et al.* [25, 26], which present a comparison of the IQN-LS and IBQN-LS (discussed in Section 5.4.3) QN methods, based respectively on the residual and block-Newton equations.

5.4 Quasi-Newton Coupling Algorithms for FSI

Quasi-Newton methods are a class of root finding algorithms based on Newton's method, in which the system Jacobian is approximated rather than fully computed.

5.4.1 Multi-vector update method

The requirements for a partitioned coupling algorithm are that it should be both efficient and robust across a wide range of FSI problems. In the context of black-box partitioning, we assume that the largest cost with respect to each of the coupling iterations in a given time step is the time required by each of the respective field solvers. In order to guarantee an efficient solution procedure it becomes important to minimise the number of solver calls and hence the number of coupling iterations. It would further be ideal if the chosen coupling algorithm can be applied to a wide range of problems without the need for tuning a set of problem specific heuristics. In this section, we propose a new, multi-vector update quasi-Newton method (MVQN) to approximate the system Jacobians, which we believe satisfies all the expected requirements.

In order to facilitate the discussion, let us assume that we have completed k FSI coupling iterations within time step $n+1$. We therefore have k interface displacements, which transferred to the fluid solver, subsequently provided k corresponding interface tractions. Using the displacement and traction observations, we can now construct two differencing observations matrices, namely

$$\Delta \boldsymbol{D}_F^{n+1} = [\boldsymbol{d}_k - \boldsymbol{d}_{k-1}, \boldsymbol{d}_{k-1} - \boldsymbol{d}_{k-2}, ..., \boldsymbol{d}_2 - \boldsymbol{d}_1] \tag{5.12}$$

$$\Delta \boldsymbol{F}_F^{n+1} = \left[\boldsymbol{f}_k - \boldsymbol{f}_{k-1}, \boldsymbol{f}_{k-1} - \boldsymbol{f}_{k-2}, ..., \boldsymbol{f}_2 - \boldsymbol{f}_1\right], \tag{5.13}$$

where the subscript F denotes information with regards to the fluid field solver. To simplify notation, the superscript $n+1$ was dropped from each of the respective observation vectors. It should be noted here that the order and form of the differencing in (5.12) and (5.13) can readily be replaced by subtracting the last observation k from all previous observations $1, 2, ..., k-1$. In exact arithmatic the order of the differencing in (5.12) and (5.13) is not important, but may result in minor numerical differences when implemented using finite precision.

Given the two differencing matrices, it is then possible to construct a generalised secant equation for the fluid solver:

$$\boldsymbol{J}_F^{n+1}\Delta\boldsymbol{D}_F^{n+1} = \Delta\boldsymbol{F}_F^{n+1}. \tag{5.14}$$

Here $\boldsymbol{J}_F^{n+1}$ is the fluid system Jacobian relating a change in displacement to a change in interface tractions. The system is however underdetermined, with only k available observations (with each observation of size m) to describe the full interface Jacobian of size $m \times m$, where m is the total number of interface degrees of freedom (DOFs), with $m \gg k$. As such, there are infinite possible solutions which will satisfy (5.14). One way in which to obtain a uniquely defined system is through the minimum norm solution of (5.14), i.e. solving for the minimum $||\boldsymbol{J}_F||$ subject to the constraint $\boldsymbol{J}_F^{n+1}\Delta\boldsymbol{D}_F^{n+1} = \Delta\boldsymbol{F}_F^{n+1}$. Solving for $\boldsymbol{J}_F^{n+1}$ via a minimum norm solution will add curvature to the approximation in only the k directions in which information is available, leaving all other directions within $\boldsymbol{J}_F^{n+1}$ unchanged. The minimum norm yields an approximate Jacobian of the form

$$\boldsymbol{J}_F^{n+1} = \Delta\boldsymbol{F}_F \left([\Delta\boldsymbol{D}_F]^T \Delta\boldsymbol{D}_F \right)^{-1} [\Delta\boldsymbol{D}_F]^T, \tag{5.15}$$

where the superscript $n+1$ was once again dropped from the observation matrices in order simplify notation.

The properties of the minimum norm are quite important here. The approximation in (5.15) can be thought of as a linear interpolator between the known quantities on the LHS and RHS in (5.14). What makes the minimum norm special is that it can exactly reproduce these approximations, but does not impose assumed curvature in the directions where information has not been observed. This is very different when using general interpolation schemes such as radial basis functions or the moving least squares. Both these interpolation schemes will also provably pass through the known observations, but also introduces curvature in all directions as a function of the chosen basis or weighting functions. Because of the strong non-linearity of the coupled interface problem, choosing search directions based on assumed field information will invariably lead to incorrect locations and the eventual divergence of such a scheme.

The approximate Jacobian given by (5.15) is now only based on observations from the current time step. The approximation can be greatly improved by reusing inform-

ation from previous time steps. One manner in which to do so, would be to include interface information from previous time steps in the observation matrices (5.12) and (5.13). Doing so will in fact yield the exact same Jacobian approximation as provided by the popular least-squares, reduced order modelling (IBQN-LS) method of Degroote *et al.* [26] (discussed in Section 5.4.3). While the method has on several occasions been demonstrated to be a capable coupling method, it does suffer from a very problem-dependent choice of how far in time histories should be retained. We therefore would like a Jacobian approximation method that can retain information from multiple time steps without appending additional information.

The MVQN method is therefore based on an iterative updating scheme. Starting with an initial Jacobian from the previous time step $\boldsymbol{J}^n$, we iteratively update the Jacobian based on an update rule of the form

$$\boldsymbol{J}_{F,k+1}^{n+1} = \tilde{\boldsymbol{J}}_{F,k+1}^{n+1} + \boldsymbol{J}_F^n, \tag{5.16}$$

where $\tilde{\boldsymbol{J}}_{F,k+1}^{n+1}$ denotes the Jacobian update approximation. We can then construct a secant equation for $\tilde{\boldsymbol{J}}_{F,k+1}^{n+1}$:

$$\tilde{\boldsymbol{J}}_{F,k+1}^{n+1} \Delta \boldsymbol{D}_F^{n+1} = \left(\Delta \boldsymbol{F}_F^{n+1} - \boldsymbol{J}_F^n \Delta \boldsymbol{D}_F^{n+1} \right). \tag{5.17}$$

Solving for the minimum norm of $\tilde{\boldsymbol{J}}_{F,k+1}^{n+1}$ subject to the constraint of (5.17) we obtain a new Jacobian update rule:

$$\boldsymbol{J}_{F,k+1}^{n+1} \;=\; \boldsymbol{J}_F^n \;+\; \left(\Delta \boldsymbol{F}_F^{n+1} - \boldsymbol{J}_F^n \Delta \boldsymbol{D}_F^{n+1} \right) \left(\left(\Delta \boldsymbol{D}_F^{n+1} \right)^T \Delta \boldsymbol{D}_F^{n+1} \right)^{-1} \left(\Delta \boldsymbol{D}_F^{n+1} \right)^T. \tag{5.18}$$

The update equation (5.18) bears a marked resemblance to the classical rank-1 Broyden's update method (discussed briefly in Section 5.4.2). It is possible to reformulate (5.18) as the Jacobian update which minimises $\left|\left| \boldsymbol{J}^{n+1} - \boldsymbol{J}^n \right|\right|$, which closely relates to Broyden's method, where the update is based on minimising the difference between two successive iterates $\left|\left| \boldsymbol{J}_k^{n+1} - \boldsymbol{J}_{k-1}^{n+1} \right|\right|$.

The primary difference between our proposed multi-vector update and Broyden's update method is the ability of the multi-vector scheme to exactly reproduce all the information from the current time step, as opposed to the last observation of the current time step only. Information from preceding time steps is then matched in a minimum

norm sense only, and is hence naturally less emphasised in the approximation. An interesting effect thereof is a certain inherent consistency with Newton's method, where the non-linear FSI equations are linearised about the current time step. Furthermore, if newer information from the current time step falls along the same directions as previous information, the older information is replaced, thereby entirely removing the possibility of contradictory information.

In a similar fashion, using interface tractions transferred to the solid solver, we obtain k interface displacements and can therefore construct a similar Jacobian for the solid solver

$$\boldsymbol{J}_{S,k+1}^{n+1} = \boldsymbol{J}_S^n + \left(\Delta \boldsymbol{D}_S^{n+1} - \boldsymbol{J}_S^n \Delta \boldsymbol{F}_S^{n+1}\right) \left(\left(\Delta \boldsymbol{F}_S^{n+1}\right)^T \Delta \boldsymbol{F}_S^{n+1}\right)^{-1} \left(\Delta \boldsymbol{F}_S^{n+1}\right)^T, \quad (5.19)$$

where the subscript S now denotes information with regards to the solid field solver.

A brief summary of the method is outlined in Algorithm A.1, attached as an appendix. Because the MVQN method requires that at least two coupling iterations have been performed in order to construct a Jacobian approximation, for the first iteration of the first time step we make use of a fixed point iteration scheme with relaxation factor ω:

$$\boldsymbol{d}_1^{S,0} = \omega \tilde{\boldsymbol{d}}_1^{S,0} + (1-\omega)\, \boldsymbol{d}_0^{S,0}. \quad (5.20)$$

The Jacobian update rules in (5.18) and (5.19) further requires the availability of $\boldsymbol{J}_F^n$ and $\boldsymbol{J}_S^n$. Since we do not have an available starting Jacobian, we set $\boldsymbol{J}_F^0 = [\boldsymbol{0}]$ and $\boldsymbol{J}_S^0 = [\boldsymbol{0}]$ for the first iteration in time step 1.

Remark 1: The convergence tolerance of each of the field solvers should be lower than the FSI coupling tolerance. If this is not true, it remains possible for interface information from the current time step to be contradictory within finite precision.

Remark 2: It is advised that the inverse required by the solution of the Jacobian update equations (5.18) and (5.19) not be solved using matrix inversion but rather through matrix factorisation. While the DOFs of the matrices $\left(\left(\Delta \boldsymbol{D}\right)^T \Delta \boldsymbol{D}\right)$ and $\left(\left(\Delta \boldsymbol{F}\right)^T \Delta \boldsymbol{F}\right)$ are sufficiently small (of size $[k \times k]$) to warrant being solved using matrix inversion it poses the risk of compounding matrix ill-conditioning if present.

Remark 3: The number of vectors in $\Delta \boldsymbol{D}$ and $\Delta \boldsymbol{F}$ should never exceed the number of interface DOFs. If this is the case, the secant equation becomes over-determined

for which the minimum norm solution is no longer valid. The convergence behaviour of the multi-vector update method is sufficiently good that this should rarely be the case, except for systems with very small interface DOFs (for example a 1D FSI problem). Should the number of retained vectors exceed the number of interface DOFs, the Jacobians in (5.18) and (5.19) should be updated and the observation matrices (5.12) and (5.13) cleared. The convergence behaviour will be minimally affected if the problem DOFs are indeed sufficiently small, such that a premature update is necessary. An alternative approach (though not implemented for purposes of this study) would be to use subspace methods like proper orthogonal decomposition, or singular value decomposition to remove the over-defined subspace information, or solving for the least squares Jacobian.

5.4.2 Broyden's Method

Broyden's method is a classical rank-1 update quasi-Newton method. The method was originally developed for the solutions of systems of non-linear equations for which the computation of the Jacobian is excessively expensive [14] . Unlike other popular rank-1 update quasi-Newton methods (for example DFP, BFGS, SR1 etc.), Broyden's method places no restrictions on the update to the system Jacobian to be either symmetric or positive definite (in the case of approximating the system Hessian).

Considering the widespread application and large body of research focusing on black-box coupling techniques it is perhaps surprising that Broyden's method or other similar rank-1 quasi-Newton methods have received little to no attention. One noteworthy exception is the work of Haelterman [47] in which the numerical properties of several quasi-Newton and Krylov methods, including Broyden's method, were investigated for coupled problems. The numerical experiments were however limited to relatively simple 1D test cases.

Broyden's method requires that at least two iterations have been performed. A secant equation can then be constructed based on the difference between the two iteration vectors

$$\boldsymbol{J}_{F,k}^{n+1}\Delta\boldsymbol{d}_{F,k}^{n+1} = \Delta\boldsymbol{f}_{F,k}^{n+1}, \tag{5.21}$$

where $\Delta\boldsymbol{f}_{F,k}^{n+1} = \boldsymbol{f}_{F,k}^{n+1} - \boldsymbol{f}_{F,k-1}^{n+1}$ and $\Delta\boldsymbol{d}_{F,k}^{n+1} = \boldsymbol{d}_{F,k}^{n+1} - \boldsymbol{d}_{F,k-1}^{n+1}$. In Broyden's method [14] the underdetermined system with regards to $\boldsymbol{J}_{F,k}$ is solved by minimising $||\boldsymbol{J}_k - \boldsymbol{J}_{k-1}||$

which results in a Jacobian update formula

$$\boldsymbol{J}_{F,k}^{n+1} = \boldsymbol{J}_{F,k-1}^{n+1} + \frac{\Delta \boldsymbol{f}_{F,k}^{n+1} - \boldsymbol{J}_{F,k-1}^{n+1} \Delta \boldsymbol{d}_{F,k}^{n+1}}{\left|\left|\Delta d_{F,k}^{n+1}\right|\right|^2} \left(\Delta \boldsymbol{d}_{F,k}^{n+1}\right)^T. \tag{5.22}$$

As with the MVQN method, because access to an initial Jacobian is not available, the initial Jacobian is set to zero $\left(\boldsymbol{J}_F^0 = [\boldsymbol{0}]\right)$ for the first iteration of the first time step. The solid field solver Jacobian may equivalently be approximated as

$$\boldsymbol{J}_{S,k}^{n+1} = \boldsymbol{J}_{S,k-1}^{n+1} + \frac{\Delta \boldsymbol{d}_{S,k}^{n+1} - \boldsymbol{J}_{S,k-1}^{n+1} \Delta \boldsymbol{f}_{S,k}^{n+1}}{\left|\left|\Delta \boldsymbol{f}_{S,k}^{n+1}\right|\right|^2} \left(\Delta \boldsymbol{f}_{S,k}^{n+1}\right)^T. \tag{5.23}$$

5.4.3 Least squares, reduced order modelling quasi-Newton method

One of the more popular, and in the author's opinion, to date, the most promising black-box quasi-Newton coupling schemes is the "Interface Block Quasi-Newton with an approximation for the Jacobians from Least-Squares models" (IBQN-LS) introduced by Vierendeels *et al.* [110]. The basic premise is the construction of approximate Jacobians for the block-Newton equations outlined in (5.8) and (5.9) from least squares reduced order models (ROM) of interface observations. A similar method, the "Inverse approximation of the Jacobian from Least-Squares model" (IQN-LS) introduced by Degroote *et al.* [25] follows the same premise as the IBQN-LS model but only applied to the interface residual equation (5.11). There are several publications available on comparisons of the IBQN-LS and IQN-LS method [26, 28], including a comparison of the IQN-LS method to a full monolithic solver [25]. The method has further been demonstrated, for linear systems, to be convergent within $m+1$ iterations [48], where m is the number of system DOFs.

As in Section 5.4.1, let us consider that we have completed k coupling FSI iterations, where we can then construct in the same fashion as before a set of differencing observation matrices

$$\Delta \boldsymbol{D}_F^{n+1} = [\boldsymbol{d}_k - \boldsymbol{d}_{k-1}, \boldsymbol{d}_{k-1} - \boldsymbol{d}_{k-2}, ..., \boldsymbol{d}_2 - \boldsymbol{d}_1] \tag{5.24}$$

$$\Delta \boldsymbol{F}_F^{n+1} = \left[\boldsymbol{f}_k - \boldsymbol{f}_{k-1}, \boldsymbol{f}_{k-1} - \boldsymbol{f}_{k-2}, ..., \boldsymbol{f}_2 - \boldsymbol{f}_1\right]. \tag{5.25}$$

Based on the work of Vierendeels *et al.* [110], an approximation of the interface change in displacement can be written as a linear combination of previous observations via a set of linear expansions coefficients $\boldsymbol{\alpha}$

$$\Delta \boldsymbol{d}_F = \Delta \boldsymbol{D}_F \boldsymbol{\alpha}. \tag{5.26}$$

Similarly, via the same linear expansion coefficients a change in interface tractions may be approximated as

$$\Delta \boldsymbol{f}_F = \Delta \boldsymbol{F}_F \boldsymbol{\alpha}. \tag{5.27}$$

Solving for $\boldsymbol{\alpha}$ via the least squares solution of (5.26) gives

$$\boldsymbol{\alpha} = \left(\Delta \boldsymbol{D}^T \Delta \boldsymbol{D}\right)^{-1} \Delta \boldsymbol{D}^T \Delta \boldsymbol{d} \tag{5.28}$$

which, when inserted into equation (5.27), results in a relation between the change in interface tractions to a change in interface displacements

$$\Delta \boldsymbol{f}_F = \Delta \boldsymbol{F} \left(\Delta \boldsymbol{D}^T \Delta \boldsymbol{D}\right)^{-1} \Delta \boldsymbol{D}^T \Delta \boldsymbol{d}_F, \tag{5.29}$$

where $\boldsymbol{J}_F = \Delta \boldsymbol{F} \left(\Delta \boldsymbol{D}^T \Delta \boldsymbol{D}\right)^{-1} \Delta \boldsymbol{D}^T$ can be viewed as the approximate fluid Jacobian.

The proficiency of the least squares quasi-Newton Jacobian can be greatly improved by including historical observations from previous time steps, as noted by Degroote *et al.* [26]. Degroote *et al.* therefore suggests appending the converged solution observation matrices from q previous time steps such that:

$$\Delta \boldsymbol{D} = \left[\Delta \boldsymbol{D}^{n+1}, \Delta \boldsymbol{D}^{n}, ..., \Delta \boldsymbol{D}^{n+1-q}\right] \tag{5.30}$$

$$\Delta \boldsymbol{F} = \left[\Delta \boldsymbol{F}^{n+1}, \Delta F^{n}, ..., \Delta \boldsymbol{F}^{n+1-q}\right]. \tag{5.31}$$

The convergence performance of the IBQN-LS method is however strongly dependent on the choice of how far in time, q, histories are retained. The choice of q is in fact a problem-dependent heuristic which has to be chosen prior to runtime, with no *a priori* way of determining the optimal choice of q. Most problems become unstable unless q is chosen to be small, while other problems may benefit from a much larger

choice. Degroote *et al.* [26] attribute this behaviour to the fact that histories from far back in time are no longer relevant for newer time steps. While there is certainly some truth to this, we will show in Section 5.5 via test problems that changing the time step size in general does not result in the optimal choice of q changing.

A far more complete explanation can be found by reformulating the IBQN-LS method as a quasi-Newton method where the Jacobian approximation is based on minimising $||\boldsymbol{J}||$ subject to the constraints of the secant equation $\boldsymbol{J}_F \Delta \boldsymbol{D}_F = \Delta \boldsymbol{F}_{\boldsymbol{F}}$. Trying to match information from multiple time steps leads to the possibility of contradicting information. To put it simply, imagine the fluid solver is provided with two identical (or very similar) changes in displacement. It is now possible that for two different time steps, two different changes in interface tractions are observed (or conversely for the solid solver). A minimum norm solution to the secant equation, with internally contradicting vectors, is a least squares fit. This translates to the approximated gradients being computed as the average of the set of contradicting vectors. In so doing, neither of the observations can be exactly matched, where at least one of these observations may be relevant to the current solution space.

An equally relevant problem for minimum norm solutions is linear dependence. This manifests itself in ill-conditioned matrices which affect the overall solution accuracy and the loss of precision. While linearly dependent observations remain possible for both the newly proposed MVQN and Broyden's method, the probability of obtaining linearly dependent vectors increase significantly with an increase in the number of appended (retained) vectors from previous time steps. There are several numerical techniques available to minimise the effects of ill-conditioning. In the chapter to follow, we will pose one such possibility, by augmenting the IBQN-LS method through the use of proper orthogonal decomposition. While this is potentially beneficial for some problems by enabling the method to remain well conditioned and stable for larger choices of q, it significantly increases the computational complexity while providing no solution to the problem of contradictory observations.

It is important to note that the set of remarks highlighted in Section 5.4.1 are also in general applicable here for the IBQN-LS method.

5.4.4 Aitken's dynamic relaxation

Aitken's Δ^2 dynamic relaxation method has attained a large degree of popularity within the field of partitioned FSI coupling primarily due to its simplicity and relative efficiency while facilitating the use of black-box field solvers.

Fixed point iteration schemes iterate on the field operators until the interface displacement residual-drops below some given tolerance ϵ where

$$\boldsymbol{r}_k^{n+1} = \boldsymbol{d}_k^{n+1} - \boldsymbol{d}_{k-1}^{n+1}. \tag{5.32}$$

A displacement update including a relaxation factor ω then becomes

$$\boldsymbol{d}_k^{n+1} = \boldsymbol{d}_k^{n+1} + \omega \boldsymbol{r}_k^{n+1}, \tag{5.33}$$

where as before, k indicates the current coupling iteration and $n+1$ the new time step. For complex FSI problems, for example the class of problems encountered in biomedical problems, fixed relaxation parameters are often insufficient to obtain convergence. Aitken's method augments the fixed point iterates using dynamic relaxation, where the relaxation parameter, ω, is modified at the start of each iteration using the displacement results from two preceding coupling iterates:

$$\omega_{k+1} = -\omega_k \frac{(\boldsymbol{r}_{k-1})^T (\boldsymbol{r}_k - \boldsymbol{r}_{k-1})}{(\boldsymbol{r}_k - \boldsymbol{r}_{k-1})^T (\boldsymbol{r}_k - \boldsymbol{r}_{k-1})}. \tag{5.34}$$

We include Aitken's method in our discussion on quasi-Newton methods by illustrating that Aitken's method is in fact the simplest form of a quasi-Newton method available, where the system Jacobian is approximated with the results from only two coupling iterations. To demonstrate this, let us return to quasi-Newton residual equation (5.11), solving for the updated displacement,

$$\boldsymbol{d}_{\Gamma,k+1}^{n+1} = \boldsymbol{d}_{\Gamma,k}^{n+1} - \left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{d}}\right)^{-1} \boldsymbol{r}_{\Gamma,k+1}^{n+1}. \tag{5.35}$$

Given the results from two previous iterations, we can construct an approximation for

the inverse Jacobian via a secant or FD approximation

$$\Delta \boldsymbol{d} = \left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{d}}\right)^{-1} \Delta \boldsymbol{r}, \tag{5.36}$$

where $\Delta \boldsymbol{r} = \boldsymbol{r}_{\Gamma,k+1} - \boldsymbol{r}_{\Gamma,k}$ and $\Delta \boldsymbol{d} = \boldsymbol{d}_{\Gamma,k} - \boldsymbol{d}_{\Gamma,k-1}$. Given the differencing vectors we can then compute the minimum $\left|\left|\left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{d}}\right)^{-1}\right|\right|$ satisfying the secant equation (5.36) providing a displacement update of

$$\boldsymbol{d}^{n+1}_{\Gamma,k+1} = \boldsymbol{d}^{n+1}_{\Gamma,k} + \frac{\left(\boldsymbol{r}^{n+1}_{k}\right)^T \left(\boldsymbol{r}^{n+1}_{k+1} - \boldsymbol{r}^{n+1}_{k}\right)}{\left(\boldsymbol{r}^{n+1}_{k+1} - \boldsymbol{r}^{n+1}_{k}\right)^T \left(\boldsymbol{r}^{n+1}_{k+1} - \boldsymbol{r}^{n+1}_{k}\right)} \left(-\boldsymbol{r}^{n+1}_{\Gamma,k+1}\right). \tag{5.37}$$

It is easy to show that this yields an identical formulation to Aitken's method seen by inserting the dynamic relaxation parameter from (5.34) into (5.33).

While Aitken's method is simple to implement, the limitation of the method stems primarily from its inability to re-use more iterate data. In contrast, the three other quasi-Newton methods already discussed allows for the re-use of some or all interface data acquired through the course of a simulation.

5.5 Test Problems

In this section we analyse the quasi-Newton methods applied to four incompressible flow benchmark FSI problems. The problems cover a wide spectrum of problem classes and are all considered to be strongly coupled FSI problems, with large added mass ratios requiring the use of implicit coupling strategies.

