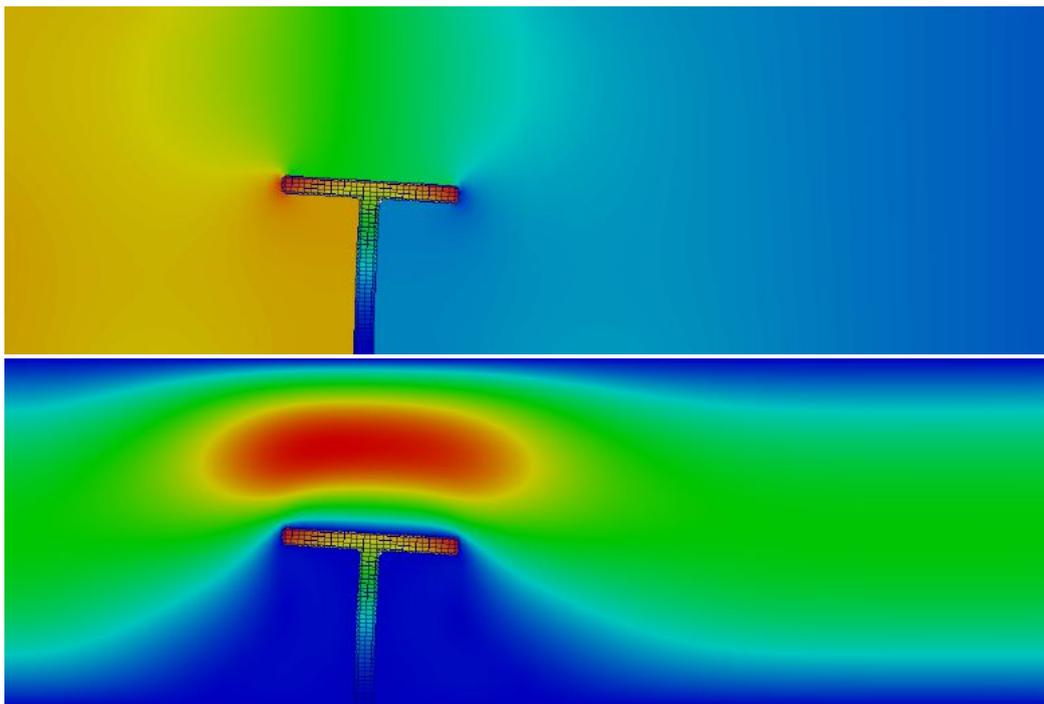


Levelset based fluid-structure interaction modeling with the eXtended Finite Element Method

An explorative research with a Newton-Raphson solver

Thijs Bosma

Master of Science Thesis



Levelset based fluid-structure interaction modeling with the eXtended Finite Element Method

An explorative research with a Newton-Raphson solver

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Mechanical Engineering at Delft
University of Technology

Thijs Bosma

November 28, 2013

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of
Technology



The work in this thesis was supported by the Center of Aerospace Structures (CAS) at the University of Colorado, Boulder, USA. Their cooperation is hereby gratefully acknowledged.



Copyright © Precision and Microsystems Engineering (PME)
All rights reserved.

*PME - the Ultimate in
Mechanical Engineering*

Abstract

This study focuses on solving steady state fluid-structure interaction (FSI) problems subjected to incompressible flow at low Reynolds numbers and linear elastic structural behavior. The fluid-structure interface geometry is described with a levelset method (LSM). The governing equations are discretized in space with the eXtended Finite Element Method (XFEM). At the fluid-structure interface, a traction and a no-slip condition are enforced to model the interaction between the two phases. Combining XFEM and LSM gives crisp and clear physical behavior at the fluid-structure interface, which is beneficial for topology optimization purposes. Two different solvers, a staggered and a monolithic solver, are used to test the accuracy of the results of the steady state solution and to check whether a numerically consistent Jacobian is built. The results show that the implementation needs to be improved in order to monolithically solve the system successfully. Finite differences are used to check the consistency of the Jacobian and show that Jacobian terms related to the levelset field are inconsistent. The proposed LSM introduces secondary coupling between neighboring elements, which is not correctly handled in the current implementation.