The approximate Jacobians for both the fluid and solid field solvers are constructed along the solid interface. This is done since the number of DOFs along the solid interface is typically smaller; the coupling performance is however independent of this choice. It is also worth mentioning, while not documented in the results to follow, that the coupling behaviour of the quasi-Newton methods are largely independent of the mesh sizes chosen for the discretisation of each of the domains. The inverse $()^{-1}$ computations required for the MVQN and IBQN-LS methods are solved using QR factorisation. Furthermore, we adopt the notation IBQN-LS(q) to indicate that information from q previous time steps is retained.

It is often suggested that the interface position be approximated at the start of each time step via the extrapolation of a higher order polynomial in time (see for example [25]). In the analyses to follow, we are primarily interested in the ability of the quasi-Newton methods to approximate the system behaviour, including the validity of the Jacobians for reuse over multiple time steps. For this reason, the interface position at the start of each time step is not approximated but set to the converged solution from the previous time step, $\boldsymbol{d}_0^{n+1} = \boldsymbol{d}^n$.

The coupling convergence criterion used for all the benchmark problems to follow is

$$\frac{|\boldsymbol{d}_{\Gamma,k+1} - \boldsymbol{d}_{\Gamma,k}|}{\sqrt{m}} \leq \epsilon, \tag{5.38}$$

where m is the number of DOFs along the interface and is included to remove the dependency of the solution residual to the interface mesh size. The choice of ϵ is varied for the test problems to illustrate performance dependencies on convergence tolerance, if any are present. The convergence criterion is varied from suitable values at which reasonable results can be obtained to 10^{-9} across all problems to gain insights into the asymptotic convergence behaviour. In all cases the convergence tolerance for each of the separate field solvers is set to at least an order of magnitude lower. The initial relaxation factor required for the first iteration of the first time step is set to $\omega = 0.001$ for all test problems and quasi-Newton methods. Lastly, the time step sizes chosen for each of the problems are chosen to be similar to that encountered in literature, where the same problems were solved using monolithic and/or partitioned solution schemes with exact Jacobians. We therefore aim to demonstrate that the approximate Jacobians analysed in this thesis are sufficiently accurate to provide stable solutions despite the comparatively large time steps and fine convergence tolerances.

5.5.1 Flow Induced Oscillating Flexible Beam

The first problem we analyse is flow around a flexible beam attached to a fixed rigid square. The problem considers large deformations of the beam induced by oscillating vortices formed by flow around a square bluff body. The problem was first introduced by Wall [112], and represents the simplest of the test problems we investigate in this chapter; the density of the beam is approximately two orders of magnitude larger than

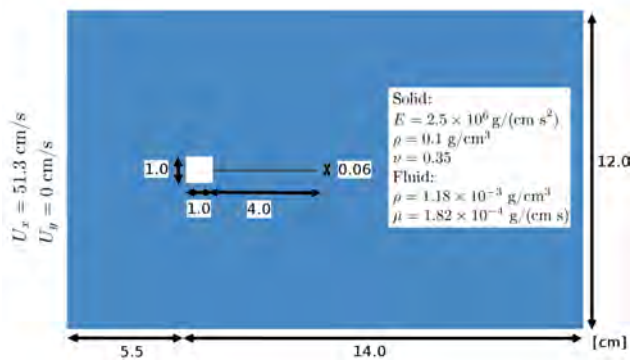


Figure 5.1: Flexible tail behind rigid square test problem set-up.

that of the fluid. The added mass effect, while present, is minimal. The oscillating flexible beam problem is an ideal example in which partitioned solution schemes can be competitive in comparison to monolithic procedures.

The problem set-up is depicted in Figure 5.1. The beam is 0.06cm thick and 4cm long, with linear elastic material properties of Young's modulus $E = 2.5 \times 10^6 \text{g}/(\text{cm s}^2)$, density of $\rho = 0.1 \text{g}/\text{cm}^3$ and a Poisson's ratio $\nu = 0.35$. The material properties result in a fundamental frequency associated with an oscillation period of approximately 0.33 seconds. The square bluff body has side lengths of 1.0cm, with a domain inlet velocity of 51.3cm/s. Slip conditions are applied to the wall boundaries with zero-gradient velocity outlet conditions with the outlet pressure set to 0. The fluid flow density is set to $\rho_f = 1.18 \times 10^{-3} \text{g}/\text{cm}^3$ with a viscosity of $\mu = 1.82 \times 10^{-4} \text{g}/(\text{cm s})$ as per [112].

The mesh used for the analysis is shown in Figure 5.2. A total of 20 quadratic, full integration solid elements along with 3104 linear triangular fluid elements are employed for the purposes of the analysis. Mesh movement is performed using radial basis function interpolation as per [20] where the entire fluid domain is deformed without any requirements for re-meshing. The tip displacement for three time step sizes is shown in Figure 5.3.

A summary of the performance of each of the quasi-Newton coupling methods is given in Table 5.1 for different time steps sizes and convergence tolerances. The average, minimum and maximum number of coupling iterations required for convergence are outlined along with the normalised relative CPU time. For this particular problem all the methods exhibit similar performance, with the MVQN being the most efficient of the four quasi-Newton methods. All four methods are convergent for large time

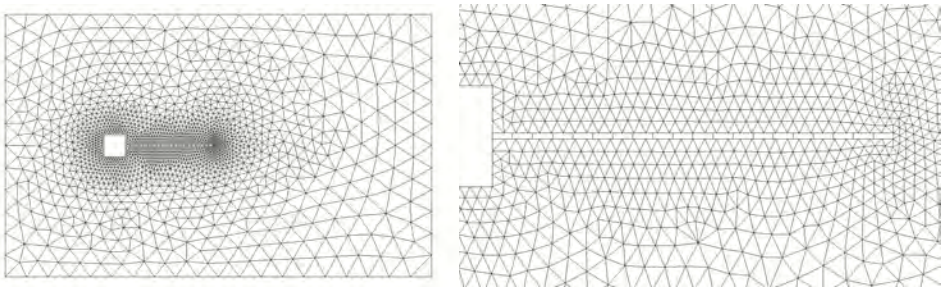


Figure 5.2: Domain discretisation for the oscillating flexible tail test problem. Total of 3104 linear, triangular fluid elements with 20 quadratic, full integration solid elements.

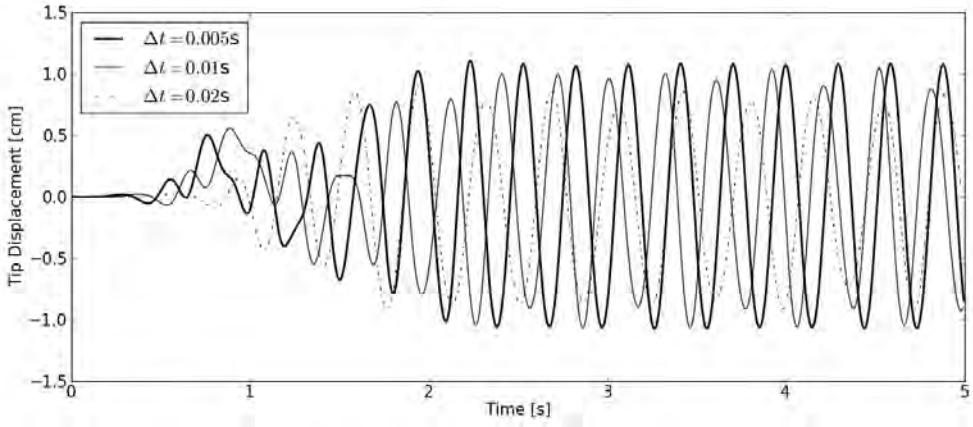


Figure 5.3: Beam tip displacement for the flexible tail problem shown for 3 different time step sizes.

Table 5.1: Comparison of the number of iterations and relative computational time for the flexible tail problem, for different time steps sizes and convergence criterion ϵ. The notation IBQN-LS(q) indicates information from q preceding time steps are retained. If a scheme is non-convergent, the diverging time step as a ratio of the total required time steps is indicated in the CPU column.

	$\Delta t = 0.02, \epsilon = 10^{-6}$				$\Delta t = 0.02, \epsilon = 10^{-9}$				$\Delta t = 0.005, \epsilon = 10^{-9}$			
	Mean	Min	Max	CPU	Mean	Min	Max	CPU	Mean	Min	Max	CPU
MVQN	3.16	3	5	1.00	4.30	4	6	1.00	4.27	3	5	1.00
Aitken	6.12	5	7	1.89	8.76	7	10	1.96	9.91	9	11	2.25
Broyden	3.34	3	5	1.06	4.66	4	6	1.08	4.70	4	6	1.11
IBQN-LS(0)	3.90	3	5	1.25	5.52	5	6	1.28	5.70	4	6	1.33
IBQN-LS(1)	3.57	3	5	1.12	5.02	4	6	1.17	5.00	3	6	1.17
IBQN-LS(2)	3.84	3	6	1.24	5.48	4	10	1.27	5.27	3	8	1.26
IBQN-LS(3)	4.2	3	7	1.33	5.25	4	7	(5/250)	5.89	3	9	1.41
IBQN-LS(4)	4.50	3	8	1.43	5.25	4	7	(5/250)	6.21	3	16	1.51
IBQN-LS(5)	5.30	3	11	(10/250)	5.25	4	7	(5/250)	6.73	3	29	1.59

steps and small convergence tolerances. Similarly, all the methods present favourable properties when comparing the mean number of iterations required to converge for the same step size for a change in ϵ from 10^{-6} to 10^{-9}, on average requiring between 1 to 2 additional iterations. Specifically, the MVQN requires an additional 1.14 iterations.

The typical convergence rates for the MVQN method at different times are shown in Figure 5.4. The benefit of additional information is apparent here when comparing the convergence behaviour of the first to later time steps. Aside from the very first time step, the displacement residual decreases by three orders of magnitude for each additional iteration (super-linear). Within the limit of reasonable convergence criteria, the quasi-Newton method is directly comparable (in terms of the required number of coupling iterations) to a Newton based coupling scheme.

5.5.2 1D dynamic piston-channel problem

We re-analyse here the 1D piston channel problem introduced in Section 4.5.4, with specific focus on the coupling complexity of the problem. For convenience, the problem layout is reproduced here in Figure 5.5. The problem is well documented to be a strongly coupled problem, and is sufficiently difficult that simple fixed point iterative schemes are insufficient to obtain convergence.

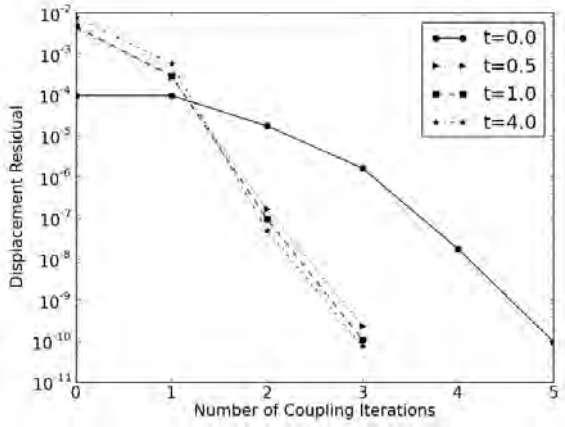


Figure 5.4: Typical MVQN convergence rates for flexible tail problem at different times for $\Delta t = 0.02$s, $\epsilon = 10^{-9}$.

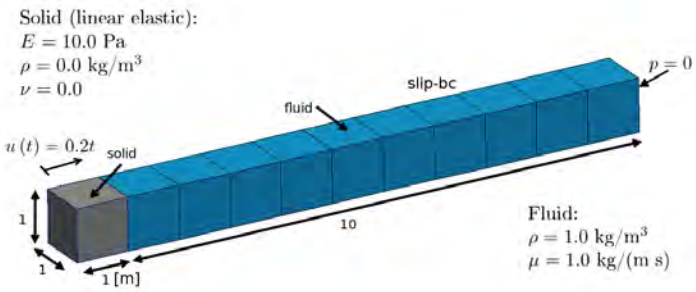
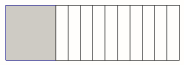


Figure 5.5: 1D piston-channel problem set-up.

(a)



(b)

(c)

Figure 5.6: Domain deformation at simulation time (a) 0s, (b) 8.85s and (c) 10.0s.

In addition to being difficult to solve, from a coupling stability perspective, the problem also requires a capable fluid solver which can handle fairly large Courant numbers. Towards the end of the simulation, the fluid domain is compressed to such an extent that the fluid elements become very small. To illustrate this consider Figure 5.6 showing the fluid domain deformation at three different times. It is also important to note, the choice of coupling schemes do not influence the accuracy of the final solution. The choice of coupling scheme only influences the number of iterations required to obtain a solution (and of course, whether convergence is obtained at all). The interface displacements and velocities are therefore the same as previously reported in Section 4.5.4.

The typical convergence rates for the MVQN method at multiple time steps are shown in Figure 5.7. Even though the interface DOFs are small, the benefit in retaining information from multiple time steps can still be seen, with the progressive improvement in convergence rates compared to the first time step. Once again, aside from the first time step, super-linear asymptotic convergence rates are observed.

A summary of the results of the quasi-Newton methods is shown in Table 5.2. By comparing the results, it is apparent that the MVQN method provides superior results, especially in the presence of large time steps and a fine convergence tolerance.

Providing an explanation as to why the other three schemes perform so poorly is

not straightforward. Firstly, the 1D piston-channel problem is a very strongly coupled problem. The 1D problem is well known to be more challenging than the 2D flexible beam problem analysed in Section 5.5.1 (comparing the results in Table 5.2 to those in Table 5.1 confirms this). Due to the complexity of the problem a very accurate Jacobian approximation is required. In this regard, the approximations provided by both Broyden and Aitken is simply not accurate enough to provide convergent results for large time steps and very fine convergence tolerances.

Explaining why the IBQN-LS method fails is however slightly more involved. While the method has been demonstrated in numerous publications to provide sufficiently accurate Jacobian approximations for most problems, it clearly struggles in this instance. The reason for this can be traced back to the minimum norm approximation. While the problem is 1D in nature, we solve it in three dimensions; along the interface we therefore have 3 DOFs. If more than 3 observation vectors are used to generate the Jacobian approximation, the system becomes over-defined, and the minimum norm solution to the secant conditions become a least squares solution. In other words, neither of the observations that were used to generate the approximation can be exactly matched, with the end result that the convergence behaviour deteriorates.

To rectify the over-defined problem for the IBQN-LS method would require removing vectors from the list of retained observations vectors. Unfortunately, which vectors to drop is not obvious. Intuition would stipulate dropping the oldest information. Often however this will result in dropping information which is critical to the current spatial and temporal location, and again cause a deteriorated convergence behaviour. On the other hand, the updating scheme for the MVQN method bypasses this problem. Whenever the number of observation vectors in the update approximations in (5.18) or (5.19) exceeds the number of interface DOFs, an update can be performed and all retained vectors be cleared. This effectively resets the Jacobian approximation without loosing any of the information which has been gained up to that point. For a true 1D problem (i.e. 1 DOF along Γ_{FSI}) this would render the MVQN method identical to Broyden's method.

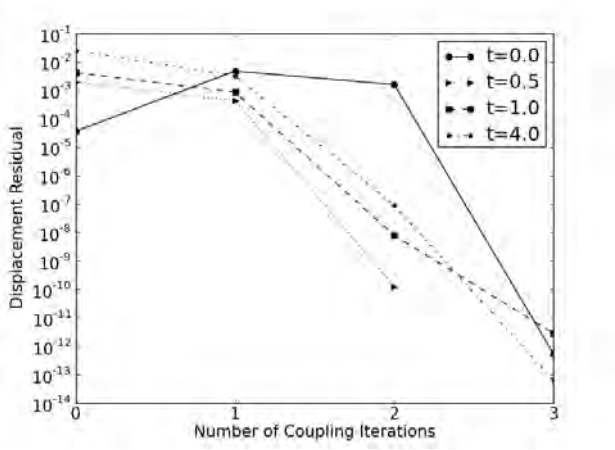


Figure 5.7: Typical MVQN convergence rates for 1D piston-channel problem at different times, $\Delta t = 0.02$s, $\epsilon = 10^{-9}$.

Table 5.2: Comparison of the number of iterations and relative computational time for the 1D piston-channel problem, for different time steps and convergence criterion ϵ. The notation IBQN-LS(q) indicates information from q preceding time steps are retained. If a scheme is non-convergent, the diverging time step as a ratio of the total required time steps is indicated in the CPU column.

	$\Delta t = 0.02, \epsilon = 10^{-6}$				$\Delta t = 0.02, \epsilon = 10^{-9}$				$\Delta t = 0.01, \epsilon = 10^{-6}$			
	Mean	Min	Max	CPU	Mean	Min	Max	CPU	Mean	Min	Max	CPU
MVQN	3.00	3	4	1.00	3.53	3	5	1.00	3.00	3	4	1.00
Aitken	16.12	9	67	3.95	–	–	–	(1/500)	–	–	–	(1/1000)
Broyden	3.53	3	4	1.20	4.47	3	24	(344/500)	3.01	3	4	1.01
IBQN-LS(0)	3.96	3	4	1.36	4.13	4	5	(170/500)	3.95	3	4	1.31
IBQN-LS(1)	4.33	3	11	(489/500)	6.29	3	32	(473/500)	3.93	3	7	1.29
IBQN-LS(2)	4.32	3	9	(449/500)	6.33	3	28	(498/500)	4.09	3	7	1.35

5.5.3 Flexible Tube, $\rho_s = 1.2\mathbf{g/cm^3}$ and $\rho_f = 1.0\mathbf{g/cm^3}$

Let us now turn our attention to a 3D internal flow problem through a flexible tube proposed in [44], which is inspired by the type of flow problems encountered in haemodynamics. The density ratios of the fluid and solid are near unity, which in conjunction with internal incompressible flow results in a very strongly coupled FSI problem. Realistic results can however be obtained for relatively small discretisation DOFs and as a result has gained a lot of popularity as an FSI benchmark problem.

The flexible tube has a length of $l = 5$cm, with an inner and outer radius of $r_i = 0.5$cm and $r_0 = 0.6$cm respectively. A hyper-elastic, neo-Hookean material model with Young's modulus of $E = 3 \times 10^6$dynes/cm^2, density $\rho = 1.2$g/cm^3 and Poisson's ratio of 0.3 is modelled with 96 twenty noded quadratic brick elements with full integration. The fluid flow has a density of $\rho = 1.0$g/cm^3 and viscosity of $\mu = 0.03$P which is modelled using 1600 linear elements. In order to avoid spurious pressure waves which may occur if a high pressure is applied from the start of the simulation due to impacting on the solid tube, we apply a smooth input pressure wave $p(t)$ $\left[\text{dynes/cm}^2\right]$ at the inlet in the form

$$p(t) = \begin{cases} 1.3332 \times 10^4 \left(\left(\sin\left(\frac{2\pi t}{0.003} + \frac{3}{2}\pi\right)+1\right)/2\right) & \text{if } t < 0.003 \\ 0 & \text{if } t \geq 0.003. \end{cases} \tag{5.39}$$

The time step size for the simulation is $\Delta t = 0.0001$s, where the convergence behaviour is studied for convergence tolerances ϵ of 10^{-7} and 10^{-9}. The pressure pulse propagation at time steps 0.003s and 0.005s is shown in Figure 5.8.

Of the problems analysed thus far, the flexible tube benchmark represents a problem where the benefit of retaining many observations from multiple time steps is significant. Consider the summarised results in Table 5.3. The IBQN-LS(q) method, while capable of producing good convergence behaviour, now requires the retention of interface observations from close to $q = 30$ time steps (in contrast to approximately only 1 required for the flexible beam and 1D piston-channel problems). While the performance of the IBQN-LS(q) method is relatively uniform from a choice of $q = 5$ onwards, the performance when compared to the MVQN method ranges all the way from 15 to 155 percent more expensive depending on the choice of q.

While retaining more information is beneficial, retaining too many vectors (as is

the case for $q = 40$ for $\Delta t = 10^{-4}$ and $\epsilon = 10^{-9}$) results in an ill-conditioned system Jacobian due to linearly dependent vectors. The only reason the IBQN-LS(q) is capable of providing convergent results for $q \geq 30$ (where the condition number is already very high) is due to QR factorization. Solving the inverses in equation (5.29) using singular value decomposition (SVD) or proper orthogonal decomposition (POD) would enable the retention of all the time step information. For the flexible tube problem, retaining all the information does in fact provide noticeable benefit (illustrated in the Chapter to follow). While SVD or POD solves issues surrounding ill-conditioning, it resolves little of the problems pertaining to an over-defined system or the presence of contradictory information from multiple time steps (as was encountered in the problems of Section 5.5.1 and 5.5.2).

The convergence rates for the MVQN method are shown in Figure 5.9. In the first time step, the convergence behaviour is not monotonically decreasing (though this can be improved by modifying the relaxation parameter ω in the first iteration of time step 1). However, as the Jacobian is trained through the course of the simulation the convergence behaviour improves considerably and from time step 2 onwards is comparable to an inexact-Newton scheme. The convergence behaviour of the MVQN method can in fact be likened to the typical convergence behaviours reported by Dettmer *et al.* [32] when selected chain rule terms were omitted from the computation of the exact cross-coupling terms for partitioned FSI problems. This is especially noteworthy considering the quasi-Newton method(s) investigated here allow for the re-use of partitioned black-box field solvers and do not require any exact gradient computations.

5.5.4 Flexible Tube, $\rho_s = 0.12\mathbf{g/cm^3}$ and $\rho_f = 1.0\mathbf{g/cm^3}$

Let us now reconsider the pressure wave propagation through the flexible tube described above, where the solid density is reduced to $\rho_s = 0.12\text{g/cm}^3$. The fluid density is now almost an order of magnitude higher than the solid density, with the result that the added mass effect is now substantially increased. This particular form of the flexible tube problem was used in [73], as an example to illustrate the limitations of partitioned, and more specifically black-box partitioned coupling schemes. A summary of the coupling results is shown in Table 5.4. Of the four coupling methods, only the MVQN and IBQN-LS methods were capable of producing convergent results, with the MVQN

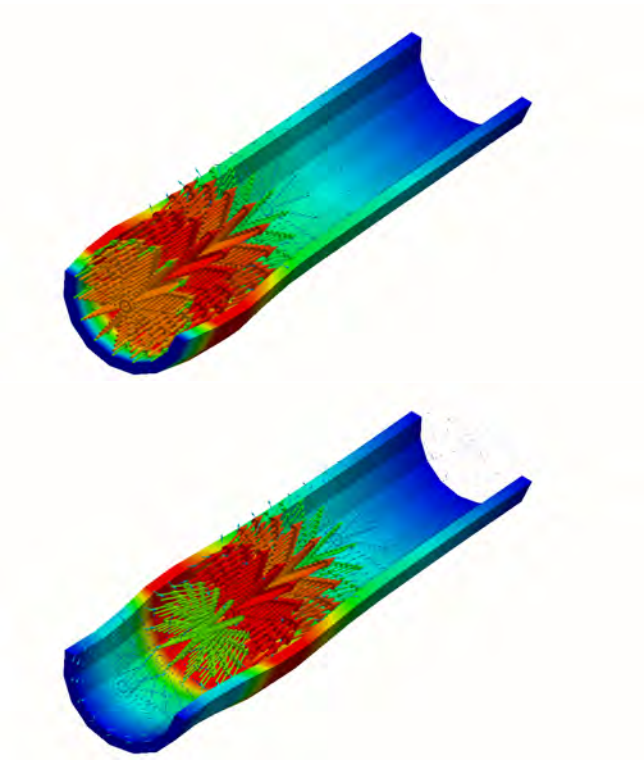


Figure 5.8: Pressure pulse propagation at 0.003 and 0.005 seconds. Wall displacement amplified 10 times.

Table 5.3: Comparison of the number of iterations and relative computational time for pressure propagation through flexible tube where $\rho_s = 1.2\text{g/cm}^3$ and $\rho_f = 1.0\text{g/cm}^3$, for different convergence criterion ϵ. The notation IBQN-LS(q) indicates information from q preceding time steps are retained. If a scheme is non-convergent, the diverging time step as a ratio of the total required time steps is indicated in the CPU column.

	$\Delta t = 10^{-4}, \epsilon = 10^{-7}$				$\Delta t = 10^{-4}, \epsilon = 10^{-9}$			
	Mean	Min	Max	CPU	Mean	Min	Max	CPU
MVQN	4.13	3	7	1.00	5.34	4	10	1.00
Aitken	35.39	12	51	7.33	51.76	29	74	8.42
Broyden	4.83	4	12	1.15	6.58	5	14	1.22
IBQN-LS(0)	10.4	7	11	2.55	13.75	10	15	2.60
IBQN-LS(1)	8.52	3	10	2.11	11.58	4	13	2.15
IBQN-LS(5)	5.96	3	7	1.44	8.5	4	10	1.57
IBQN-LS(10)	5.08	3	7	1.23	7.2	4	10	1.34
IBQN-LS(20)	4.89	3	10	1.24	6.41	4	10	1.20
IBQN-LS(30)	4.71	3	10	1.15	6.40	4	10	1.26
IBQN-LS(40)	4.98	3	10	1.24	8.35	4	24	(51/100)

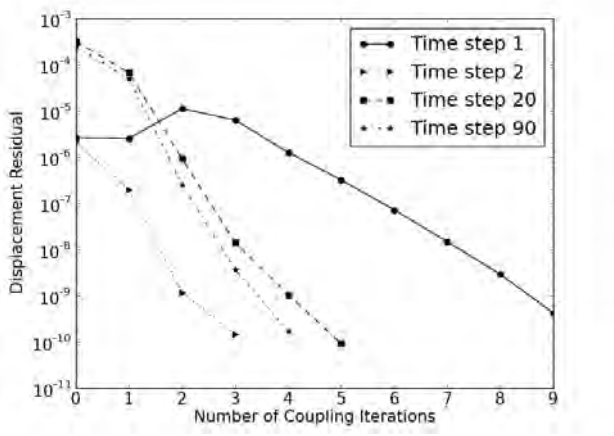


Figure 5.9: Typical MVQN convergence rates for the flexible tube problem with $\rho_S = 1.2\text{g/cm}^3$ and $\rho_f = 1.0\text{g/cm}^3$ for $\Delta t = 10^{-4}$s and $\epsilon = 10^{-9}$ at different times.

method again proving to be the most efficient. Despite the increase in problem complexity, the MVQN method only requires, on average, an additional 1.2 iterations for convergence when compared to the flexible tube with a solid density of $\rho_s = 1.2\text{g/cm}^3$. If we further compare the typical convergence behaviour for the MVQN method in Figure 5.10 (for $\rho_s = 0.12\text{g/cm}^3$) to Figure 5.9 (for $\rho = 1.2\text{g/cm}^3$), we observe that the MVQN approximate Jacobians do take longer to train. The resulting Jacobians again exhibit Newton-like convergence behaviour towards the end of the simulation.

While the solution of this problem does not constitute sufficient proof that the MVQN coupling method can be applied to all haemodynamic problems, it does provide some confidence in the applicability of the method to solve strongly coupled FSI problems. It bears further merit when one considers the possibility of accelerating the training of the approximate Jacobian in the first couple of time steps using multi-grid like methods [29].

5.6 Conclusion

In this chapter we introduced a new multi-vector quasi-Newton method for the implicit coupling of partitioned solvers, which facilitates the use of black-box field solvers. We analysed the method across a wide range of strongly coupled FSI benchmark problems for which the method was demonstrated to be both robust and efficient. The method

Table 5.4: Comparison of the number of iterations and relative computational time for pressure propagation through flexible tube where $\rho_s = 0.12\text{g/cm}^3$ and $\rho_f = 1.0\text{g/cm}^3$, for different convergence criterion ϵ. The notation IBQN-LS(q) indicates information from q preceding time steps are retained. If a scheme is non-convergent, the diverging time step as a ratio of the total required time steps is indicated in the CPU column.

	$\Delta t = 10^{-4}, \epsilon = 10^{-7}$				$\Delta t = 10^{-4}, \epsilon = 10^{-9}$			
	Mean	Min	Max	CPU	Mean	Min	Max	CPU
MVQN	5.38	4	11	1.00	6.58	5	14	1.00
Aitken	71.54	49	90	(13/100)	227	227	227	(2/100)
Broyden	–	–	–	(1/100)	–	–	–	(1/100)
IBQN-LS(0)	17.63	11	22	(16/100)	25.60	12	29	(26/100)
IBQN-LS(1)	16.50	3	19	3.09	22.51	12	25	3.46
IBQN-LS(5)	10.66	3	16	2.01	14.35	9	25	2.22
IBQN-LS(10)	8.46	3	16	1.58	10.66	8	25	1.66
IBQN-LS(20)	6.37	3	16	1.20	7.90	5	25	1.23
IBQN-LS(30)	5.64	3	16	1.08	7.98	5	25	1.27
IBQN-LS(40)	5.83	3	16	1.13	16.56	6	49	(48/50)

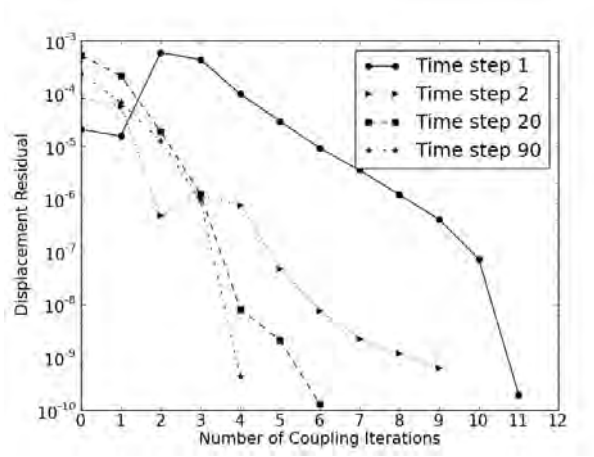


Figure 5.10: Typical MVQN convergence rates for the flexible tube problem with $\rho_S = 0.12\text{g/cm}^3$ and $\rho_f = 1.0\text{g/cm}^3$ for $\Delta t = 10^{-4}$s and $\epsilon = 10^{-9}$ at different times.

outperformed other well known quasi-Newton methods and can be applied to incompressible FSI problems with high added mass ratios. We compared the method to the least squares, reduced order modelling quasi-Newton method of Vierendeels and Degroote *et al.* We demonstrated that the IBQN-LS method can be reformulated as finding the minimum norm Jacobian which satisfies the interface secant equations. This in turn, to some extent explains the method's problem dependent behaviour with regards to the choice of the number of time histories, q, to be retained. The new MVQN updating scheme outperformed the IBQN-LS method for all the problems analysed as well as Broyden's method, which in turn showed favourable performance when compared to the IBQN-LS method; Broyden's method was however unable to obtain a convergent solution when the solid density was chosen to be lower than the fluid density. We have further demonstrated, while fixed point iterations with dynamic relaxation is easy to implement, that it is predominantly ill-suited for the coupling of FSI problems with high added mass ratios; this confirms the earlier findings of many researchers.

Chapter 6

Proposed POD Augmentation of IBQN-LS

6.1 Introduction

In this section we propose an augmentation of the IBQN-LS method via proper orthogonal decomposition (POD). It was shown in Chapter 5 that the overall performance of the IBQN-LS method was strongly linked to the choice of the number of retained historical observations. The optimal choice of q was entirely problem dependent: choosing q too large leads to the gradual deterioration of the convergence properties and eventual failure; equally, choosing q too small leads either to failure to converge or suboptimal convergence rates. This inherent dependence on q was explained in Section 5.4 to be directly linked to very the nature of the method, which is a minimum norm approximation method. As a result, there are two primary problems in relation to the choice of q. The first is what we coined as "contradictory" information, where the minimum norm method provides equal emphasis to all observed (and retained) information. If these pieces of information fall along the same search directions but provide opposing information, the IBQN-LS method degenerates to a least squares fit of said information. This in turn leads to a gradual decrease in the method's coupling performance. The second problem stems from linear dependence. As the number of observation vectors increase, so does the possibility that two or more vectors become linearly dependent, in which case the conditioning of the matrices deteriorates.

Other than limiting the number of retained vectors, there is no solution to contradictory information. On the other hand, linear dependence manifests in ill-conditioned matrices. Ill-conditioning, and the loss of accuracy which results there from can be addressed, and we do so here by proposing an augmentation of the IBQN-LS method via the use of POD. POD, also known as Karhunen-Loève decomposition, principal component analysis and singular value decomposition, provably forms the optimal linear decomposition of any given set of system observations [53]. It therefore allows us a mathematically quantifiable means by which to truncate the information such that we retain essentially "all" the information, while limiting the extent of the ill-conditioning.

6.2 POD and the Proposed IBQN-LS Augmentation

To facilitate the discussion on the POD augmentation, we start with the "method of snapshots", a solution method introduced by Sirovich [92]. The method of snapshots is a truncated solution procedure to solve for the POD basis functions for a non-square observation matrix. Let us consider the observation matrix introduced in equation (5.12), reproduced here for convenience

$$\Delta \boldsymbol{D}_F^{n+1} = \left[\boldsymbol{d}_k - \boldsymbol{d}_{k-1}, \boldsymbol{d}_{k-1} - \boldsymbol{d}_{k-2}, ..., \boldsymbol{d}_2 - \boldsymbol{d}_1\right]. \tag{6.1}$$

The matrix is of size $n \times m$, where n is the number of observation vectors and m is the DOFs of the system (or number of DOFs along the interface boundary). The method of snapshots requires the solution of the eigenvalue problem of an $n \times n$ autocorrelation matrix of the form:

$$\boldsymbol{\mathcal{R}} = \frac{1}{n} \left[\Delta \boldsymbol{D}\right] \left[\Delta \boldsymbol{D}\right]^T, \tag{6.2}$$

where the eigenvectors $\boldsymbol{a}$ of $\boldsymbol{\mathcal{R}}$ are computed as an intermediate step to computing the basis modes, i.e.

$$\boldsymbol{\mathcal{R}}\boldsymbol{a} = \boldsymbol{\lambda}\boldsymbol{a}. \tag{6.3}$$

The POD basis modes are then computed as the linear projection

$$\boldsymbol{\varphi} = \boldsymbol{a}\Delta\boldsymbol{D}. \tag{6.4}$$

The eigenvalues λ_i in (6.3) provide an indication as to how much of the system information is contained in the corresponding POD modes $\boldsymbol{\varphi}_i$. Therefore by ordering the eigenmodes $\boldsymbol{\varphi_i}$ based on the ranked eigenvalues (from largest to smallest), it is possible to truncate the system to only the c most dominant basis modes, that is,

$$\Delta\boldsymbol{d}^j \approx \sum_{i=1}^{c} \alpha_i^j \boldsymbol{\varphi_i} \tag{6.5}$$

where c is the number of retained basis modes, $\Delta\boldsymbol{d}^j$ is an approximation of the j^{th} vector of the observation matrix $[\Delta\boldsymbol{D}]$ and $\boldsymbol{\alpha}_i$ is the i^{th} eigenvector of $\boldsymbol{a}$. Similarly, the eigenvector matrix $\boldsymbol{a}$ contains the expansion coefficients to project the basis modes back into real space. As such the original observation matrix can be exactly reproduced by

$$\Delta\boldsymbol{D} = \boldsymbol{a}\boldsymbol{\varphi}, \tag{6.6}$$

if the full matrices $\boldsymbol{a}$ and $\boldsymbol{\varphi}$ are used. Alternately, the $[\Delta\boldsymbol{D}]$ may be approximated, by truncating $\boldsymbol{a}$ and $\boldsymbol{\varphi}$ to the c most dominant basis modes only. Therefore, in exactly the same way as described in Section 5.4, we can now construct our IBQN-LS approximate Jacobian as:

$$\boldsymbol{J}_F = [\Delta\boldsymbol{F}]\,\boldsymbol{a}\left[\left(\boldsymbol{\varphi}^T\boldsymbol{\varphi}\right)^{-1}\boldsymbol{\varphi}^T\right], \tag{6.7}$$

with the primary distinction to the Jacobian approximation given in equation (5.29) is that we now have to ability to directly truncate $\boldsymbol{a}$ and $\boldsymbol{\varphi}$ in accordance to the ranked eigenvalues λ. By retaining only the c most dominant modes in the reconstruction, the condition number of the least square solution in (6.7) of $\left(\boldsymbol{\varphi}^T\boldsymbol{\varphi}\right)^{-1}\boldsymbol{\varphi}^T$ is then provably equal to $\sqrt{\frac{\lambda_1}{\lambda_c}}$, where λ_1 and λ_c are the eigenvalues related to the largest and smallest retained basis modes. In so doing we can directly curtail the conditioning, while retaining the most dominant pieces of information. Percentage wise, this can be expressed as retaining $\frac{\lambda_1-\lambda_c}{\lambda_1}$ of the information in the original observation matrix. In a similar fashion, we can compute a POD based approximation for $\boldsymbol{J}_S$.

6.3 Analysis via Flexible Tube Problem

Let us return to the flexible tube problem with a propagating pressure wave, analysed in Section 5.5.3. The problem is ideal for demonstrating the potential advantage of the proposed POD augmentation. Not only is the problem strongly coupled with high added mass (therefore difficult from an FSI perspective), but the very nature of the boundary deformation does not lend itself to contradictory information. The primary reason for history, q, dependence of the IBQN-LS applied to this problem is then primarily due to linear dependence when q becomes too large. The overall problem description remains the same. Here we solve for a displacement convergence tolerance of $\epsilon = 10^{-7}$, and transfer pressure only (neglecting shear stresses). The reader at this point is reminded that the linear systems for IBQN-LS method, without POD, are solved via matrix factorisation (important to mention only because the factorisation, and subsequent dual solution in effect square roots the conditioning of the matrix, allowing for very large condition numbers before the effects of ill-conditioning cause a loss of numerical accuracy).

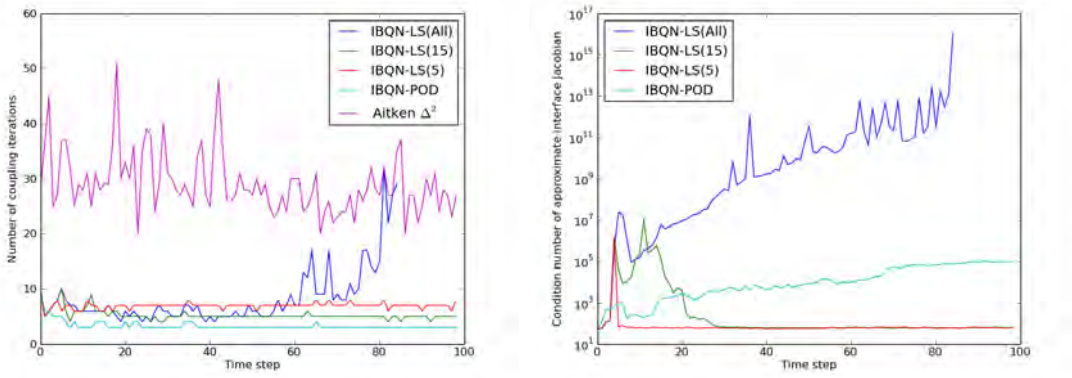
For the IBQN-POD, we select the c most dominant modes based on the criteria

$$\frac{\lambda_1}{\lambda_c} \leq 10^{11}, \tag{6.8}$$

which leads to a maximum condition number of 3.16×10^5. For the IBQN-POD coupling scheme we construct approximations of $\boldsymbol{J}_S$ and $\boldsymbol{J}_F$ by retaining all the observations, and as before IBQN-LS(q) is left to indicate that observations from only the q most recent time steps are retained.

In Figure 6.1 a plot of the number of coupling iterations as well as the associated matrix condition number is shown. The condition number is for the interface matrix $\left[\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{f}}\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{d}} - \boldsymbol{I}\right]$ given in equation (5.9). As can be observed, the IBQN-LS method incrementally improves, as q is increased, up until the eventual failure when the condition number becomes too high. Directly in contrast the IBQN-POD method remains stable with an average of 3.31 coupling iterations. The rapid and continued learning of the IBQN-POD system is illustrated by the convergence rates at different time steps, shown here in Figure 6.2.

To better illustrate, given increasing time step information, how many of the observations are in fact linearly dependent, we outline a comparison of the total observations



(a) (b)

Figure 6.1: (a) Comparison of the number of solver iterations, included here is the coupling performance of Aitken's method for comparison, with (b) the associated interface matrix, $[\boldsymbol{J}_S\boldsymbol{J}_F - \boldsymbol{I}]$, condition number.

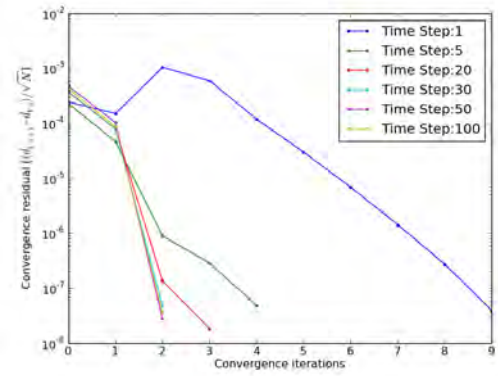


Figure 6.2: Displacement convergence rates for the IBQN-POD coupling scheme at different time steps. Notice the comparatively rapid "learning" with increasing retention of additional time observations.

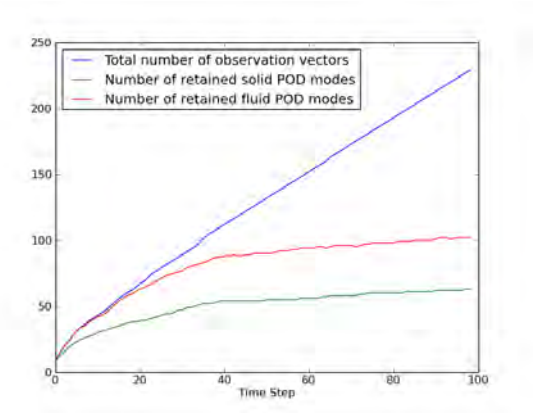


Figure 6.3: A comparison of the total number of discrete differencing vectors accumulated over the course of the simulation to the number of retained solid and fluid space basis modes in the construction of $\boldsymbol{J}_S$ and $\boldsymbol{J}_F$.

for the complete simulation to the total number of retained basis modes in Figure 6.3. Of the nearly 230 discrete differencing observations accumulated through the course of the simulation, in the end there were only approximately 100 (mostly) linearly independent fluid observations and approximately 65 solid observations.

6.4 Critical analysis of the POD augmentation

POD provides an elegant mathematical framework to limit the effects of linear dependence amongst the discrete differencing interface observation vectors. The POD augmentation is in effect a pseudo-inverse, where one is given direct control over the extent of allowed ill-conditioning. The dominant additional cost of the proposed IBQN-POD model is the solution of an $n \times n$ eigenvalue problem and the construction of the covariance matrix $\boldsymbol{\mathcal{R}}$ with associated cost $m \times n^2$. In typical problems, the number of retained observation $n \ll m$, but n can quickly grow to non-trivial values. In such cases, the cost of the eigenvalue problem can be somewhat mitigated using vector iterative methods.

Truncating the observation matrices using POD is a useful means by which to limit the effects of ill-conditioning. The dependency on q of the IBQN-LS method is therefore partially reduced. There are however obvious limitations to the proposed POD use. Even with the inclusion of POD, it is not possible for all problems to retain all vectors,

either due to large time envelopes or small time steps. The additional computational cost may become prohibitively large, and the POD method provides no solution for contradictory information. As such, the IBQN-POD method still requires limiting the total number of observation vectors. Since the POD subspace information cannot be used to ascertain which of the real spaced vectors are linearly dependent, and therefore safely discarded, the method still requires an upfront decision on an appropriate choice of q. The POD augmentation is perhaps interesting should an environment exist where the IBQN-LS method is already implemented. The modification is fairly easy to implement and does decrease the likelihood of sudden divergence if q is inadvertently chosen too large.

Overall though, the MVQN method still remains a preferred coupling approach. The MVQN method also allows, in a minimum norm sense, that all observations and accumulated information be retained. Furthermore, the unique input-output mapping of the fluid and solid problems, provides with certainty that contradictions in the minimum norm solutions are not possible. Similarly, given the fact that the minimum norm of only the vectors in the current time steps are needed, the possibility of ill-conditioning is significantly reduced. Of course, the POD augmentation can also be applied to the MVQN method. This would with mathematical certainty prevent ill-conditioning, while the additional POD costs are naturally limited, since n is then limited to only the number of iterations in the current time step (k). It may be noted though, that conditioning has not been an issue in any of the problems analysed in this thesis (both in the previous chapter and the chapters still to follow), thus the need for a pseudo-inverse like method in conjunction with the MVQN method is not clear.

Chapter 7

Fully Enclosed Partitioned FSI

7.1 Introduction

In this chapter we introduce the idea of combining artificial compressibility (AC) with quasi-Newton (QN) methods to solve strongly coupled, fully/quasi-enclosed fluid-structure interaction (FSI) problems. Partitioned, incompressible, FSI, based on Dirichlet-Neumann domain decomposition solution schemes cannot be applied to problems where the fluid domain is fully enclosed. An incompressible fluid domain with a prescribed inlet flow and fully enclosed by a Dirichlet type interface manifests itself as an ill-posed problem and has been described by Küttler *et al.* [72] as the "incompressibility dilemma".

In this context, artificial compressibility (AC) is a useful method by which the incompressibility constraint can be relaxed by including a source term within the fluid continuity equation. The attractiveness of AC stems from the fact that this source term can readily be added to almost any fluid field solver, including most commercial solvers. Once included, both the modified fluid solver and structural solver can then continued to be treated as "black-box" field operators.

AC was originally introduced in the context of FSI as a means by which to remove some of the instabilities related to strongly coupled, incompressible problems [85, 88]. AC/FSI allows for a linearised approximation of the solid domain's stress-displacement relationship to be included within the fluid set of equations. The solid domains response is based on an approximation of the solid domain's volume change in response to a change in interface pressure [24, 28, 60]. AC/FSI is therefore ideally suited to

problems that are strongly governed by such a relationship, see for example [28, 60], where AC/FSI is applied to the simulation of a propagating pressure pulse in arteries.

There are however many problems to which such a relationship is not well suited, for example flow around a flexible beam [24], or problems with large structure inertial forces. In this chapter we therefore pose the possibility of combining AC/FSI and QN methods. The AC/FSI allows for the solution of fully-enclosed problems, where the additional QN sensitivities would then further stabilise the method.

We will demonstrate the performance of the proposed coupling scheme via numerical experiments, for both fully enclosed balloon inflation type problems and a well known benchmark problem of flow induced oscillations of a flexible beam. We will demonstrate that the proposed combination of AC/FSI+QN methods not only improves on the performance of AC but can at times yield near-quadratic convergence rates. The chapter will initially introduce AC/FSI, including the interface test load method to approximate the AC coefficients. The application of QN to AC/FSI is then outlined, and the chapter concludes with performance evaluations via numerical examples.

7.2 Artificial Compressibility (AC)

Artificial compressibility in the context of fluid structure interactions (AC/FSI) was originally introduced by Riemslagh *et al.* [88] in a 1D FSI framework as a means by which to stabilise the coupling between incompressible fluids and elastic walls. The method has since been applied to more complex problems [28, 60, 85]. AC/FSI can be viewed as a variation of the original AC algorithm first proposed in [17] as a means to couple the momentum and continuity equations and in turn stabilise incompressible Navier-Stokes equations. With regards to AC/FSI, a pressure time derivative is inserted into the continuity equation, where this additional term is based on a physical approximation of the pressure-volume response of the structure. Not only does this additional term have a stabilising effect on the non-linear FSI equations, but it also serves as a means by which to resolve the incompressibility dilemma.