Contents

Acknowledgements	ix
1 Introduction	1
2 General Concepts	3
2-1 eXtended Finite Element Method (XFEM)	3
2-1-1 Introduction to Finite Element Method (FEM)	4
2-1-2 Extending FEM to XFEM	7
2-2 Levelset method (LSM)	11
2-2-1 The concept of levelset methods	11
2-3 Non-linear solver - Newton-Raphson	12
2-4 Differentiation in numerical modeling	13
2-4-1 Finite Differences	14
2-4-2 Discrete analytical derivatives - Adjoint formulation	15
2-5 Summary	16
3 The complete XFEM model	17
3-1 Levelset procedure	17
3-1-1 Separate meshes - 3-field setup	17
3-1-2 Levelset initialization	19
3-1-3 Tracking structural interface deformation	20
3-1-4 Fluid levelset update	23
3-1-5 Fluid levelset field derivatives	27
3-1-6 Reflection on proposed levelset method	31
3-2 Fluid Structure Interaction	33
3-2-1 Mechanics	33
3-2-2 Initialization and boundary conditions	37

3-2-3	FSI coupling in XFEM	38
3-3	Consistent Jacobian	45
3-3-1	Jacobian overview	45
3-3-2	Analytical terms	47
3-4	FSI solvers	48
3-4-1	Monolithic solver	49
3-4-2	Staggered solver	50
3-5	The problem setup	51
3-5-1	Boundary conditions	51
3-5-2	Material parameters	52
3-5-3	Initial configuration	53
3-6	Summary	54
4	Model verification - Staggered setup	56
4-1	Results - Staggered setup	56
4-1-1	Steady state solution - XFEM model	56
4-1-2	Steady state solution - COMSOL benchmark	59
4-2	Solver studies	60
4-2-1	Solver stages	61
4-2-2	Convergence plots	62
4-2-3	Parameter studies	64
4-3	Analysis - Staggered setup	66
4-3-1	COMSOL comparison	66
4-3-2	Convergence in XFEM models with changing DOFs	68
4-4	Summary	71
5	Model verification - Monolithic setup	73
5-1	Results - Monolithic setup	73
5-2	Jacobian - 2 element Finite Differences	73
5-2-1	The finite differences process	75
5-2-2	Methods to compare Jacobians	75
5-3	Finite Differences - Results	76
5-3-1	Two elements with only left element intersected	76
5-3-2	Two elements with both elements intersected	78
5-3-3	Detailed analysis of results	81
5-3-4	Choosing a correct 2 element problem	88
5-4	Summary	89
6	Discussion	91
7	Conclusions and Recommendations	93

A	Derivation of weak form of non dimensional Navier-Stokes equations	97
B	No-slip condition - Condensing out the additional stress field in intersected elements	100
C	Algorithms	102
D	MATLAB code for fluid residual to structural displacements Jacobian	104
E	Spy-plots for the two elements, two intersections case	116
F	Relative residual plots - Staggered setup	119
	Bibliography	120
	Glossary	122
	List of Acronyms	122

List of Figures

2-1	Discontinuity within a Finite Element	4
2-2	The 4-node bi-linear quadrilateral Q4 element [Felippa, 2013] - η and ξ are the natural coordinates and range from -1 to 1	6
2-3	The bi-linear quadrilateral Q4 element with a discontinuity inside the element - The discontinuity may, for example, be a transition from solid material (grey area) to fluid material (red area).	7
2-4	Standard FE interpolation term $\sum_{i=1}^n N_i(\mathbf{x})u_i$ - See also remark 1	9
2-5	The enrichment term $\sum_{i=1}^n N_i(\mathbf{x})E(\mathbf{x})q_i$ with the Heaviside enrichment H - See also remark 2	10
2-6	Illustration of Partition of Unity concept - The figures on the left show a truss element and associated discontinuous shape functions. The figures on the right show the physical situation based on unit Heaviside displacement and unit displacement fields - See also remark 3.	10
2-7	Illustration of the levelset method (LSM) to describe geometries	12
2-8	Optimized geometries for a classic L-bracket problem - Levelset based interface on the left, density based on the right with unphysical intermediate densities [James and Martins, 2012]	13
2-9	A 1D Newton-Raphson update process - http://commons.wikimedia.org/wiki/File:Newton-Raphson_method.png	13
2-10	Finite Differences - The purple line shows the slope of analytic derivative $df(x)/dx$ at point x , the green line the slope according to backward differences in Eq. (2-19), the blue line according to central differences in Eq. (2-20) and the red line according to forward differences in Eq. (2-18).	14
3-1	3-field setup - The fluid domain is on the left, the structural domain on the right. The arrows indicate the transfer of FSI coupling information from one domain to the other through the structural bars; \mathbf{t}_f and \mathbf{u}_s traction and structural displacements and velocities, respectively. The grey areas represent voids.	18
3-2	Fixed fluid mesh with zero levelset contour in blue lines to indicate fluid to void edge as fluid-structure interface - Inside the contour lies the void domain.	19