In order to introduce the AC modifications, let us re-introduce our fluid equations here which form the basis of our fluid interface mapping operator $\boldsymbol{F}$. The incompressible Navier-Stokes equations, in an Arbitrary Lagrangian-Eulerian (ALE) framework that accounts for the moving fluid domain, is given by

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} - \boldsymbol{u}_m) \cdot \nabla \boldsymbol{u} + \nabla p - \nu \nabla^2 \boldsymbol{u} = \boldsymbol{b}, \tag{7.1}$$

$$\nabla \cdot \boldsymbol{u} = \boldsymbol{0}. \tag{7.2}$$

Here $\boldsymbol{u}$ is the fluid velocity, $\boldsymbol{u}_m$ the ALE-coordinate system velocity at a given reference position, p the kinematic pressure, ν the kinematic viscosity and $\boldsymbol{b}$ accounts for the body forces.

The basic idea of artificial compressibility applied to FSI, is to modify the continuity equation (7.2) by inserting a pressure time derivative such that

$$\beta \left(\frac{\partial p}{\partial t} \right) + \nabla \cdot \boldsymbol{u} = \boldsymbol{0}, \tag{7.3}$$

where β is the artificial compressibility coefficient. Following the work of Råback *et al.* [85], (7.3) is equivalent to allowing the fluid density to vary with pressure such that $d\rho/\rho = \beta dp$. Inserting the pressure density relationship into the compressible flow continuity equation $\partial\rho/\partial t + \nabla \cdot \boldsymbol{u} = 0$, and neglecting the spatial derivative of ρ yields a continuity equation of the form given in (7.3).

Equation (7.3) does not require additional pseudo-iterations to solve as suggested by [72]. Instead the solution can be viewed as an iteration trick, where the time derivative of pressure is approximated in a given iteration via

$$\frac{\beta \left(p_k^{n+1} - p_{k-1}^{n+1} \right)}{\Delta t} + \nabla \cdot \boldsymbol{u}_k^{n+1} = \boldsymbol{0}, \tag{7.4}$$

where subscript k represents the current FSI coupling iteration in time step $n+1$ and Δt is the simulation time step size. Therefore, while the incompressible continuity equation (7.2) is initially violated, at convergence for time step $n+1$, the AC compressibility term disappears as $p_k^{n+1} \to p_{k-1}^{n+1}$, thereby satisfying the original continuity equation.

The AC/FSI method relies heavily on an optimal choice of β. Assuming the structure to be defined by a linear set of equations, it would be possible to define a linear relationship between the fluid domain volume change ΔV related to a corresponding

interface pressure change Δp such that

$$\frac{\Delta V}{V} = \beta \Delta p. \tag{7.5}$$

Using the relation (7.5), it is then possible to construct an analytical expression for β (see for example [85] for flow through a flexible tube). Analytical expressions are however limited to simple geometries. To allow for the AC/FSI method to be generally applicable to more complex geometries, both Järvinen *et al.* [60] and Degroote *et al.* [28] have suggested using an interface test load method.

Given two different pressures p_1 and p_0 applied along the wetted interface Γ_{FSI}, two distinct volumes V_1 and V_0 would be obtained. β may then be approximated by a finite difference approximation

$$\beta = \frac{1}{V_0} \frac{(V_1 - V_0)}{(p_1 - p_0)}. \tag{7.6}$$

While the approximation (7.6) suggests application of AC throughout the entire enclosed domain, AC can equally be applied along the interface only. Interface-artificial-compressibility (IAC), following the work of Degroote *et al.* [28] limits AC to the elements adjacent to the interface. While AC throughout the domain significantly simplifies the numerical properties of the Navier-Stokes equations, IAC has been demonstrated to produce slightly superior FSI convergence behaviour [11]. In fact, it is possible to draw some parallel similarities between IAC/FSI and Robin boundary conditions, where the fluid equations along the interface are augmented with an approximation of the solid domain sensitivities [24]. For the purposes of this study we will exclusively use IAC.

The IAC approximation of $\boldsymbol{\beta}$ requires the projected volume change of the interface elements. For small deformations, a numerically convenient way in which to approximate $\boldsymbol{\beta}$ based on this projected volume change is

$$\boldsymbol{\beta} = \frac{\Delta \boldsymbol{L}}{\boldsymbol{h}} \frac{1}{\Delta p}, \tag{7.7}$$

where $\Delta \boldsymbol{L} = \Delta \boldsymbol{d}_\Gamma \cdot \boldsymbol{n}_\Gamma$, the perpendicular change in interface displacement. $\boldsymbol{h}$ is the equivalent fluid element height which can be computed as any given element's interface

volume to area ratio in 3D (or area to length ratio in 2D)

$$h_i = \frac{V_i}{A_{\Gamma,i}}, \quad \text{for } i = 1, 2, ..., N_\Gamma.$$

It is possible that $\boldsymbol{\beta}$ may be computed as a constant (or averaged value). We however treat $\boldsymbol{\beta}$ as a non-uniform field allowing for better approximations for solid domains that have either varying structural properties or geometrically varying volume-pressure relationships.

It should be noted that the choices for p_1 and p_0 have little effect on the computed approximation of $\boldsymbol{\beta}$. In order to allow for a fair comparison of AC, the coefficient $\boldsymbol{\beta}$ is recomputed at the start of each time step. If the normal interface displacements are appropriately transferred to the fluid interface, then (7.7) can be computed using two structural solver calls only, as opposed to two full FSI iterations. Furthermore, while not demonstrated in the results to follow, when AC/FSI is combined with QN methods, there is very little difference in performance of AC/FSI+QN if $\boldsymbol{\beta}$ is adaptively updated each time step or computed once-off at the start of the simulation (and remains static for the duration of the simulation).

As a final note, shear stresses are ignored within the IAC approximation, but not in the actual simulation. This can be done without overly affecting the stabilising effect of AC because pressure forces along the interface, for most FSI problems, dominate (shear forces typically account in the order of 1 percent of the total interface forces). AC/FSI is therefore unsuited to problems that are shear driven.

7.3 IAC/FSI + quasi-Newton

IAC is a useful way in which the fluid equations can be modified to allow for the solution of fully/quasi-enclosed problems. The method is however largely only applicable to problems where the pressure-volume relationship bears relevance to the nature of the coupled problem. Furthermore, while the test load method provides a means by which to compute the optimal (or near optimal) value for the AC coefficient, AC/FSI includes only a linearised approximation of the structural system behaviour. Interrogating equations (7.3) and (7.7) it may be noted that AC/FSI and the test load method does not for example account for the structural inertial terms. Therefore, the ability to

compute what is demonstrably the near optimal AC coefficient does not guarantee that AC/FSI will not produce sub-optimal convergence behaviour for certain problems.

In this section we propose augmenting the IAC/FSI method through the use of quasi-Newton (QN) methods. The IAC terms included in the fluid equations are an approximation of the solid domain's sensitivities w.r.t. fluid interface pressures. IAC in isolation however ignores the additional fluid sensitivities with respect to the domain displacements. The aim therefore of the additional QN terms is to include these sensitivities. We will show that not only will this reduce the sensitivity to the computed values of $\boldsymbol{\beta}$, but will also provide improved convergence behaviour of IAC/FSI, thus potentially allowing IAC to be applied to a much larger range of problems.

In order to describe the QN method, let use first rewrite our fluid system mapping operator (5.2) as a root finding problem, such that

$$\boldsymbol{r}_F = \boldsymbol{F}_{AC}\left(\boldsymbol{d}\left(\boldsymbol{f}\right)\right) - \boldsymbol{f} = \boldsymbol{0}, \tag{7.8}$$

where $\boldsymbol{F}_{AC}$ is now the fluid interface operator including the AC terms. We can now define a Newton system such that

$$\left[\begin{array}{cc} \dfrac{\partial \boldsymbol{F}_{AC}}{\partial \boldsymbol{d}}\dfrac{\partial \boldsymbol{d}}{\partial \boldsymbol{f}} - \boldsymbol{I} & \dfrac{\partial \boldsymbol{F}_{AC}}{\partial \boldsymbol{d}} \end{array}\right] \left\{\begin{array}{c} \Delta \boldsymbol{f} \\ \Delta \boldsymbol{d} \end{array}\right\} = \left[-\boldsymbol{F}_{AC}\left(\boldsymbol{d}\left(\boldsymbol{f}\right)\right) + \boldsymbol{f}\right], \tag{7.9}$$

which re-arranged results in a block-Newton equation to solve for the fluid force update

$$\left[\frac{\partial \boldsymbol{F}_{AC}}{\partial \boldsymbol{d}}\frac{\partial \boldsymbol{d}}{\partial \boldsymbol{f}} - \boldsymbol{I}\right] \Delta \boldsymbol{f} = -\left(\boldsymbol{F}_{AC}\left(\boldsymbol{d}\left(\boldsymbol{f}\right)\right) - \boldsymbol{f}\right) - \frac{\partial \boldsymbol{F}_{AC}}{\partial \boldsymbol{d}}\Delta \boldsymbol{d}. \tag{7.10}$$

An update for the interface tractions is then found by

$$\boldsymbol{f}_{k+1}^{n+1} = \boldsymbol{f}_{k}^{n+1} + \Delta \boldsymbol{f}. \tag{7.11}$$

The updated interface tractions $\boldsymbol{f}_{k+1}^{n+1}$ is then transferred to the solid interface solver, which is now based on the additional sensitivities of the fluid solver.

It is important to note here, because of the QN update step, that the starting pressure for the modified AC equations be updated as well. Recall from equation (7.4)

that the modified fluid continuity equation is now

$$\frac{\boldsymbol{\beta}\left(p_k^{n+1} - p_{k-1}^{n+1}\right)}{\Delta t} + \nabla \cdot \boldsymbol{u}_k^{n+1} = \mathbf{0}. \tag{7.12}$$

At the start of the next FSI coupling iterate in time step $n+1$, p_{k-1}^{n+1} will have been modified because of the QN update (7.11). p_{k-1}^{n+1} should therefore also be modified based on the pressure component of $\Delta \boldsymbol{f}$ such that

$$p_{k-1,\text{new}}^{n+1} = p_{k-1}^{n+1} + \Delta \boldsymbol{f} \cdot \boldsymbol{n}_\Gamma. \tag{7.13}$$

Without this pressure correction a Newton-like method in conjunction with AC/FSI cannot work.

In order to solve the block-Newton system described by equation (7.10), we now wish to construct approximations of the fluid and solid system Jacobians $\boldsymbol{J}_F = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{d}} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{d}}$ and $\boldsymbol{J}_S = \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{f}} = \frac{\partial \boldsymbol{d}}{\partial \boldsymbol{f}}$. To do so, we make use of the newly introduced MVQN method, which constructs approximations of the coupling Jacobians using only observations of the interface displacements and stresses.

There is one potential difference in the construction of $\boldsymbol{J}_F$ compared to the originally introduced MVQN method in Section 5.4.1. The displacement vectors in the fluid solver displacement observation matrix,

$$\Delta \boldsymbol{D}_F^{n+1} = \left[\boldsymbol{d}_{F,k-1} - \boldsymbol{d}_{F,k}, \boldsymbol{d}_{F,k-2} - \boldsymbol{d}_{F,k}, ..., \boldsymbol{d}_{F,1} - \boldsymbol{d}_{F,k}\right],$$

should be the displacements received as output from the solid field solver transferred to the fluid interface. However, because of artificial compressibility, there is in fact now an additional "perceived" displacement field in addition to the true boundary displacement. This additional displacement field can be included to the fluid displacement vectors such that

$$\boldsymbol{d}_{F,k} = \boldsymbol{d}_{F,k} + \boldsymbol{u}_{AC}, \tag{7.14}$$

where $\boldsymbol{u}_{AC}$ incorporates the normal displacement accounted for by AC. The $\boldsymbol{u}_{AC}$ term may then be approximated by

$$\boldsymbol{u}_{AC,k,i}^{n+1} = \left(\beta_i \left(p_{F,k,i}^{n+1} - p_{F,k-1,i}^{n+1}\right) h_i\right) \cdot \boldsymbol{n}_{\Gamma,k,i} \text{ for } i = 1, 2, ..., N_\Gamma, \tag{7.15}$$

where N_Γ is the number of elements along the interface. The additional $\boldsymbol{u}_{AC}$ includes the $\boldsymbol{\beta}$ dependency into the fluid system quasi-Newton approximation. The term in essence accounts for the fact that $\boldsymbol{F}_{AC}$ is not only dependent on displacement, $\boldsymbol{F}(\boldsymbol{d})$, but is also functionally dependent on the modified starting pressure, i.e. $\boldsymbol{F}_{AC}(\boldsymbol{d}_{S,k}, p_{F,k-1})$. However, since $\boldsymbol{J}_F$ is already only based on an approximation, $\boldsymbol{u}_{AC}$ can be safely ignored and will only minimally affect the overall convergence behaviour. In the numerical examples we will illustrate the effect of including the additional AC displacement field, referred to as option A, where option B will refer to the omission of the $\boldsymbol{u}_{AC}$ term.

These additional $\boldsymbol{u}_{AC}$ displacement terms should not be included in the approximation of the solid interface Jacobian $\boldsymbol{J}_S$, as the solid domain is not a function of the newly included AC terms.

Again, because the MVQN method requires that at least two coupling iterations have been performed, the first iteration of the first time step is a simple fixed point iterate. Importantly, because of the AC we do not require an additional relaxation term. As before, the starting Jacobians are initialised to $\boldsymbol{J}_F^0 = [\boldsymbol{0}]$ and $\boldsymbol{J}_S^0 = [\boldsymbol{0}]$. For the first iteration of subsequent time steps, the approximate Jacobian from the preceding time step is used but the update formulae are only applied from iteration count 2 onwards. A summary of the IAC/FSI+MVQN in pseudo code is included in Algorithm B.1 attached as an appendix.

As a side note, in Chapter 5, we mentioned that for typical DN based partitioned schemes, the system can be expressed as a displacement residual problem (see equation (5.11)). A similar notion here would be to solve for the residual equations of the fluid field forces such that

$$\left[\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{f}}\right]_k \Delta \boldsymbol{f} = -\boldsymbol{r}_k, \tag{7.16}$$

where $\boldsymbol{r}_k = \boldsymbol{f}_k - \boldsymbol{f}_{k-1}$, and $\left[\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{f}}\right]_k$ is the Jacobian for the current coupling iteration. This however is not a feasible alternative when considering the modified AC fluid equations. Because of the AC terms, the fluid interface forces are now directly linked to displacement. The relation between two successive interface tractions (on the basis of directions and magnitudes alone) are therefore insufficient to describe the system behaviour. To better explain this, let us consider the fluid equations without AC. Given a particular displacement $\boldsymbol{d}_{\Gamma,k}^{n+1}$, with the exclusion of the AC term, will always produce the same $\boldsymbol{f}_{\Gamma,k}^{n+1}$. On the other hand, with the inclusion of the AC term, a given displacement

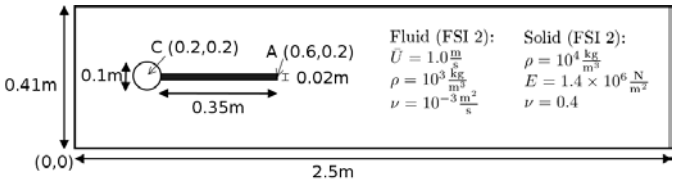


Figure 7.1: 2D Flexible beam problem description .

$\boldsymbol{d}_{\Gamma,k}^{n+1}$ may now yield different interface forces depending on the interface pressure guess from the previous iteration. It is therefore important to include the system sensitivities of forces with respect to displacements and displacements with respect to forces, as described by the block-Newton equation (7.10).

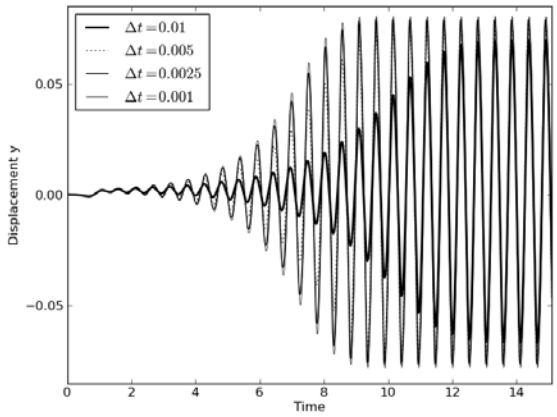
7.4 Test Problems

In this section we analyse artificial compressibility as well as the combination with quasi-Newton methods, via numerical examples.

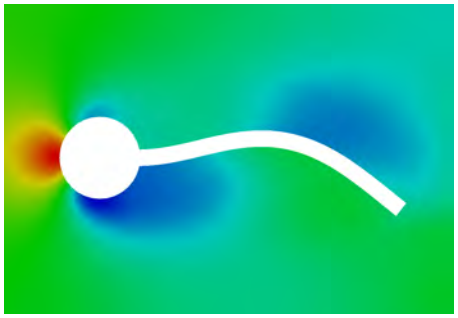
7.4.1 2D Flexible Beam

The first problem we analyse is flow around a fixed cylinder, with an attached flexible beam. The problem was first proposed by Turek *et al.* [105] with substantial numerical verifications. The flexible beam problem, as already mentioned in [24], represents an example of a class of problems for which AC/FSI is not well suited to solve.

The problem layout and material descriptions is provided in Figure 7.1. We only analyse here the transient FSI problem referred to as FSI2 in [105]. FSI2 consists of a 0.02m thick, 0.35m long flexible beam attached to a fixed circular cylinder with 0.1m diameter. The fluid domain has a parabolic inflow with mean velocity of $\bar{U} = 1.0\mathrm{m/s}$. The properties of the problem are such that the flexible beam oscillates as a result of vortex development following the fixed cylinder. The centre of the cylinder is by design constructed to be non-symmetric to remove dependence on numerical errors for the onset of any oscillations. The inlet flow velocity for the transient FSI2 is slowly ramped up for $t < 0.5\mathrm{s}$ via $\left(1 - \cos\left(\pi t/2\right)\right)/2$. The top, bottom, fixed cylinder and flexible beam walls are defined as non-slip boundaries, where a zero reference pressure is prescribed at the domain outlet.



(a)



(b)

Figure 7.2: (a) Beam tip displacement of point A for the flexible beam problem shown for different time steps, along with (b) a snapshot of the beam deformation with pressure contours shown here for time step 12.58s.

The fluid domain is discretised using 1725 linear elements, where the solid domain is defined with a geometrically non-linear, plane strain formulation with 72 full-integration, biquadratic elements. We make use of the comparatively coarse discretisation to allow for the use of comparatively large time step sizes. The problem is solved here for four different time step sizes of $\Delta t = 0.01$s, $\Delta t = 0.005$s, $\Delta t = 0.0025$s and $\Delta t = 0.001$s. A convergence criterion of $\epsilon = 10^{-8}$ is used across all time step sizes. A plot of the beam tip displacement (point A) is shown in Figure 7.2, along with a snapshot of the domain deformation and pressure contours at time step 12.58s.

In Table 7.1 we compare the number of coupling iterations required by each of the coupling schemes as well as a summary of the beam tip displacement for all four time

Table 7.1: Comparison of the minimum, maximum and mean number of iterations required for a convergence tolerance of $\epsilon = 10^{-8}$ for the flexible beam benchmark problem. (-) indicates non-convergence and the beam tip displacement at point A (u_y) is provided in the form of mean±amplitude[frequency] and compared to the reference results of Turek *et al.* [105].

	# Iterations	MVQN Only	IAC/FSI	IAC/FSI + MVQN	$u_y[10^{-3}]$
$\Delta t = 0.01$s	Min	3	–	3	1.44 ± 68.3 [1.9]
	Max	5	–	5	
	Mean	3.28	–	3.63	
$\Delta t = 0.005$s	Min	3	–	3	1.40 ± 74.9 [1.9]
	Max	5	–	5	
	Mean	3.18	–	3.46	
$\Delta t = 0.0025$s	Min	3	3	3	1.31 ± 77.8 [1.9]
	Max	5	129	5	
	Mean	3.27	50.58	3.48	
$\Delta t = 0.001$s	Min	3	3	3	1.20 ± 79.2 [1.9]
	Max	5	14	6	
	Mean	3.32	6.98	3.60	
				Reference [105]	1.23 ± 80.6 [2.0]

step sizes. Because IAC/FSI is so poorly suited to solve this class of problems, and the problem is not fully enclosed, which will enable standard coupling techniques to be used, we additionally compare the coupling performance of using the MVQN method (without AC).

From the results, it is immediately evident that IAC/FSI in isolation is severely limited. For the two coarse time step sizes, the scheme fails to converge entirely. For $\Delta t = 0.0025$s, the relative change in displacement between two time steps is sufficiently small to allow for IAC/FSI to provide convergent results, albeit with a very large number of required iterations. The performance of the IAC/FSI+MVQN coupling scheme compares very well with the coupling behaviour of using only MVQN. Therefore, despite the fact that the AC terms included within the continuity equation, and the additional test load approximation is in no way suited to solve for a flexible beam type problem, the combination of techniques does allow for a comparatively efficient coupling methodology.

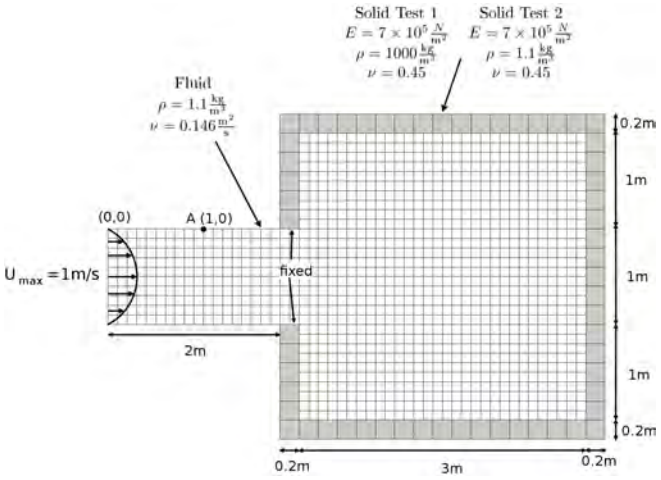


Figure 7.3: Fully enclosed balloon inflation test problem setup.

7.4.2 Balloon Inflation

We analyse here a balloon inflation test problem first introduced by Küttler *et al.* [72]. The problem consists of a fully enclosed fluid domain with an inlet flow velocity. The aim of the test problem is firstly to outline the feasibility, associated performance and the limitations of the IAC method. We then demonstrate, by comparison, the potential merits of IAC in conjunction with quasi-Newton methods.

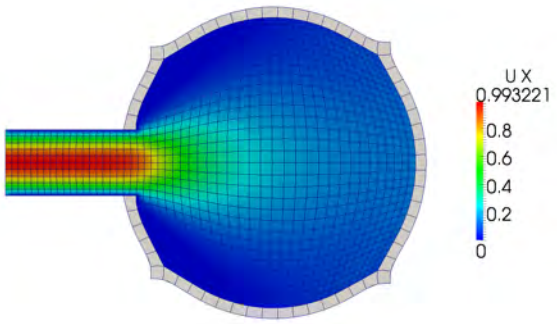
The inflation test problem is outlined in Figure 7.3. The problem consists of a 3m $\times$ 3m internal volume, enclosed by a 0.2m thick shell. A parabolic inflow velocity is applied at the inlet, which is slowly ramped for $t < 1$s from $U_{\max \text{ in}} = 0$ to 1m/s via the sinusoidal curve $\left(\sin\left(\pi\left(t + 3/2\right)\right) + 1\right)/2$ to avoid spurious pressure oscillations which may occur given a large initial inlet flux. The fluid density and kinematic viscosity used for the numerical experiment is $\rho_f = 1.1\text{kg/m}^3$ and $\nu = 0.146\text{m}^2/\text{s}$. The structural domain is described by a geometrically non-linear formulation and is solved using 59 quadratic full integration elements, where the fluid domain is solved using 1100 linear elements. The simulation is solved for a time step size of $\Delta t = 0.05$s.

In order to demonstrate the various associated properties of IAC, we use two different choices of material properties. The first test concerns itself with a comparatively high elastic modulus with a high solid to fluid density ratio. For the second test, likened to the complexity of biomechanics problems, the solid density is reduced to the same as the fluid density. The two sets of solid material properties are provided in Figure 7.3.

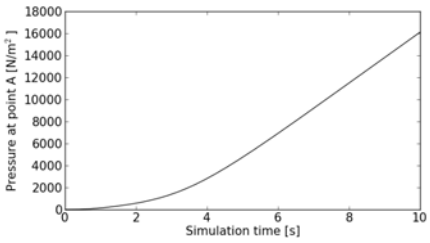
A snapshot of the velocity for Test 1 is shown in Figure 7.4, along with a plot of pressure at point A. It should be noted that, while reducing the density in Test 2 complicates the problem from a coupling perspective, the pressure at point A remains identical to that of Test 1. The internal pressure state is a function of the structural stress state, which for the balloon inflation problem is a function of the elastic properties. The boundary flux error made in each of the time steps is shown in Figure 7.5 (for a convergence tolerance of $\epsilon < 10^{-7}$). The relative boundary flux error can be related directly to the volume error made in each time step, as the flux along the enclosed interface (defined by the deformation of the solid interface), should equal the total flux entering the domain (based here on a prescribed inlet velocity). It is not possible to formulate an analytical solution to the problem, primarily due to the transient nature and the higher order wall motions that occur as a result of the rapidly increasing internal pressure. The relative flux error in each time step does however confirm that continuity remains satisfied (within the limit of convergence tolerance). The maximum flux error is 1.83×10^{-7} with a cumulative error of 2.73×10^{-6} (which can of course be further improved should the convergence tolerance, time step or discretisation be refined).

Performance plots of the investigated methods are shown in Figure 7.6 along with a summary of both the number of iterations and relative CPU time shown in Table 7.2. The IAC/FSI method allows for the solution of a fully enclosed, incompressible FSI problem. The performance is however approximately improved by a factor 2 through the inclusion of the additional QN terms. For test 2, the unit fluid-solid density ratio, along with the stiff structure results in a problem which initially provides serious difficulty for IAC in isolation. This can primarily be attributed to the material properties resulting in a coupled problem with very high interface gradients, and an inability of IAC/FSI to properly account for all the non-linearities present within the coupled FSI problem. The addition of the extra QN sensitivities are however sufficiently accurate to overcome this limitation.

It is important to understand that the convergence rates shown in Figure 7.6 are only representative of the typical convergence behaviour for each of the methods. At time step 150, option A (IAC/FSI+MVQN) is capable of producing near quadratic convergence. However, this will not be the case for all time steps. It does nevertheless provide an indication of how good the MVQN Jacobian approximations can be. Additionally, the comparative results between Test 1 and Test 2 illustrate that the ad-



(a)



(b)

Figure 7.4: Balloon inflation results for solid test 1 showing (a) $\boldsymbol{u}_x$ contour plot at 10 seconds and (b) pressure level at point A.

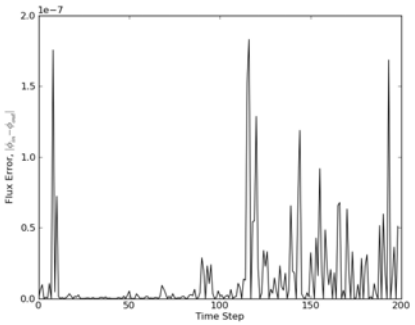


Figure 7.5: Boundary flux error made in each of the time steps for the balloon inflation test problem. The cumulative error across the entire simulation is 2.73×10^{-6} with a maximum error of 1.83×10^{-7}.

Table 7.2: Comparison of the number of iterations and relative computational time for the balloon inflation test problem.

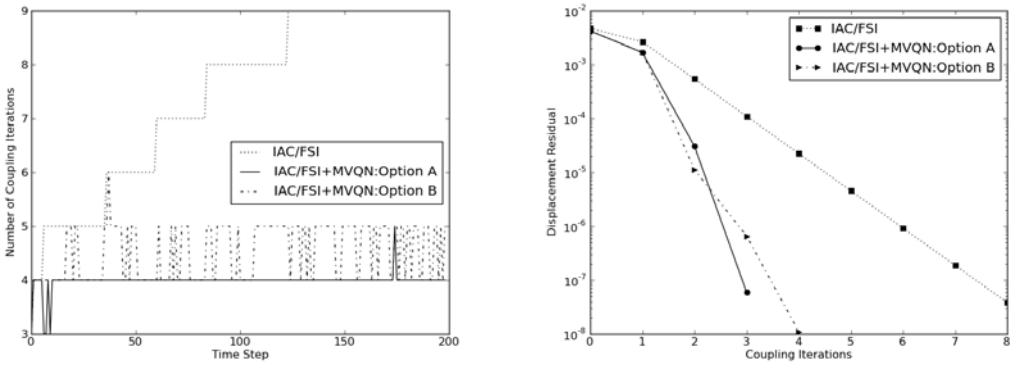
	Test 1				Test 2			
	Mean	Min	Max	CPU	Mean	Min	Max	CPU
IAC/FSI	7.44	3	9	2.07	10.65	6	21	2.45
IAC/FSI+MVQN: Option A	3.98	3	5	1.00	5.07	4	8	1.04
IAC/FSI+MVQN: Option B	4.53	3	6	1.14	4.85	4	8	1.00

ditional $\boldsymbol{u}_{AC}$ term has little influence on the coupling behaviour of IAC/FSI+MVQN. For Test 1 we observe a minor benefit if it is included but not so for Test 2. One possible explanation for the small difference between options A and B is that the additional IAC term dissapears quickly during the convergence of the coupling iterations, minimising the impact of the additional sensitivity approximations.

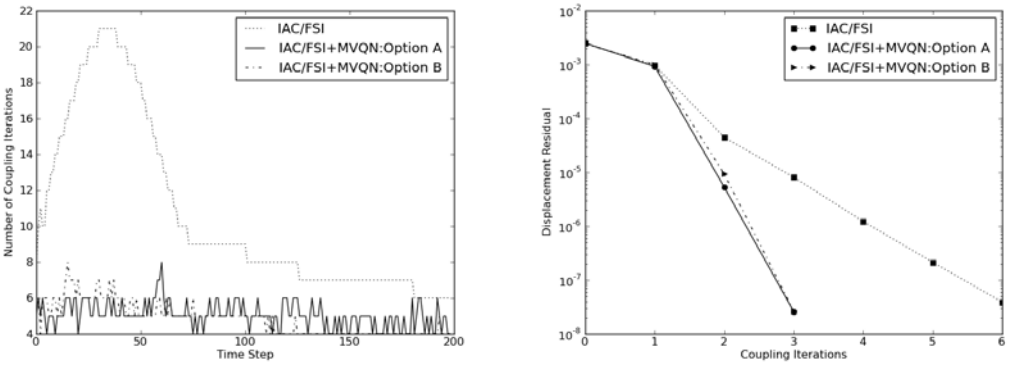
Lastly, in Figure 7.7 we perform an exhaustive search over 3 different time steps illustrating the effect $\boldsymbol{\beta}$ has on the performance of IAC/FSI. The intention of the search is to illustrate that there is a very narrow bandwidth for acceptable AC coefficients. In all the times analysed here, the test load method does provide the optimal/near optimal solution to $\boldsymbol{\beta}$. Despite this, AC/FSI in isolation does suffer from sub-optimal convergence rates. The inclusion of the additional QN terms not only removes this narrow-bandwidth with respect to the computed values of $\boldsymbol{\beta}$, but also reduces the overall number of coupling iterations.

7.4.3 Damped Structural Instability Fully Enclosed Test Problem

The last problem we analyse is a more interesting fully enclosed domain problem than the one analysed in Section 7.4.2. The problem's geometry is based on the test problem introduced in [72] and is depicted here in Figure 7.8. The problem consists of a curved fluid domain which is in turn fully enclosed by thin structural membranes of differing stiffnesses. The fluid domain is loaded with a gravitational body force of $g_y = -1\frac{\mathrm{m}}{\mathrm{s}^2}$ and asymmetrical parabolic inflows is prescribed, with the right inflow being slightly higher. The velocities are slowly ramped up for $t < 1.0$s via the sinusoidal curve, $\left(\sin\left(\pi\left(t + 3/2\right)\right) + 1\right)/2$. The respective short edges of the structural domains are fixed. The structure is described by a geometrically non-linear FEM formulation



(a)



(b)

Figure 7.6: Performance plots illustrating the number of coupling iterations required for each of the time steps and the typical convergence behaviour for each of the methods (shown for time step 150) for the balloon inflation test problem for (a) Test 1 ($E = 7 \times 10^5 \text{kN/m}^2$, $\rho_s = 1000\text{kg/m}^3$, $\rho_f = 1.1\text{kg/m}^3$) and (b) Test 2 ($E = 7 \times 10^5 \text{kN/m}^2$, $\rho_s = 1.1\text{kg/m}^3$, $\rho_f = 1.1\text{kg/m}^3$).

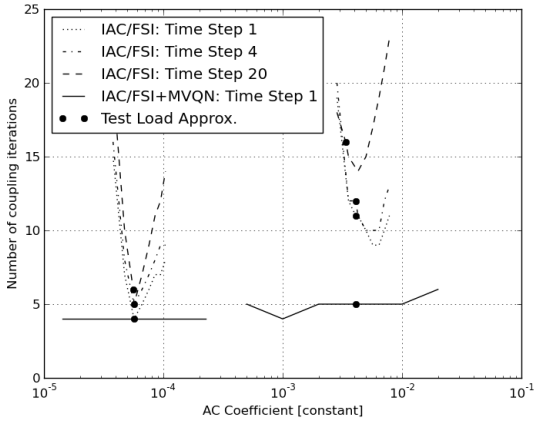


Figure 7.7: Effect of $\boldsymbol{\beta}$ on the number of coupling iterations, shown for three different time steps. On the left are the results for test 1 with test 2 results on the right. $\boldsymbol{\beta}$ for the exhaustive search is computed as a constant field value. Notice the fine bandwidth of acceptable values of $\boldsymbol{\beta}$ for the IAC/FSI implementation and the smoothing effect of the additional QN terms.

and is modelled with 100 quadratic, full integration elements where the fluid domain is modelled using 3600 linear FV elements. The time steps utilised for the simulation is $\Delta t = 0.005$s with convergence criterion of $\epsilon = 1 \times 10^{-7}$.

The primary idea behind the problem is that as fluid is introduced into the domain, the top (less stiff) structure initially deforms to accommodate the additional fluid flow. As some critical pressure is reached, the bottom structure eventually collapses. As in [72] we solve the problem ignoring the potential cavitation effects which may arise in a problem of this nature, and are again primarily interested in the comparative convergence behaviour of the coupling procedures analysed here.

In Figure 7.9 we show the velocities along with the domain deformations at various time steps. The corresponding reference pressure at point A is shown in Figure 7.10. The results, despite the minor differences in problem setup, correspond qualitatively well to those presented in [72]. As the fluid domain reaches some critical pressure, the bottom structure slowly collapses; with the collapse occurring on the side with a higher inflow velocity. The collapse leads to an initial decrease in internal pressure followed by a period of highly oscillating pressure as the structure and fully enclosed domain undergo rapid deformations. Again, to demonstrate that continuity is satisfied at convergence within each time step we illustrate the flux error made in each time step

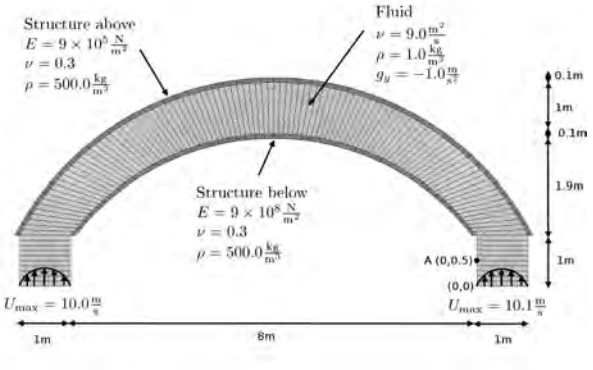


Figure 7.8: Problem description for the structural instability fully enclosed domain problem.

in Figure 7.11.

The damped structural instability problem represents an interesting challenge for the artificial compressibility method. Firstly, we have varying structural stiffnesses which highlight the need for a non-constant AC coefficient along the interface. At the same time, the problem, as with the balloon inflation problem, poses a difficulty to the AC method due the inherent duality in the problem. Firstly, as the structural domain undergoes large deformations, the internal solid stresses increase, resulting in an increased resistance to any further deformation. This is again in contradiction to the increasing inertial forces due to the high structural densities and relatively high velocities (especially once the snap through of the bottom structure occurs). The problem is further complicated by the fluid body force.

In Table 7.3 we summarise the comparative performance of IAC/FSI and IAC/FSI+MVQN. While IAC/FSI in isolation is entirely capable of solving the problem, the performance is improved through the additional quasi-Newton approximations. Figure 7.12 shows the total number of coupling iterations for each of the time steps as well as the representative convergence behaviour (shown here for time step 150). Moreover, as demonstrated previously, the penalty in omitting the additional artificial compressibility displacement u_{AC} is near negligible.

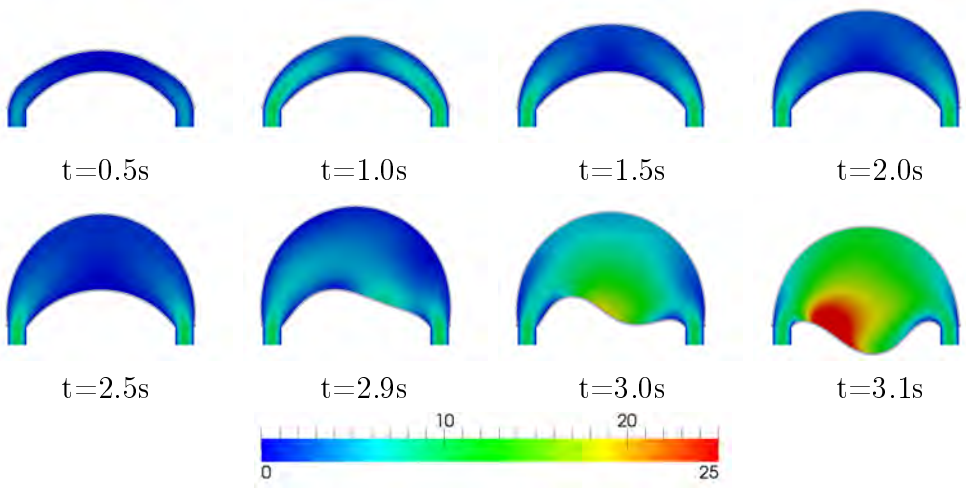


Figure 7.9: Velocity fields with structural displacements shown at different time steps for the curved domain fully enclosed test problem.

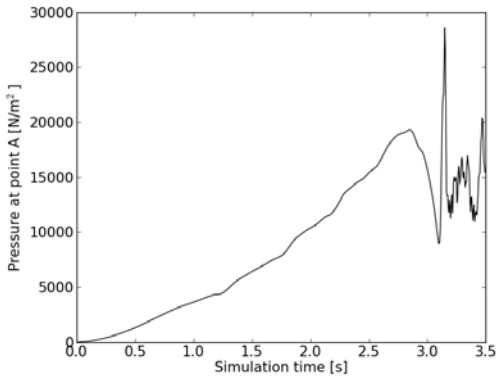


Figure 7.10: Plot of the pressure level at point A, for the curved domain fully enclosed problem.

Table 7.3: Comparison of the number of iterations and relative computational time for the damped structural instability problem.

	Mean	Min	Max	CPU
IAC/FSI	12.21	3	17	2.84
IAC/FSI+MVQN: Option A	4.21	3	5	1.00
IAC/FSI+MVQN: Option B	4.23	3	6	1.00

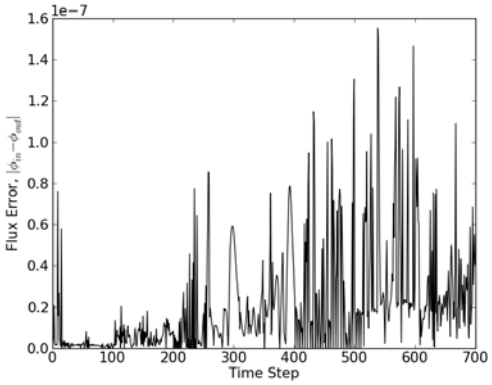


Figure 7.11: Total flux error made within each of the time steps for the curved domain fully enclosed problem.

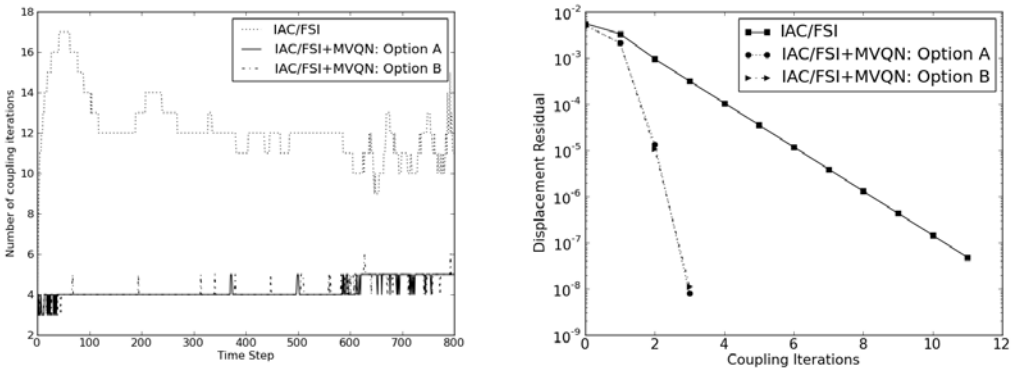


Figure 7.12: Comparative performance plots for the curved domain, fully enclosed test problem showing the number of coupling iterations within each time step and the typical convergence behaviour (shown here for time step 150).

7.5 Conclusion

In this chapter we addressed the augmentation of AC/FSI through the inclusion of quasi-Newton methods. We demonstrated how the inclusion of the AC term enabled the solution of fully-enclosed, incompressible, FSI problems. Despite the test load method providing a reasonable means by which to approximate the AC coefficients, the coupling performance of AC/FSI was demonstrated, to at times provide sub-optimal convergence. The additional QN terms in turn provide a means by which to better account for the variety of non-linearities present in the system, and consequently results in an improved convergence behaviour. One of the main attractions of the AC/FSI method is its ability to treat the solid solver and the modified fluid field solver as black-box operators, which is retained for the proposed AC/FSI+QN coupling scheme. The combination of AC/FSI+QN is however slightly more intrusive than the original AC/FSI as the QN updated pressures from the previous iteration has to be returned to the fluid solver. The MVQN method was further demonstrated to be a suitable choice for approximating the required system sensitivities.

Chapter 8

Partitioned Coupling of Steady State, Free Surfaces, and Solid Body Contact

8.1 Introduction

In this chapter, we shift our focus away from the traditional FSI problems already analysed, namely incompressible, strongly coupled, transient FSI problems. The most powerful defence of the black-box, partitioned coupling philosophy is the inherent modularity it offers. Either through the use of mature software technologies which can solve a wide variety of different kinds of problems, or by the inherent ability to replace one or both of the field solvers. To this end, in this chapter we extend the analysis of our black-box coupling environment to steady state problems, FSI problems with free surfaces, and FSI problems with solid body contact. The primary aim is to determine whether the coupling environment, with consistent interface information transfer in conjunction with the QN methods, are sufficiently robust to work "as is" for the range of alternate FSI problems.

8.2 Steady State

Steady state FSI problems are, generally speaking, far less prevalent than transient problems. Researchers and designers are usually more interested in the dynamic interaction of flow induced structural deformations or vice versa. Steady state problems are

however of potential interest as a benchmarking tool (see for example [105]), and have found application in topology and shape optimisation (see for example [118] and the citations therein). While most FSI publications focus on the complexities of strongly coupled problems, specifically with high added mass instabilities, steady state problems offer their own set of challenges. While steady state problems are not density driven, they lack time steps which typically impose physical limits on possible structural deformations.

It is of course possible for steady state problems to be solved using pseudo-time iterations. By this, we imply that the boundary conditions, for example inflow velocity, can be slowly ramped up over several iterations. In doing so, the steady state problem can directly be related to the transient FSI methods developed in the preceding sections of this thesis. Similarly, a steady state problem can be solved using a transient solver, where if the problem is by nature a steady state problem, the transient simulation will eventually, given enough time, reach steady state.

In this section, we are interested in how well the various QN methods perform for a true steady state problem. For steady state problems, it is important to note that the IBQN-LS and the newly proposed MVQN methods are identical. To implement steady state, the only change made to the coupled solver is replacing the incompressible, transient fluid solver *pimpleDyMFoam*, with *simpleFoam* (an incompressible steady state solver based on the SIMPLE coupling algorithm).

8.2.1 Steady State Flow Past Elastic Structure

In this section, we solve for the steady state solution of cross-flow over a thin, elastic membrane; the geometry and material properties are shown in Figure 8.1. A slip boundary condition is imposed at the top, and both the bottom and FSI interface have a no-slip boundary condition. Since the complexity of steady state problems is not a function of density ratios or a given time step size, we modify the difficulty of the problem by changing the inlet flow velocity. We therefore have two inflow velocities, namely $U_{\text{in}} = 0.1\text{m/s}$ and $U_{\text{in}} = 1.0\text{m/s}$. In total, 1276 linear triangular FVM elements are used with 40 quadratic, geometrically non-linear FEM elements.

The elastic deformation and the fluid flow velocities are shown in Figure 8.2, and the convergence behaviour of the QN methods are shown in Figure 8.3. Overall, the black-

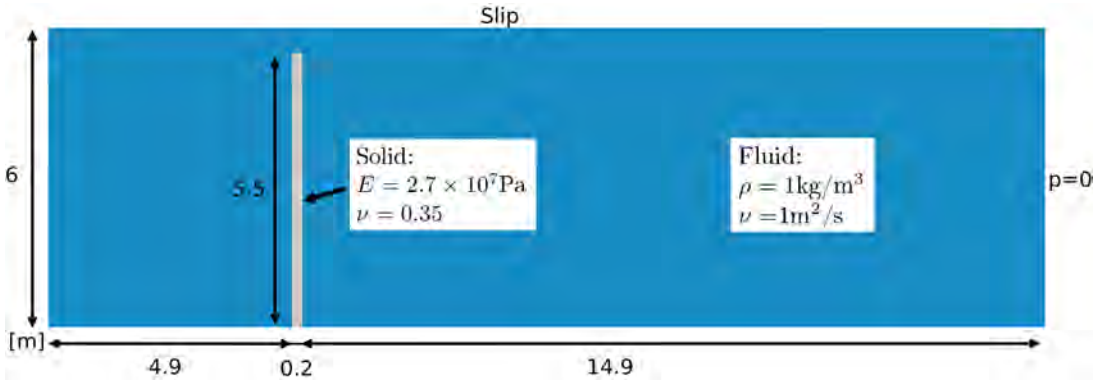


Figure 8.1: Steady state domain description.

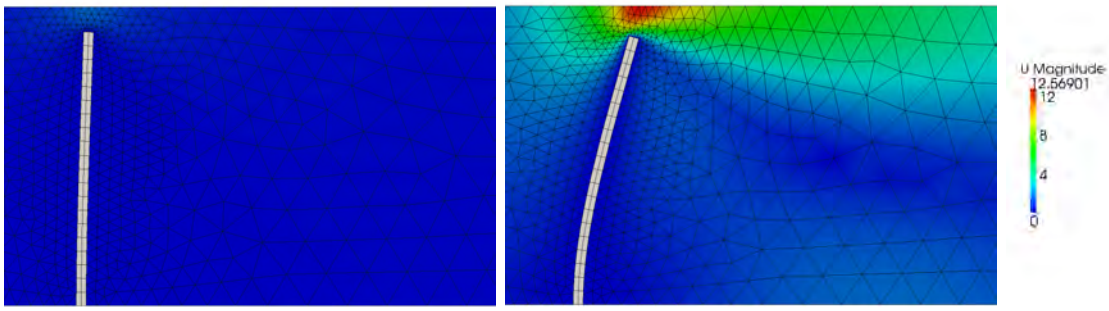
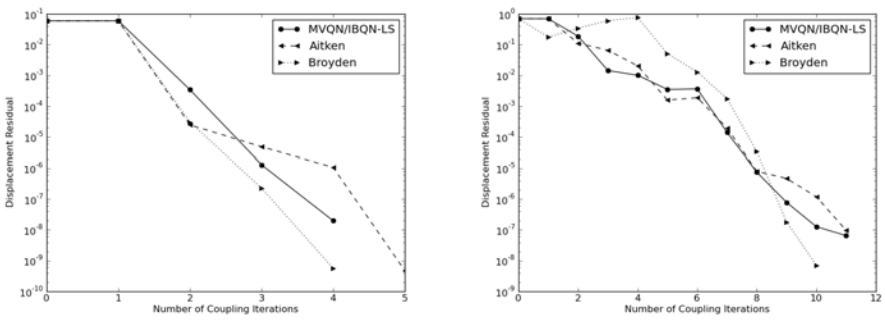


Figure 8.2: Steady state simulation results of a flexible membrane for inlet velocities of $U_{\text{in}} = 0.1$m/s (left) and $U_{\text{in}} = 1.0$m/s (right).

box coupling environment developed in this thesis appears to be suitable for steady state problems. For both test cases the total number of convergence iterations required are far fewer than the total interface DOFs.

Aitken's method is surprisingly competitive for the steady state problems, while Broyden's method in general provides the best asymptotic convergence rates. The comparative convergence between $U_{\text{in}} = 0.1$m/s and 1.0m/s, does suggest that pseudo-iterations may be a necessity given any additional complexity. In fact, while not shown, the current problem fails to converge given an inlet velocity much higher than 1.0m/s. Additional pseudo-iterations, while slowly ramping up the inlet velocity, would provide a physical means by which the deformation within a given pseudo-iterate can be limited.



(a) (b)

Figure 8.3: Convergence rates for steady state simulation for inlet velocities (a) $U_{\text{in}} = 0.1\text{m/s}$ and (b) $U_{\text{in}} = 1.0\text{m/s}$.

8.3 FSI with Free Surfaces

FSI with free surfaces, involving totally or partially submerged bodies, is of interest in a number of fields, including civil and offshore engineering. Multiphase flow requires accurately capturing the interface between two or more fluids, and can be relatively unsteady due to large density differences within the flow domain. Due to these, potentially large, density differences within the flow domain, the FSI interface can now be subjected to abruptly changing interface loading conditions (consider for example an advancing wave striking a flexible structure). FSI with free surfaces have been the focus of several studies, and have been solved both using partitioned methods and monolithic schemes [27, 68, 81, 111, 113].

In this section we aim to demonstrate how the black-box coupling environment copes with free surface flows. The QN methods introduced in this study, are methods that are *trained* over time. The primary interest in our analysis is thus how well these QN methods deal with very sharp and abruptly changing interface forces. To this end, we analyse two test problems. Firstly, we consider a simple dam break problem where an advancing wave strikes a thin elastic structure. Secondly we analyse a submerged cube with a density lower than the surrounding fluid, resulting in the cube rising, and eventually breaking through the free surface interface. Once again, the only change made to the coupled environment is replacing the fluid solver.

In OpenFOAM, free surface flow problems are solved using a variant of the Volume

of Fluid (VoF) method [90, 106]. In general, the VoF method considers the entire flow domain as a "single" fluid, where the fluid material properties vary, based on "volume fractions". Consider for example a two phase flow problem, where $\Omega_f = \Omega_{1,f} \bigcup \Omega_{2,f}$, where $\Omega_{1,2}$ are two distinct fluid phases (for example water and air), sharing an interface Γ_{FS}. The fluid properties (ρ,ν, etc.) are then functions of the volume fraction function γ_i such that

$$\rho = \gamma_1 \rho_1 + (1 - \gamma_2)\, \rho_2, \quad \text{and} \quad \nu = \gamma_1 \nu_1 + (1 - \gamma_2)\, \nu_2. \tag{8.1}$$

The volume fraction is strictly bounded, $0 \leq \gamma_i \leq 1$, where

$$\gamma_i = \begin{cases} 1, & \text{for } \Omega_{1,f} \\ 0, & \text{for } \Omega_{2,f} \\ 0 \leq \gamma_i \leq 1 & \text{in transition zone} \end{cases} . \tag{8.2}$$

Since the finite volume method cannot handle distinct discontinuities, the interface transition zone involves a smooth, continuously varying function. The primary distinction between various VoF methods relate to the specific formulation of the transition zone and the compression method used to maintain a sharp warp free interface. In this thesis, we make use of the OpenFOAM free surface flow solver *interDyMFoam* (which also natively handles moving meshes). The exact variant of interface compression used by *interFoam* has not been published, but for a closely related method we refer the reader to the PhD of Rusche [90]. For a validation of *interFoam* applied to coastal systems see [78]. We further refer interested readers to the PhD of Kassiotis [68] for additional information and validation of *interDyMFoam* applied in a strongly coupled partitioned FSI environment.

8.3.1 Dam Break with Elastic Obstruction

The first free surface problem we analyse is that of a collapsing column of water, striking an elastic wall, previously analysed in [68, 111]. The problem setup is shown in Figure 8.4, and consists of a 29.2cm column of water which collapses under gravity, striking an 8cm tall, 1.2cm wide, elastic obstacle. The tank is open at the top, and surface tension effects are ignored due to the large length scales. The dam break prob-

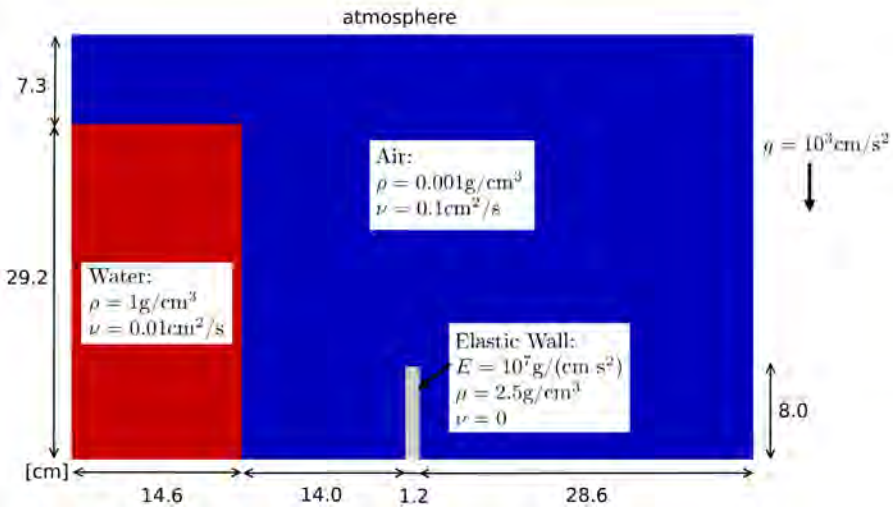


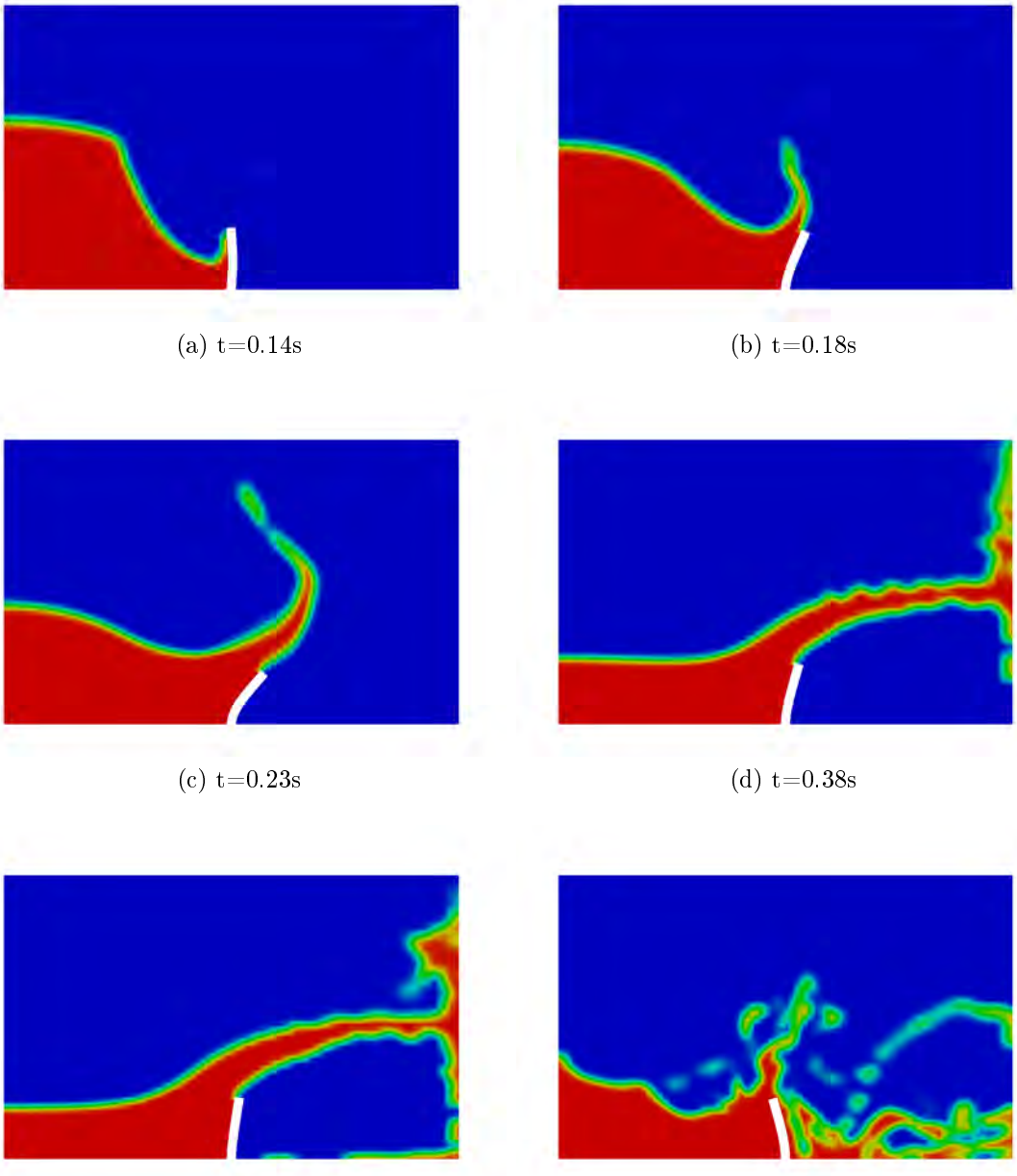
Figure 8.4: Dam break with elastic structure problem description.

lem conceptually poses difficulties for the QN methods. Prior to the striking of the advancing front, the coupling algorithms spend several time steps training the system Jacobians given a very low loading condition, followed by a very sharp and sudden increase in interface forces.

The problem is solved here using 3672 linear FVM elements and 14 quadratic, full integration elements. The chosen time step size is 0.001s, an order of magnitude larger than what was used in [68]. In Figure 8.5, the advancing front along with the elastic deformation is shown for various time steps.

In general, the problem is well behaved, and appears to pose little challenge to the black-box coupling environment. In Table 8.1, we compare the number of iterations and relative CPU time using each of the QN methods previously discussed. Further plots of the typical convergence behaviour and number of coupling iterations per time step are shown in Figure 8.6. The typical performance behaviour observed in Chapter 5 holds true here as well. Aitken's method, while capable of solving the problem, remains inefficient. As previously observed, the MVQN method provides the most efficient coupling method, and is again capable of providing near quadratic convergence behaviour, whilst the IBQN-LS performance and stability remains linked to the choice of (q).

In general, it appears that the QN methods are quite adept at coping with large and sudden changes in loading condition; with the possible exception of the IBQN-LS



(a) t=0.14s (b) t=0.18s

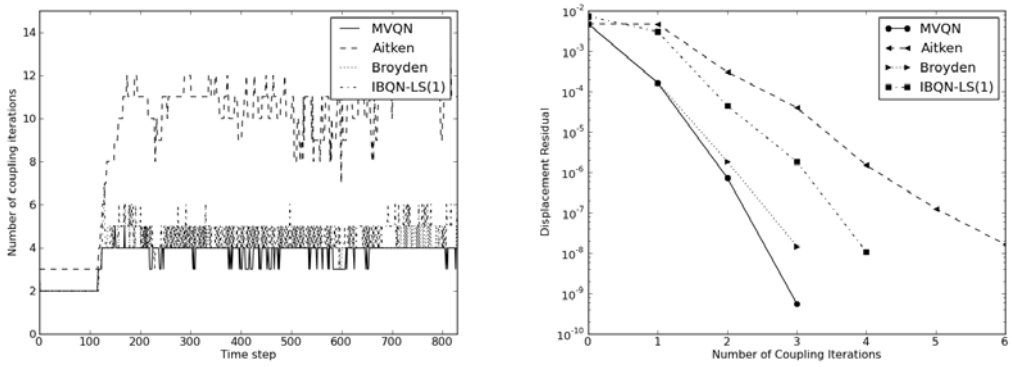
(c) t=0.23s (d) t=0.38s

(e) t=0.46s (f) t=0.8s

Figure 8.5: Wave interaction with elastic wall at different time steps.

Table 8.1: Comparison of the number of iterations required and comparative CPU time for the dam break benchmark problem.

	Mean	Min	Max	CPU
MVQN	3.60	2	5	1.00
Aitken	9.46	3	14	3.08
Broyden	3.87	2	5	1.07
IBQN-LS(1)	4.27	2	7	1.19
IBQN-LS(2)	non-convergence (129/850)			



(a) (b)

Figure 8.6: Comparative plots of the (a) number of coupling iterations required in each time step and (b) the typical convergence behaviour shown here for time step 130 (directly following the wave striking the elastic structure).

method, which requires restricting the QN approximation to a single time step.

8.3.2 Submerged Square

We now analyse a single, low density square submerged in water. Due to the density differences, the square rises, until eventually "popping" free of the water-air interface. The geometry and material properties for the problem are shown in Figure 8.7. The square here is analysed using a single linear FEM element. The problem can therefore be likened to a DEM-CFD (discrete element modelling) class of problems, but solved here with a fully coupled, implicit FSI solver. The density of the submerged cube is chosen to be two orders of magnitude lower than the enclosing water. The water is contained within a tank with an atmosphere boundary condition imposed at the top.

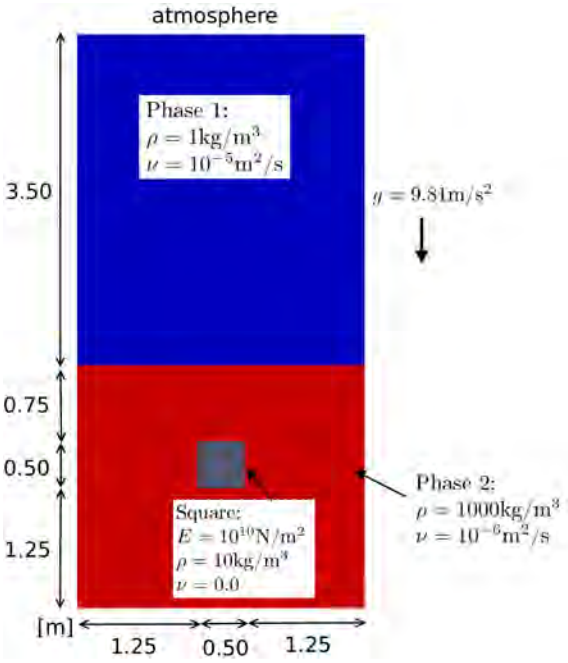


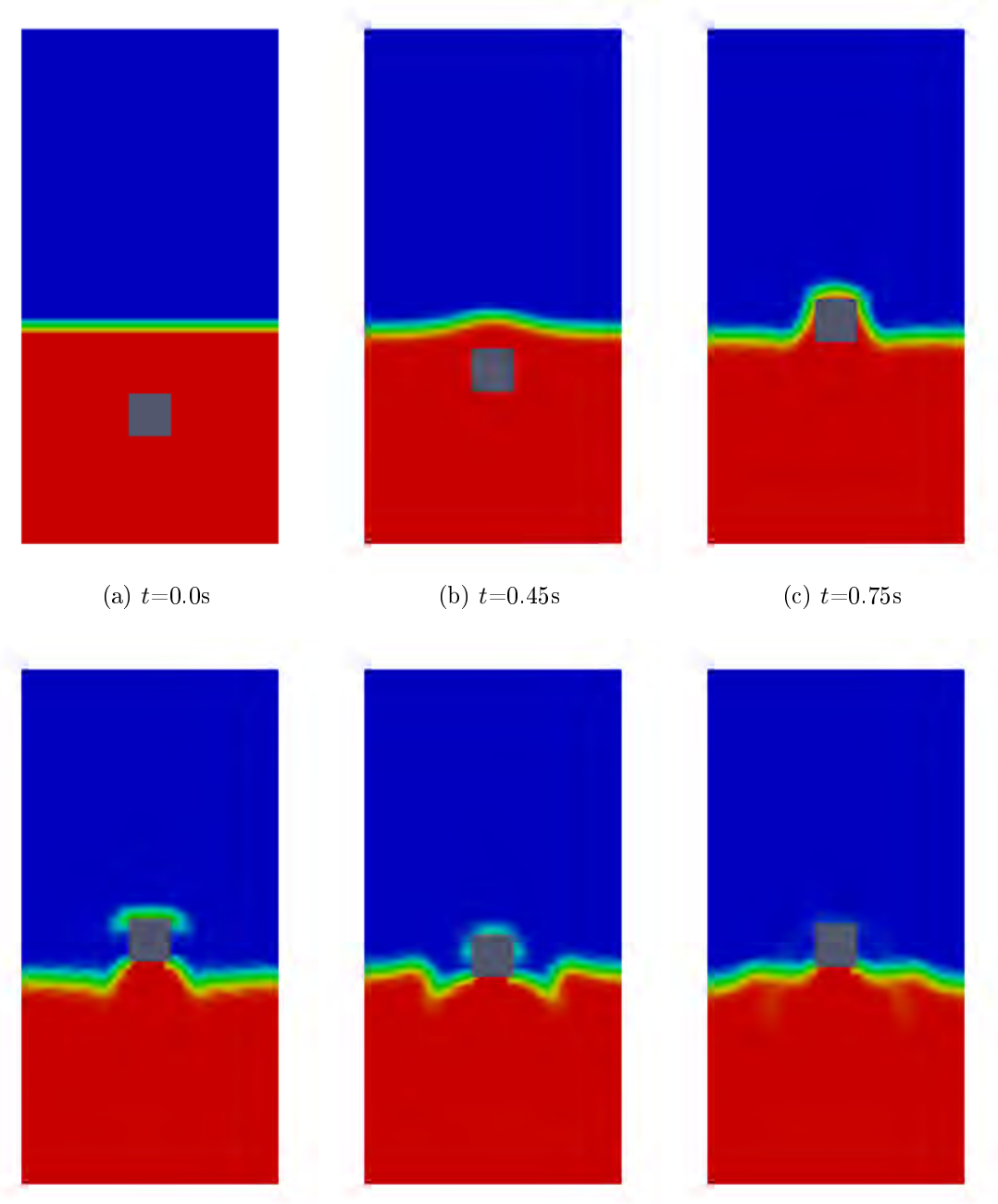
Figure 8.7: Submerged square test problem description.

For this problem, both the square and liquid are subjected to a gravitational force.

From an interface transfer perspective, the problem serves as an additional validation test case. The higher density water provides a linearly varying hydrostatic pressure field, which is transferred to a single structural element. In total, 1866 linear hexahedral FVM elements are used compared to the single structural element. We solve the system for 5 seconds, with a time step size of $\Delta t = 0.01$s.

In Figure 8.8, the square displacement is shown at different time steps. The hydrostatic pressure forces the square upwards, and the eventual exit of the square through the free surface causes waves, which in turn are contained by the walls, causing the square to bob up and down. This can more clearly be demonstrated by the y-displacement plot in Figure 8.9, where the y-displacement is also shown for an equal density square, $\rho_s = 1000\text{kg/m}^3$, which remains perfectly stationary. A further indication of the overall accuracy of the numerical implementation is the lack of rotation. While unrealistic from a physical perspective, the system is in perfect symmetry, and any rotation would have been the result of numerical and spatial modelling inaccuracies.

The coupling performance of the different QN methods is summarised in Table 8.2 and Figure 8.10. Despite the comparatively large time steps, and the solid density



(a) t=0.0s (b) t=0.45s (c) t=0.75s

(d) t=1.1s (e) t=1.6s (f) t=2.5s

Figure 8.8: Square displacement shown at different time steps.

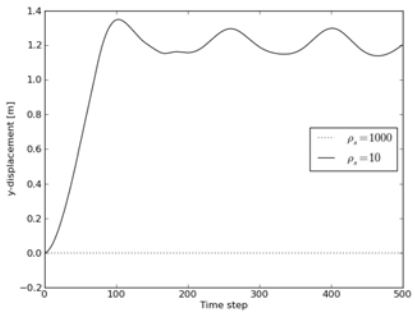


Figure 8.9: Square y-displacement for $\rho_f = 1000\text{km/m}^3$ with $\rho_s = 10\text{kg/m}^3$ and $\rho_s = 1000\text{kg/m}^3$.

Table 8.2: Comparison of the required number of iterations for the rising square test problem for $\rho_s = 10\text{kg/m}^3$.

	mean	min	max
MVQN	3.91	2	7
Aitken	5.47	3	10
Broyden	4.14	2	14
IBQN-LS(0)	non-convergence (16/500)		
IBQN-LS(1)	5.86	2	18
IBQN-LS(2)	non-convergence (87/500)		

being 2 orders lower than the fluid, all 4 QN methods provide converging results. Surprisingly, the IBQN-LS method is the worst performing QN method. This problem is an excellent example of a problem where information from multiple time steps will be linearly dependent; the relative change in displacement and interface forces remain unchanged for several time steps until the square eventually rises free of the free surface interface.

8.4 Solid Contact

8.4.1 Problem Description

In this section we aim to provide a preliminary analysis of the QN methods applied to FSI problems with solid body contact. Examples of FSI with solid body contact include opening and closing heart valves [3, 107], interaction between blood cells, or blood cells

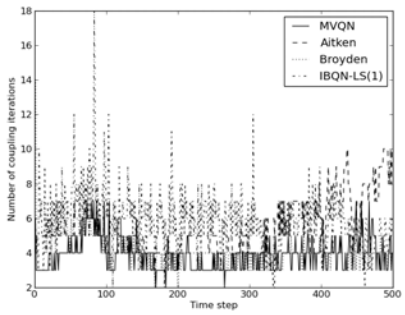


Figure 8.10: Plot of the number of required iterations for the rising square problem for $\Delta t = 0.01$s and $\rho_s = 10\text{kg/m}^3$.

flowing through an occlusion or capillary. Most mature FEM packages, including for example Abaqus, and in our case Calculix, support solid body contact. Solid body contact is commonly done through the inclusion of springs, to impose forces to resist movement or deformation [33, 108]. The contact force magnitude is then based on the penetration distance and the chosen spring stiffness function.

Solid body contact via the insertion of spring forces pose a very serious challenge to FSI solvers. Prevention of large inter-body penetration may involve arbitrarily large contact forces, and these occur sporadically, only when contact is active. Therefore, while Calculix natively supports contact, we implement a simple contact algorithm externally to the solid solver. We do so in order to have the contact forces available, and thus can directly compare the ability of the QN methods to deal with contact when the forces are both known and not known. The intention therefore is to analyse how well the QN methods perform when implemented in a true black-box setting compared to when the contact forces are made visible when constructing the interface force-displacement sensitivity approximations.

It should be mentioned that flow solvers based on an ALE description of the fluid domain (as used in this thesis) are not ideally suited to contact problems. The event of contact implies that the fluid region between two bodies in contact should disappear, which cannot be done naturally using solvers such as OpenFOAM. For these classes of problems, methods such as the immersed boundary method [46], fixed grid methods [45] or the fictitious domain method [23, 107] are better suited. The primary point of interest in our analysis is the behaviour of the QN methods, and we therefore avoid this

issue entirely by forcing a small gap to remain present at all times.

In order to fully close off a domain, interface tracking methods, based on ALE reference frames, will require topology changes. The topology changes refers to both localised remeshing, when the mesh quality using mesh movement deteriorates, and redefining surface boundaries. This is both time consuming, problem specific, and may introduce errors when mapping solutions between different domain or topology definitions.

As an alternative, in this study, we symbolically close the small gap via the fluid equations, to avoid topological changes to the fluid domain and mesh. The fluid velocity in the cells between the two surfaces in contact is forced to be equal to the adjoining interface velocities. This allows for the small gap to remain, while the effect on the flow profile will resemble the flow profile when the fluid cells are removed, barring for the geometrical error introduced by the small gap. This geometrical error is directly proportional to the size of the enforced gap, and is a function of the solver CFL requirements (the smaller the enforced gap, the smaller the maximum time step size). Interestingly, closing of the gap via the equations does not induce the "incompressibility dilemma" observed for fully enclosed problems, which is probably due to the iterative (SIMPLE-like) velocity-pressure solution algorithm.

8.4.2 Contact Formulation via Springs

In this thesis we implement a simple spring analogy contact formulation, based on the penetration distance of slave surface co-ordinates in a master surface. For the purposes of this analysis, we ignore frictional forces and only include normal contact forces; this is illustrated in Figure 8.11. The contact algorithm implemented here can be summarised as follows:

1. Pair up slave nodes to master surface elements.
2. Compute the distance between slave nodes and master surface elements.
3. If slave node penetrates master surface, connect a linear spring, with direction normal to master surface.

The spring force, when active, is given by

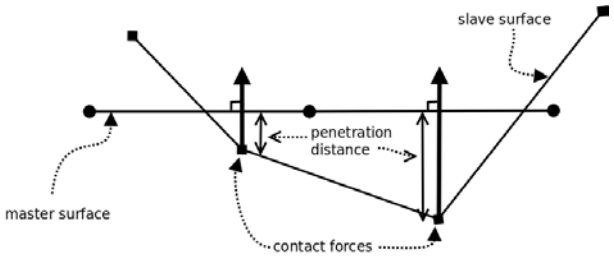


Figure 8.11: Simple depiction of slave nodes to master surface contact.

$$F_{\text{contact}} = (Kd_p)\,\boldsymbol{n}_m, \tag{8.3}$$

where K is the spring stiffness coefficient, d_p the penetration distance and $\boldsymbol{n}_m$ the master surface normal vector. Choosing K as a large constant will result in hard contact, or conversely a small K will result in soft contact which leads to larger penetrations. Because we model 2D systems in 3D with a single element through the thickness, two contact springs will be added, whenever contact is active for a given node. While this does not alter the behaviour of the contact system, it is mentioned here for completeness.

8.4.3 Including Contact Forces in QN Approximations

In the test problem to follow (Section 8.4.4), when we mimic a black-box implementation of contact, we only add the extra contact forces to the solid interface but do not make these forces visible to the QN approximations. As we will demonstrate, the performance of the QN approximations are enhanced when these contact forces are included in the constructions of these approximations. There are however two options of how to include the additional contact forces within the $\Delta \boldsymbol{F}_F$ and $\Delta \boldsymbol{F}_S$ matrices for the MVQN and IAC/FSI+MVQN methods.

The primary difference comes from whether the $\Delta \boldsymbol{F}$ matrices are constructed using integrated force vectors or traction vectors which still need to be integrated along the structural interface, i.e.

$$\boldsymbol{f}_k = \left\{ \begin{array}{c} \boldsymbol{f}_{x,k} + \boldsymbol{f}_{x,k,\text{contact}} \\ \boldsymbol{f}_{y,k} + \boldsymbol{f}_{y,k,\text{contact}} \\ \boldsymbol{f}_{z,k} + \boldsymbol{f}_{z,k,\text{contact}} \end{array} \right\}, \tag{8.4}$$

or

$$\boldsymbol{f}_k = \left\{ \begin{array}{c} p_k \\ \boldsymbol{S}_{x,k} \\ \boldsymbol{S}_{y,k} \\ \boldsymbol{S}_{z,k} \\ \boldsymbol{f}_{x,k,\text{contact}} \\ \boldsymbol{f}_{y,k,\text{contact}} \\ \boldsymbol{f}_{z,k,\text{contact}} \end{array} \right\}, \tag{8.5}$$

where $\boldsymbol{S}_k$ represents the surface shear stresses for iteration k, in time step $n+1$. Notationally each entry indicates vectors, for example $\boldsymbol{f}_{x,k} = \left\{ f_{x,k}^1, ..., f_{x,k}^n \right\}^T$. In general, the coupling performance of the QN methods are largely unaffected by the form of $\boldsymbol{f}_k$. Constructing $\Delta \boldsymbol{F}$ using the concentrated nodal forces in (8.4) is therefore the obvious choice as it leads to smaller matrices. For the IAC/FSI+MVQN method, $\boldsymbol{f}_k$ in the form of (8.5) might lead to larger matrices, but overall simplifies the approximation significantly. This stems from equation (7.13), where the IAC/FSI pressure p_{k-1}^{n+1} has to be updated within the fluid domain to account for the AC approximation, via

$$p_{k-1,\text{new}}^{n+1} = p_{k-1}^{n+1} + \Delta \boldsymbol{f}_{\text{pressure component}}. \tag{8.6}$$

If $\boldsymbol{f}_k$ was to be constructed in the form of (8.4) with contact forces included, then it is not strictly obvious, if even possible, how to determine which portions of the normal components of $\Delta \boldsymbol{f}$ are due to pressure and not to contact. Therefore, when generating the QN approximations in the numerical experiments to follow, we construct $\Delta \boldsymbol{F}$ for the MVQN method using (8.4) and for the IAC/FSI+MVQN we use the form provided in (8.5).

8.4.4 Flexible Beam with Solid Body Contact

In this section, we analyse a simple valve like problem first analysed in [107]. The problem is representative of a mitral valve where the papillary muscle has been omitted, allowing for the valve to fully snap-through given the inlet velocity. The domain geometry is shown in Figure 8.12, where we choose the contact plane such that the valve is in contact from the start of the simulation. The fluid and valve properties are chosen

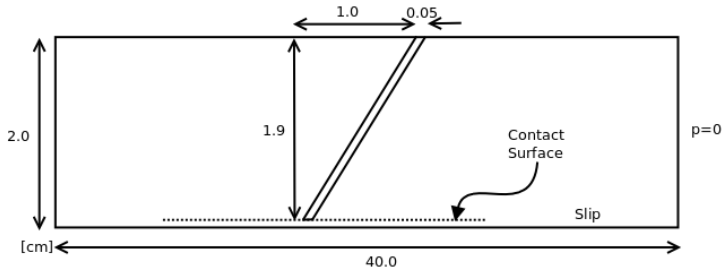


Figure 8.12: Flexible beam with solid body contact problem description.

to be the same as those used for the 3D flexible tube problem analysed in Section 5.5.3. As such, the fluid properties are described by a density of $\rho_f = 1.0\text{g/cm}^3$, and viscosity of 0.03P. The valve is described by a geometrically non-linear FEM formulation with $E = 3 \times 10^6\text{dynes/cm}^2$, $\rho_s = 1.2\text{g/cm}^3$ and $\nu = 0.3$. The problem presents two complexities in the form of high added mass with large deformations, and solid to rigid body contact.

The fluid domain is discretised using 1861 linear fluid elements, and 40 linear FEM elements are employed for the structural domain. We utilise linear elements for the solid domain due to our choice of using a slave *node*-master surface contact formulation. For higher order elements, a more sophisticated surface-surface contact scheme would be required. While linear solid elements are not ideal, they are sufficient for the purposes of illustrating the behaviour of the QN methods with solid body contact.

The inlet velocity is given by

$$U_{\text{in}} = 20.0\sin(2\pi t)\ \text{cm/s}. \tag{8.7}$$

We model the system up to 0.2 seconds, with a time step size of $\Delta t = 0.0001\text{s}$. To better understand the behaviour of the QN methods with contact, we analyse the system for two different contact stiffnesses, namely $K = 2 \times 10^3\text{g/cm}^2$ and $K = 1 \times 10^4\text{g/cm}^2$ which we hereafter refer to as soft and hard contact respectively. In Figure 8.13 we illustrate the difference in penetration distance for hard and soft contact. Soft contact allows for a much lower rate of contact force increase, and therefore is much easier to account for in the given approximations. On the other hand, finding a suitably soft spring coefficient is much more difficult, as it has to be chosen low enough to be easy to approximate but high enough that the gap does not become too small for the flow

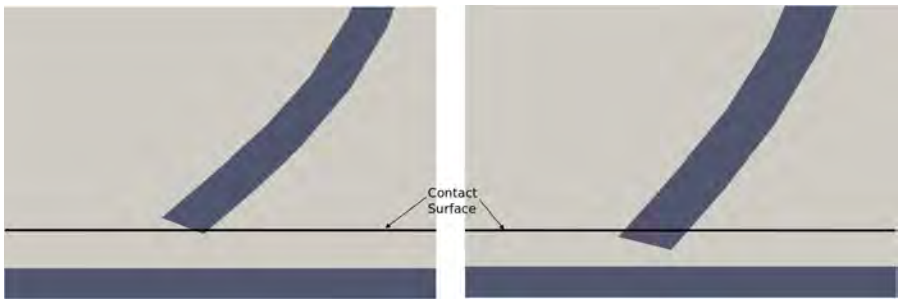


Figure 8.13: Contact penetration difference between hard contact, $K = 1 \times 10^4$ (left), and soft contact, $K = 2 \times 10^3$ (right).

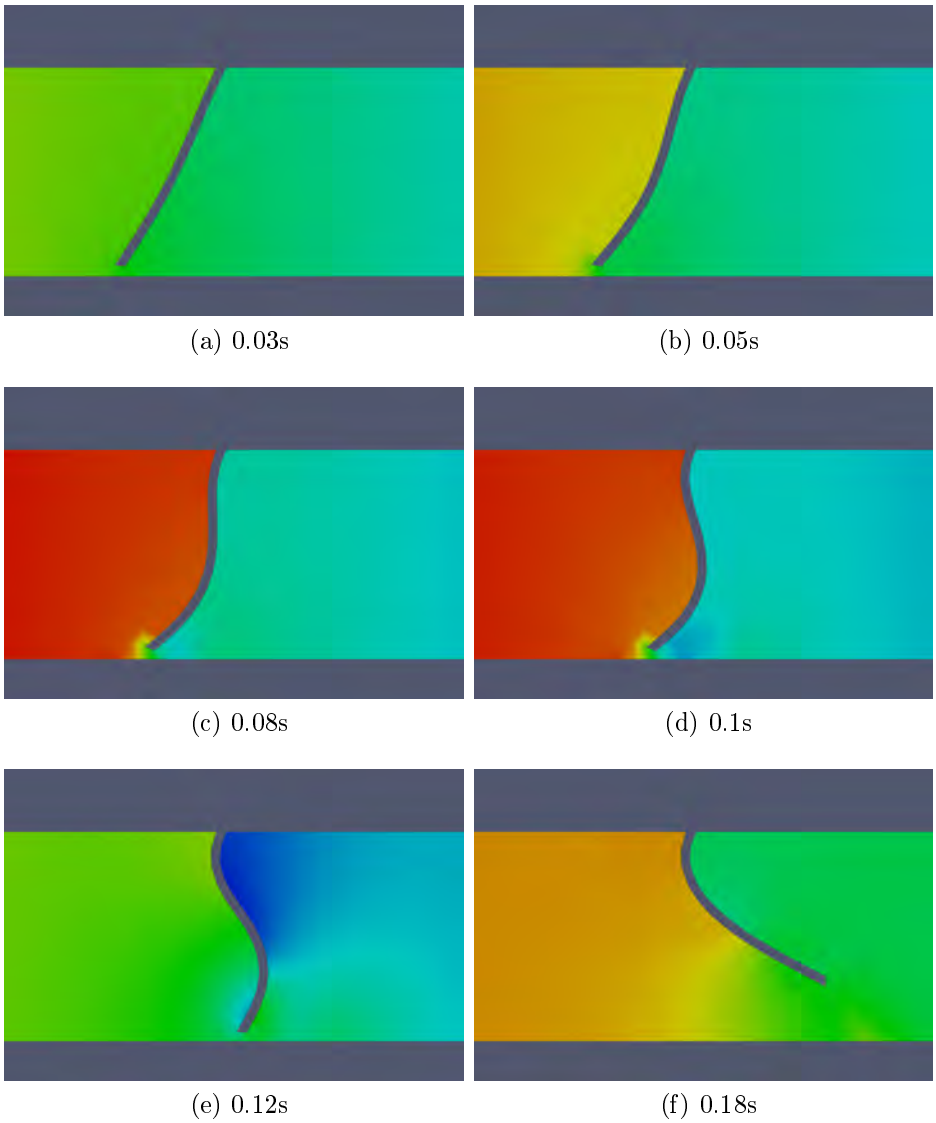
solver to remain stable.

The pressure plots as well as the valve displacement, for different time steps, are shown in Figure 8.14 for hard contact. Despite the small gap, there is still a large pressure drop across the restricted valve. In Table 8.3 we outline the number of iterations required by MVQN and IAC/FSI+MVQN with contact, with the typical convergence behaviour shown in Figure 8.15. The particular choice of $K = 2.0 \times 10^3$ is low enough for FSI with black-box contact to work using the MVQN method. Including the contact forces in the construction of the QN approximations, appropriately accounts for the system sensitivities to these contact forces; in so doing, the IAC/FSI+MVQN also becomes convergent for both soft contact and hard contact.

In general, it would appear that the partitioned coupling methods introduced in this thesis are not as adept at dealing with black-box solid body contact. The non-linearity posed by the large and rapid force increase is too large and perhaps abrupt for the QN methods. Choosing a lower spring stiffness function is one way in which to soften the problem, but this introduces the difficulty of finding a suitably soft stiffness. The stiffness must be high enough to limit the penetration, but soft enough to be solvable. Of course, smaller time step sizes are always possible but with an obvious penalty in computational times.

8.5 Conclusion

In this chapter we extended our analysis of partitioned FSI to include a steady state FSI problem, FSI with free surfaces and FSI with solid body contact. The primary



(a) 0.03s (b) 0.05s

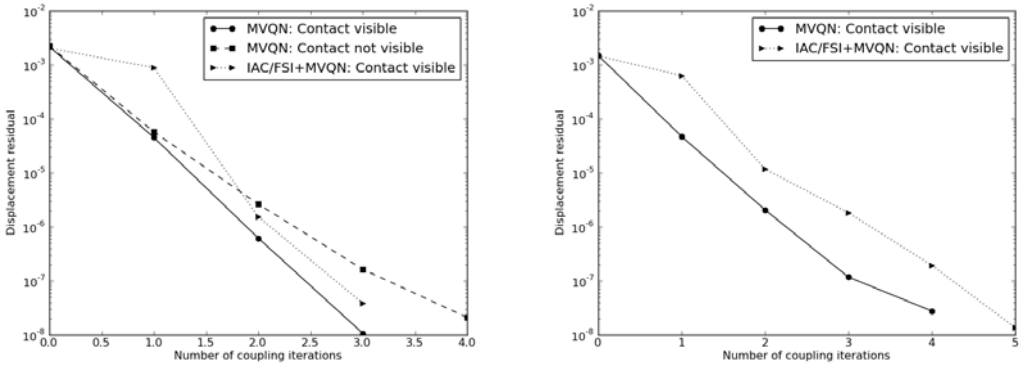
(c) 0.08s (d) 0.1s

(e) 0.12s (f) 0.18s

Figure 8.14: Pressure and valve displacement plots at different time steps, for $\Delta t = 0.0001$s and $K = 10^4$.

Table 8.3: Comparison of the number of iterations required for the valve with contact problem.

	contact forces	soft contact ($K = 2 \times 10^3$)			hard contact ($K = 1 \times 10^4$)		
	included	min	max	mean	min	max	mean
MVQN	yes	3	7	3.57	3	7	4.04
MVQN	no	3	8	3.71	non-convergence (47/2000)		
IAC+MVQN	yes	3	6	3.96	3	8	4.49
IAC+MVQN	no	non-convergence (164/2000)			non-convergence (47/2000)		



(a) (b)

Figure 8.15: Typical convergence behaviour for valve with contact problem, shown here for time step 0.11s for (a) soft and (b) hard contact.

purpose of the analysis was to evaluate the performance of the coupling methods in the presence of the additional non-linearities in these problems. Of the three classes of problems, only solid body contact posed any real difficulties. Our findings suggest that the non-linearity of the contact forces is too large for QN methods to deal with it in a black-box setting. These contact forces therefore need to be included within the QN approximations, in which case the convergence behaviour of the MVQN method is rendered comparable to those observed in preceding chapters.

Chapter 9

Conclusion

In this study, the development of a strongly coupled, partitioned FSI solver has been presented, with a particular focus on methods which enable the use of black-box solvers. The increasing trend within the FSI community is that monolithic, or partitioned solvers with exact sensitivities, are the preferred solution methods for strongly coupled, incompressible FSI problems. The potential merits offered by allowing for the re-use of field specific solvers is, however, sufficiently attractive to warrant additional research into partitioned solution methods.

This work focused on three primary areas of interest: a new coupling acceleration scheme, an improved solution for fully-enclosed FSI domains and demonstration of the feasibility of consistent interface information transfer.

In this study, interface information transfer was performed using radial basis function interpolation. We focused on a comparison between a conservative and a consistent formulation. While the conservative formulation is the preferred formulation in much of the literature, we have confirmed the findings in [22], demonstrating the conservative formulation to be a 0-order method. The conservative approach should therefore be approached with caution. Similarly, the consistent formulation, while not provably conservative, does converge within the limit of mesh refinement. In this regard, we share the views of Bathe *et al.* [6], that the patch test should be treated as an important FSI benchmark problem.

Radial basis function interpolation, with consistent surface integration, was demonstrated to be a sufficiently accurate means to transfer information across a non-matching, non-conforming interface; both the absolute errors, and convergence rates were below

the errors introduced by the linear fluid solver. One limitation of the formulation, as posed in this study, is the inability to exactly transfer information across matching interface meshes, when transferring between finite volume and finite element methods. While generating matching interface grids can be a complex operation, it is achievable. It may therefore serve as a valuable means of guaranteeing that no errors (or acceptably small errors) are generated, thereby minimising one source of numerical errors within an FSI simulation. For the consistent formulation, exact transfer of information is only possible when transferring constant fields.

A new "multi-vector, iteratively updated quasi-Newton" method has been introduced and investigated. The method can be summarised as finding the matrix which satisfies the interface secant conditions, by finding the minimum modification to the Jacobian, from the previous time step, using all observations from the current time step. In many ways, the MVQN method can be regarded as the combination of the current state-of-the-art IBQN-LS method and the classical Broyden's method. The result is a heuristic independent, black-box coupling scheme which demonstrates both improved robustness and coupling efficiency. The newly proposed MVQN method was compared to the IBQN-LS method, Broyden's method and Aitken's dynamic relaxation method, on a host of strongly coupled FSI problems. The FSI problems analysed cover a wide spectrum of relevance and complexity, and provide some confidence in the general applicability of the new coupling scheme.

While the newly proposed MVQN method is a clear improvement over existing black-box coupling methods, no comparison was made with monolithic solution schemes. Early indications, based on the near Newton-like convergence behaviour, and by comparison with similar related studies, suggest that the MVQN method can provide comparable performance. A monolithic scheme will, however, undoubtedly outperform any black-box partitioned FSI solver, when applied to strongly coupled problems. Comparable performance, with the modularity on offer, and re-usability of (virtually any) field solvers, is nonetheless, an attractive proposition. The strength of the argument in favour of (or against) the MVQN method would benefit from a proper comparison with a monolithic solution scheme.

We have further proposed the combination of the MVQN method with artificial compressibility. AC/FSI is one potential solution to the incompressibility dilemma for fully-enclosed FSI problems (when solved using partitioned schemes). Many of the

limitations of AC/FSI have been highlighted in this work, and the improvement offered by the proposed combination of methods has been demonstrated.

In the final chapter we provided additional examples of the potential merits of the coupled environment. Three FSI examples, each containing additional sources of non-linearities, were introduced and analysed. Of the three additional problem classes, only FSI with solid-body contact provided any real complications.

One omission from the current study is a stronger theoretical evaluation of the coupled environment. While the detailed study presented here provides confidence in the suitability of the MVQN for partitioned FSI (via extensive numerical experiments), there is as yet no formal proof of convergence. Preliminary analysis suggests, however, that such a proof will most readily be achieved for linear systems. While the usefulness of such an analysis is limited in the context of FSI problems, which are highly non-linear, it could provide some insight into the extent of its applicability. It would furthermore be of interest to determine whether the MVQN can be applied, with similar success, to alternative multi-physics problems; or to the solution of more general domain-decomposition methods.

At this stage, the accuracy of these approximated QN interface sensitivity matrices is an open question. Based on the performance in the partitioned FSI environment, to obtain super-linear (or at times, near quadratic) convergence rates, we can only infer that they are fairly accurate. Are they however sufficiently accurate to be re-used for extended applications? For example, can these QN interface sensitivities be used to couple design variables in one sub-domain to the other, for application to FSI optimisation problems?

Overall, we believe the work presented in this thesis to be a valuable addition to the field of partitioned FSI. The coupled environment proposed in this study provides an efficient, and reliable means, of solving for FSI problems through the re-use of existing field solvers.

Appendix A

MVQN Algorithm

Algorithm A.1 MVQN: Multi-vector update quasi-Newton method applied to the block-Newton equations (5.8) and (5.9).

1: $t := 0$; $n := 0$
2: $\boldsymbol{f}_1^{F,0} := \boldsymbol{F}\left(\boldsymbol{d}_0^0\right)$; $\boldsymbol{f}_1^{S,0} := \boldsymbol{f}_1^{F,0}$; $\boldsymbol{d}_1^{S,0} := \boldsymbol{S}\left(\boldsymbol{f}_1^{S,0}\right)$
3: $\boldsymbol{r}_1^0 := \boldsymbol{d}_1^{S,0} - \boldsymbol{d}_0^0$; $\boldsymbol{J}_F^0 := [\boldsymbol{0}]$; $\boldsymbol{J}_S^0 := [\boldsymbol{0}]$
4: **while** $t < t_{\text{end}}$ **do**:
5: $\quad k := 0$
6: $\quad$ **while** $||\boldsymbol{r}_{k+1}^{n+1}|| > \epsilon$ **do**:
7: $\quad\quad$ **if** ($k = 0$ and $n = 0$) **then**:
8: $\quad\quad\quad \boldsymbol{d}_1^{S,0} := \boldsymbol{d}_1^{S,0} + \omega_0 \boldsymbol{r}_1^0$
9: $\quad\quad$ **else**:
10: $\quad\quad\quad \boldsymbol{f}_{k+1}^{F,n+1} := \boldsymbol{F}\left(\boldsymbol{d}_k^{F,n+1}\right)$
11: $\quad\quad\quad$ **if** ($k > 0$) **then**:
12: $\quad\quad\quad\quad$ compute $\boldsymbol{J}_F^{n+1}$ using (5.18)
13: $\quad\quad\quad$ **end if** ($k > 0$) **then**:
14: $\quad\quad\quad$ solve for $\Delta \boldsymbol{f}$: equation (5.8)

$$\text{i.e. } \left(\boldsymbol{J}_F \boldsymbol{J}_S - \boldsymbol{I}\right) \Delta \boldsymbol{f} = -\left(\boldsymbol{f}_{k+1}^{F,n+1} - \boldsymbol{f}_k^{S,n+1}\right) - \boldsymbol{J}_F \left(\boldsymbol{d}_k^{F,n+1} - \boldsymbol{d}_k^{S,n+1}\right)$$

15: $\quad\quad\quad \boldsymbol{f}_{k+1}^{S,n+1} := \boldsymbol{f}_k^{S,n+1} + \Delta \boldsymbol{f}$
16: $\quad\quad\quad \boldsymbol{d}_{k+1}^{S,n+1} := \boldsymbol{S}\left(\boldsymbol{f}_{k+1}^{S,n+1}\right)$

Algorithm A.1 (Continued)

if $(k > 0)$ **then**:
  compute $\boldsymbol{J}_S^{n+1}$ using (5.19)
end if
solve for $\Delta \boldsymbol{d}$: equation (5.9)
  i.e. $\left(\boldsymbol{J}_S \boldsymbol{J}_F - \boldsymbol{I}\right) \Delta \boldsymbol{d} = -\left(\boldsymbol{d}_{k+1}^{S,n+1} - \boldsymbol{d}_k^{F,n+1}\right) - \boldsymbol{J}_S \left(\boldsymbol{f}_{k+1}^{S,n+1} - \boldsymbol{f}_{k+1}^{F,n+1}\right)$
$\boldsymbol{d}_{k+1}^{F,n+1} := \boldsymbol{d}_k^{F,n+1} + \Delta \boldsymbol{d}$
end if
$k := k + 1$
end while
$t := t + \Delta t$; $n := n + 1$
end while

Appendix B

IAC/FSI + MVQN Algorithm

Algorithm B.1 IAC/FSI+MVQN algorithm.

$t := 0$; $n := 0$

$\boldsymbol{f}_1^{F,0} := \boldsymbol{F}_{AC}\left(\boldsymbol{d}_0^0\right)$; $\boldsymbol{f}_1^{S,0} := \boldsymbol{f}_1^{F,0}$; $\boldsymbol{d}_1^{S,0} := \boldsymbol{S}\left(\boldsymbol{f}_1^{S,0}\right)$

$\boldsymbol{r}_1^0 := \boldsymbol{d}_1^{S,0} - \boldsymbol{d}_0^0$; $\boldsymbol{J}_F^0 := [\boldsymbol{0}]$; $\boldsymbol{J}_S^0 := [\boldsymbol{0}]$

while $t < t_{\text{end}}$ **do**:

  $k := 0$

  while $||\boldsymbol{r}_{k+1}^{n+1}|| > \epsilon$ **do**:

    approximate $\boldsymbol{\beta}$ using (7.6) or (7.7)

    $\boldsymbol{f}_{k+1}^{F,n+1} := \boldsymbol{F}_{AC}\left(\boldsymbol{d}_k^{F,n+1}\right)$

    if $(k > 0)$ **then**:

      compute $\boldsymbol{J}_F^{n+1}$

    end if

    if $(k > 1$ or $n > 0)$ **then**:

      solve for $\Delta\boldsymbol{f}$: equation (7.10)

      i.e. $\left(\boldsymbol{J}_F\boldsymbol{J}_S - \boldsymbol{I}\right)\Delta\boldsymbol{f} = -\left(\boldsymbol{f}_{k+1}^{F,n+1} - \boldsymbol{f}_k^{S,n+1}\right) - \boldsymbol{J}_F\left(\boldsymbol{d}_k^{F,n+1} - \boldsymbol{d}_k^{S,n+1}\right)$

    $\boldsymbol{p}_k^{F,n+1} := \boldsymbol{p}_k^{F,n+1} + \Delta\boldsymbol{f} \cdot n_\Gamma$

    $\boldsymbol{f}_{k+1}^{S,n+1} := \boldsymbol{f}_k^{S,n+1} + \Delta\boldsymbol{f}$

Algorithm B.1 (Continued)

16: **else:**
17: $\boldsymbol{f}_{k+1}^{S,n+1} := \boldsymbol{f}_{k+1}^{F,n+1}$
18: **end if**
19: $\boldsymbol{d}_{k+1}^{S,n+1} := \boldsymbol{S}\left(\boldsymbol{f}_{k+1}^{S,n+1}\right)$
20: **if** $(k > 0)$ **then**:
21: compute $\boldsymbol{J}_S^{n+1}$ using (5.19)
22: **end if**
23: $\boldsymbol{d}_{k+1}^{F,n+1} := \boldsymbol{d}_{k+1}^{S,n+1}$
24: $k = k + 1$
25: **end while**
26: $t := t + \Delta t$; $n := n + 1$
27: **end while**

Bibliography

[1] Fluent user's guide, Fluent Inc, Lebanon, 2005.

[2] OpenFOAM: The Open Source CFD Toolbox User Guide, Version 2.1.0, 2010.

[3] M Astorino, JF Gerbeau, O Pantz, and KF Traore. Fluid–structure interaction and multi-body contact: application to aortic valves. *Computer Methods in Applied Mechanics and Engineering*, 198(45):3603–3612, 2009.

[4] FPT Baaijens. A fictitious domain/mortar element method for fluid-structure interaction. *International Journal for Numerical Methods in Fluids*, 35(7):743–761, 2001.

[5] S Badia, F Nobile, and C Vergara. Fluid–structure partitioned procedures based on Robin transmission conditions. *Journal of Computational Physics*, 227(14):7027–7051, 2008.

[6] KJ Bathe and GA Ledezma. Benchmark problems for incompressible fluid flows with structural interactions. *Computers & Structures*, 85(11):628–644, 2007.

[7] A Beckert and H Wendland. Multivariate interpolation for fluid-structure-interaction problems using radial basis functions. *Aerospace Science and Technology*, 5(2):125–134, 2001.

[8] C Bernardi, Y Maday, and AT Patera. A new nonconforming approach to domain decomposition: the mortar element method. *Nonliner Partial Differential Equations and Their Applications*, 1994.

[9] FJ Blom. A monolithical fluid-structure interaction algorithm applied to the piston problem. *Computer Methods in Applied Mechanics and Engineering*, 167(3):369–391, 1998.

[10] AEJ Bogaers. Reduced order modeling techniques for mesh movement as applied to fluid structure interactions. Master's thesis, University of Pretoria, 2010.

[11] AEJ Bogaers, S Kok, BD Reddy, and T Franz. Treatment of fully enclosed Fluid Structure Interactions using artificial compressibility. In *Third African Congress on Computational Mechanics, AfriCOMP2013, Livingstone, Zambia*, July-August 2013.

[12] AEJ Bogaers, S Kok, BD Reddy, and T Franz. Quasi-Newton methods for implicit black-box FSI coupling. *Computer Methods in Applied Mechanics and Engineering*, 279:113–132, 2014.

[13] AEJ Bogaers, S Kok, BD Reddy, and T Franz. Extending the robustness and efficiency of artificial compressibility for partitioned fluid–structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 283:1278–1295, 2015.

[14] CG Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of Computation*, 19(92):577–593, 1965.

[15] P Causin, JF Gerbeau, and F Nobile. Added-mass effect in the design of partitioned algorithms for fluid–structure problems. *Computer Methods in Applied Mechanics and Engineering*, 194(42):4506–4527, 2005.

[16] JR Cebral and R Lohner. Conservative load projection and tracking for fluid-structure problems. *AIAA journal*, 35(4):687–692, 1997.

[17] AJ Chorin. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics*, 2(1):12–26, 1967.

[18] J Chung and GM Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method. *Journal of Applied Mechanics*, 60(2):371–375, 1993.

[19] RD Cook, DS Malkus, ME Plesha, and RJ Witt. *Concepts and Applications of Finite Element Analysis*. John Wiley & Sons, 2007.

[20] A De Boer, MS Van der Schoot, and H Bijl. Mesh deformation based on radial basis function interpolation. *Computers & Structures*, 85(11):784–795, 2007.

[21] A De Boer, AH Van Zuijlen, and H Bijl. Review of coupling methods for non-matching meshes. *Computer Methods in Applied Mechanics and Engineering*, 196(8):1515–1525, 2007.

[22] A de Boer, AH van Zuijlen, and H Bijl. Comparison of conservative and consistent approaches for the coupling of non-matching meshes. *Computer Methods in Applied Mechanics and Engineering*, 197(49):4284–4297, 2008.

[23] J De Hart, FPT Baaijens, GWM Peters, and PJG Schreurs. A computational fluid-structure interaction analysis of a fiber-reinforced stentless aortic valve. *Journal of Biomechanics*, 36(5):699–712, 2003.

[24] J Degroote. On the similarity between Dirichlet–Neumann with interface artificial compressibility and Robin–neumann schemes for the solution of fluid-structure interaction problems. *Journal of Computational Physics*, 230(17):6399–6403, 2011.

[25] J Degroote, KJ Bathe, and J Vierendeels. Performance of a new partitioned procedure versus a monolithic procedure in fluid–structure interaction. *Computers & Structures*, 87(11):793–801, 2009.

[26] J Degroote, R Haelterman, S Annerel, P Bruggeman, and J Vierendeels. Performance of partitioned procedures in fluid–structure interaction. *Computers & Structures*, 88(7):446–457, 2010.

[27] J Degroote, A Souto-Iglesias, W Van Paepegem, S Annerel, P Bruggeman, and J Vierendeels. Partitioned simulation of the interaction between an elastic structure and free surface flow. *Computer Methods in Applied Mechanics and Engineering*, 199(33):2085–2098, 2010.

[28] J Degroote, A Swillens, P Bruggeman, R Haelterman, P Segers, and J Vierendeels. Simulation of fluid–structure interaction with the interface artificial com-

pressibility method. *International Journal for Numerical Methods in Biomedical Engineering*, 26(3-4):276–289, 2010.

[29] J Degroote and J Vierendeels. Multi-level quasi-Newton coupling algorithms for the partitioned simulation of fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 225:14–27, 2012.

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

[31] S Deparis, M Discacciati, G Fourestey, and A Quarteroni. Fluid–structure algorithms based on Steklov–Poincaré operators. *Computer Methods in Applied Mechanics and Engineering*, 195(41):5797–5812, 2006.

[32] WG Dettmer and D Perić. On the coupling between fluid flow and mesh motion in the modelling of fluid–structure interaction. *Computational Mechanics*, 43(1):81–90, 2008.

[33] G Dhondt. Calculix crunchix user's manual version 2.5. 2007.

[34] D Dureisseix and H Bavestrello. Information transfer between incompatible finite element meshes: application to coupled thermo-viscoelasticity. *Computer Methods in Applied Mechanics and Engineering*, 195(44):6523–6541, 2006.

[35] C Farhat, C Degand, B Koobus, and M Lesoinne. Torsional springs for two-dimensional dynamic unstructured fluid meshes. *Computer Methods in Applied Mechanics and Engineering*, 163(1):231–245, 1998.

[36] 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*, 174(2):669–694, 2001.

[37] C Farhat, M Lesoinne, and P Le Tallec. Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 157(1):95–114, 1998.

[38] C Farhat, KG Van der Zee, and P Geuzaine. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 195(17):1973–2001, 2006.

[39] MÁ Fernández and M Moubachir. A newton method using exact Jacobians for solving fluid–structure coupling. *Computers & Structures*, 83(2):127–142, 2005.

[40] JH Ferziger and M Perić. *Computational methods for fluid dynamics*, volume 3. Springer Berlin, 2002.

[41] B Fornberg, TA Driscoll, G Wright, and R Charles. Observations on the behavior of radial basis function approximations near boundaries. *Computers & Mathematics with Applications*, 43(3):473–490, 2002.

[42] C Förster, WA Wall, and E Ramm. Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 196(7):1278–1293, 2007.

[43] L Gerardo-Giorda, F Nobile, and C Vergara. Analysis and optimization of Robin-Robin partitioned procedures in fluid-structure interaction problems. *SIAM Journal on Numerical Analysis*, 48(6):2091–2116, 2010.

[44] JF Gerbeau and M Vidrascu. A quasi-Newton algorithm based on a reduced model for fluid-structure interaction problems in blood flows. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37(4):631–647, 2003.

[45] A Gerstenberger and WA Wall. An extended finite element method/lagrange multiplier based approach for fluid–structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 197(19):1699–1714, 2008.

[46] AJ Gil, A Arranz Carreño, J Bonet, and O Hassan. The immersed structural potential method for haemodynamic applications. *Journal of Computational Physics*, 229(22):8613–8641, 2010.

[47] R Haelterman. *Analytical study of the least squares quasi-Newton method for interaction problems*. PhD thesis, Ghent University, 2009.

[48] R Haelterman, J Degroote, D Van Heule, and J Vierendeels. The quasi-Newton least squares method: a new and fast secant method analyzed for linear systems. *SIAM Journal on Numerical Analysis*, 47(3):2347–2368, 2009.

[49] X Hansong. Analysis of Dirichlet-Neumann and Neumann-Dirichlet Partitioned Procedures in Fluid-Structure Interaction Problems. Master's thesis, National University of Singapore.

[50] M Heil, AL Hazel, and J Boyle. Solvers for large-displacement fluid–structure interaction problems: segregated versus monolithic approaches. *Computational Mechanics*, 43(1):91–101, 2008.

[51] BT Helenbrook. Mesh deformation using the biharmonic operator. *International Journal for Numerical Methods in Engineering*, 56(7):1007–1021, 2003.

[52] CW Hirt, AA Amsden, and JL Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 14(3):227–253, 1974.

[53] P Holmes, JL Lumley, and G Berkooz. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge university press, 1998.

[54] GA Holzapfel. *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*, volume 24. Wiley Chichester, 2000.

[55] B Hübner, E Walhorn, and D Dinkler. A monolithic approach to fluid–structure interaction using space–time finite elements. *Computer Methods in Applied Mechanics and Engineering*, 193(23):2087–2104, 2004.

[56] GM Hulbert and J Chung. Explicit time integration algorithms for structural dynamics with optimal numerical dissipation. *Computer Methods in Applied Mechanics and Engineering*, 137(2):175–188, 1996.

[57] D Ishihara and S Yoshimura. A monolithic approach for interaction of incompressible viscous fluid and an elastic body based on fluid pressure poisson equation. *International Journal for Numerical Methods in Engineering*, 64(2):167–203, 2005.

[58] RK Jaiman, X Jiao, PH Geubelle, and E Loth. Assessment of conservative load transfer for fluid–solid interface with non-matching meshes. *International journal for Numerical Methods in Engineering*, 64(15):2014–2038, 2005.

[59] RK Jaiman, X Jiao, PH Geubelle, and E Loth. Conservative load transfer along curved fluid–solid interface with non-matching meshes. *Journal of Computational Physics*, 218(1):372–397, 2006.

[60] E Järvinen, P Råback, M Lyly, and JP Salenius. A method for partitioned fluid–structure interaction computation of flow in arteries. *Medical Engineering & Physics*, 30(7):917–923, 2008.

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

[62] OL Jay and D Negrut. A second order extension of the generalized–α method for constrained systems in mechanics. In *Multibody Dynamics*, pages 143–158. Springer, 2009.

[63] X Jiao and MT Heath. Common-refinement-based data transfer between non-matching meshes in multiphysics simulations. *International Journal for Numerical Methods in Engineering*, 61(14):2402–2427, 2004.

[64] X Jiao and MT Heath. Overlaying surface meshes, Part I: Algorithms. *International Journal of Computational Geometry & Applications*, 14(06):379–402, 2004.

[65] X Jiao and MT Heath. Overlaying surface meshes, Part II: topology preservation and feature matching. *International Journal of Computational Geometry & Applications*, 14(06):403–419, 2004.

[66] FP Kärrholm. Rhie-chow interpolation in openfoam. Technical report, Department of Applied Mechanics, Chalmers Univesity of Technology, 2006.

[67] FP Kärrholm. *Numerical Modelling of Diesel Spray Injection, Turbulence Interaction and Combustion*. PhD thesis, Chalmers University of Technology, 2008.

[68] C Kassiotis. *Nonlinear fluid-structure interaction: a partitioned approach and its application through component technology.* PhD thesis, Université Paris-Est, 2009.

[69] C Kassiotis, A Ibrahimbegovic, R Niekamp, and HG Matthies. Nonlinear fluid–structure interaction problem. Part I: implicit partitioned algorithm, nonlinear stability proof and validation examples. *Computational Mechanics*, 47(3):305–323, 2011.

[70] T Kind. RAMDISK benchmarks. *University of California*, 2011.

[71] T Krishnamurthy. Comparison of response surface construction methods for derivative estimation using moving least squares, kriging and radial basis functions. In *Proceedings of the 46th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference, AIAA-2005-1821*, pages 18–21, 2005.

[72] U Küttler, C Förster, and WA Wall. A solution for the incompressibility dilemma in partitioned fluid–structure interaction with pure Dirichlet fluid domains. *Computational Mechanics*, 38(4-5):417–429, 2006.

[73] U Küttler, M Gee, C Förster, A Comerford, and WA Wall. Coupling strategies for biomedical fluid–structure interaction problems. *International Journal for Numerical Methods in Biomedical Engineering*, 26(3-4):305–321, 2010.

[74] M Lombardi, N Parolini, and A Quarteroni. Radial basis functions for inter-grid interpolation and mesh motion in FSI problems. *Computer Methods in Applied Mechanics and Engineering*, 256:117–131, 2013.

[75] AG Malan. *Investigation into the Continuum Thermodynamic Modeling of Investment Casting Shell-Mould Drying.* PhD thesis, University of Wales Swansea, 2002.

[76] LG Margolin and M Shashkov. Second-order sign-preserving conservative interpolation (remapping) on general grids. *Journal of Computational Physics*, 184(1):266–298, 2003.

[77] HG Matthies and J Steindorf. Partitioned strong coupling algorithms for fluid–structure interaction. *Computers & Structures*, 81(8):805–812, 2003.

[78] GCJ Morgan. *Application of the Interfoam VoF Code to Coastal Wave/Structure Interaction*. PhD thesis, University of Bath, 2013.

[79] SA Morton, RB Melville, and MR Visbal. Accuracy and coupling issues of aeroelastic Navier-Stokes solutions on deforming meshes. *Journal of Aircraft*, 35(5):798–805, 1998.

[80] F Nobile and C Vergara. Partitioned algorithms for fluid-structure interaction problems in haemodynamics. *Milan Journal of Mathematics*, 80(2):443–467, 2012.

[81] E Oñate and J Garcıa. A finite element method for fluid–structure interaction with surface waves using a finite calculus formulation. *Computer Methods in Applied Mechanics and Engineering*, 191(6):635–660, 2001.

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

[83] S Piperno, C Farhat, and B Larrouturou. Partitioned procedures for the transient solution of coupled aroelastic problems Part I: Model problem, theory and two-dimensional application. *Computer Methods in Applied Mechanics and Engineering*, 124(1):79–112, 1995.

[84] G Quaranta, P Masarati, and P Mantegazza. A conservative mesh-free approach for fluid-structure interface problems. In *International Conference for Coupled Problems in Science and Engineering, Greece*. Citeseer, 2005.

[85] P Raback, J Ruokolainen, M Lyly, and E Järvinen. Fluid-structure interaction boundary conditions by artificial compressibility. In *ECCOMAS CFD*, 2001.

[86] TCS Rendall and CB Allen. Unified fluid–structure interpolation and mesh motion using radial basis functions. *International Journal for Numerical Methods in Engineering*, 74(10):1519–1559, 2008.

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

[88] K Riemslagh, J Vierendeels, and E Dick. An efficient coupling procedure for flexible wall fluid-structure interaction. In *European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS: Barcelona, Spain*, volume 13, 2000.

[89] SM Rifai, Z Johan, WP Wang, JP Grisval, TJR Hughes, and RM Ferencz. Multiphysics simulation of flow-induced vibrations and aeroelasticity on parallel computing platforms. *Computer Methods in Applied Mechanics and Engineering*, 174(3):393–417, 1999.

[90] H Rusche. *Computational fluid dynamics of dispersed two-phase flows at high phase fractions*. PhD thesis, Imperial College London (University of London), 2003.

[91] TP Scholcz, AH van Zuijlen, and H Bijl. Space-mapping in fluid–structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 281:162–183, 2014.

[92] L Sirovich. Turbulence and the dynamics of coherent structures. I-Coherent structures. II-Symmetries and transformations. III-Dynamics and scaling. *Quarterly of Applied Mathematics*, 45:561–571, 1987.

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

[94] AK Slone, K Pericleous, C Bailey, M Cross, and C Bennett. A finite volume unstructured mesh approach to dynamic fluid–structure interaction: an assessment of the challenge of predicting the onset of flutter. *Applied Mathematical Modelling*, 28(2):211–239, 2004.

[95] MJ Smith, DH Hodges, and CE S. Cesnik. Evaluation of computational algorithms suitable for fluid-structure interactions. *Journal of Aircraft*, 37(2):282–294, 2000.

[96] K Stein, T Tezduyar, and R Benney. Mesh moving techniques for fluid-structure interactions with large displacements. *Journal of Applied Mechanics*, 70(1):58–63, 2003.

[97] KR Stein, RJ Benney, V Kalro, AA Johnson, and TE Tezduyar. Parallel computation of parachute fluid-structure interactions. *UMSI research report/University of Minnesota (Minneapolis, Mn). Supercomputer institute*, 97:54, 1997.

[98] R Suliman. Development of parallel strongly coupled hybrid fluid-structure interaction technology involving thin geometrically non-linear structures. Master's thesis, MEng Thesis, University of Pretoria, 2011.

[99] TE Tezduyar and S Sathe. Modelling of fluid–structure interactions with the space–time finite elements: solution techniques. *International Journal for Numerical Methods in Fluids*, 54(6-8):855–900, 2007.

[100] TE Tezduyar, S Sathe, J Pausewang, M Schwaab, J Christopher, and J Crabtree. Interface projection techniques for fluid–structure interaction modeling with moving-mesh methods. *Computational Mechanics*, 43(1):39–49, 2008.

[101] P Thévenaz, T Blu, and M Unser. Interpolation revisited [medical images application]. *Medical Imaging, IEEE Transactions on*, 19(7):739–758, 2000.

[102] PD Thomas and CK Lombard. Geometric conservation law and its application to flow computations on moving grids. *AIAA journal*, 17(10):1030–1037, 1979.

[103] R Torii, M Oshima, T Kobayashi, K Takagi, and TE Tezduyar. Computer modeling of cardiovascular fluid–structure interactions with the deforming-spatial-domain/stabilized space–time formulation. *Computer Methods in Applied Mechanics and Engineering*, 195(13):1885–1895, 2006.

[104] S Turek. *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach*, volume 6. Springer, 1999.

[105] S Turek and J Hron. *Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow.* Springer, 2006.

[106] O Ubbink. *Numerical prediction of two fluid systems with sharp interfaces*. PhD thesis, University of London UK, 1997.

[107] R van Loon, PD Anderson, and FN van de Vosse. A fluid–structure interaction method with solid-rigid contact for heart valve dynamics. *Journal of Computational Physics*, 217(2):806–823, 2006.

[108] ABAQUS Version. 6.7. users manual. *Inc. and Dassault systemes*, 2007.

[109] HK Versteeg and W Malalasekera. *An introduction to computational fluid dynamics: the finite volume method*. Pearson Education, 2007.

[110] J Vierendeels, L Lanoye, J Degroote, and P Verdonck. Implicit coupling of partitioned fluid–structure interaction problems with reduced order models. *Computers & Structures*, 85(11):970–976, 2007.

[111] E Walhorn, A Kölke, B Hübner, and D Dinkler. Fluid–structure coupling within a monolithic model involving free surface flows. *Computers & structures*, 83(25):2100–2111, 2005.

[112] WA Wall. *Fluid-Struktur-Interaktion mit stabilisierten Finiten Elementen*. PhD thesis, Institut für Baustatik, Universität Stuttgart, 1999.

[113] WA Wall, S Genkinger, and E Ramm. A strong coupling partitioned approach for fluid–structure interaction with free surfaces. *Computers & Fluids*, 36(1):169–183, 2007.

[114] H Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4(1):389–396, 1995.

[115] H Wendland. Konstruktion und untersuchung radialer basisfunktionen mit kompaktem träger. *Göttingen, Georg-August-Universität zu Göttingen, Diss*, 1996.

[116] BI Wohlmuth. A mortar finite element method using dual spaces for the Lagrange multiplier. *SIAM Journal on Numerical Analysis*, 38(3):989–1012, 2000.

[117] BJBM Wolters, MCM Rutten, GWH Schurink, U Kose, J De Hart, and FN Van De Vosse. A patient-specific computational model of fluid–structure interaction in abdominal aortic aneurysms. *Medical Engineering & Physics*, 27(10):871–883, 2005.

[118] GH Yoon. Stress-based topology optimization method for steady-state fluid–structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 278:499–523, 2014.

[119] SZ Zhao, XY Xu, AD Hughes, SA Thom, AV Stanton, B Ariff, and Q Long. Blood flow and vessel mechanics in a physiologically realistic model of a human carotid arterial bifurcation. *Journal of Biomechanics*, 33(8):975–984, 2000.

[120] OC Zienkiewicz and RL Taylor. *The finite element method: Solid mechanics*, volume 2. Butterworth-heinemann, 2000.