3-3	Deformed structural mesh with zero levelset contour in blue lines to indicate structure to void edge as fluid-structure interface - Outside the contour lies the void domain	19
3-4	Deformed structure projected onto fixed fluid mesh, such that all void areas are filled with the other phase and physical FSI domain is complete.	20
3-5	Initialization of the levelset field - Dashed blue line is the actual geometry, the continuous blue line is the discretized geometry, which also determines the structural bars between the intersection points (the red dots). Parameters h , w , $r = w/2$ and cp indicate the height, the width, the radius and center point of the top of the structural beam, respectively. See also Figure 3-23 for the complete problem setup.	21
3-6	An intersected <i>structural</i> element considered in undeformed and initial configuration - The red dots are the intersection points between the zero contour and the structural element edges. These red dots will also be defined as the bar ending points.	21
3-7	The updated location of the structural bar within the <i>fluid</i> element that coincided with the undeformed structural element from Figure 3-6 - The bar ending points have moved (green dots) and hence also the intersection points (red dots).	22
3-8	Projected deformed structure on fluid mesh with bar endpoints as green dots, intersections as red dots, Gauss points as blue dots and the red dashed line as the interface between the projected intersection points	24
3-9	Bar intersection with fluid element edge - s and r are local coordinates of the intersection point along the element edge and the bar, respectively. The configuration is chosen arbitrarily, just to illustrate how the structural bars may intersect 1 of the 4 element edges.	25
3-10	Nodal perpendicular distances (green lines) to the interfaces between the projected intersection points.	26
3-11	Overview update process of the fluid levelset field - The undeformed configuration on the left and a deformed configuration on the right, where the same intersected fluid element is shown but the structural projection has changed	27
3-12	Mesh discrepancy - Grey is void in fluid mesh and does not align perfectly with the white wire frame of the structure. The discrepancy is clearly visible, especially at the top of the beam, where the structural details are finer.	31
3-13	Problematic intersections - Double intersections on one element edge in red circle and submersed bar in yellow circle	32
3-14	Refined mesh to handle multiple intersection case - The structure has a smoother interface, which is less likely to create a problematic intersection configuration.	32
3-15	Intersected element with embedded boundary Γ^+	38
3-16	Triangulation of intersected elements into 4 areas for integration purposes	39
3-17	Gauss points (blue dots) on the interface for numerical integration of the no-slip condition in fluid elements.	41
3-18	Projected Gauss points (blue dots) on the interface for numerical integration of the no-slip condition in fluid elements with a discrepancy between the fluid interface and the actual location of the structural bars.	41
3-19	Gauss points (blue dots) on structural bar and the bar is defined between the intersections points P_1^Γ and P_2^Γ within a fluid element	42
3-20	Projected Gauss points (blue dots) on the structural bars for numerical integration of the traction condition in structural elements with a discrepancy between the fluid interface and the actual location of the structural bars.	43

3-21	Code overview monolithic setup - The great majority of this MATLAB code was developed at the research group of Prof. Kurt Maute at the University of Colorado, Boulder, CO, USA	49
3-22	Staggered setup - Solid arrows indicate solution updates, dashed arrows indicate coupling information transfer. Individual Newton-Raphson updates are indicated with the small dots along the solid arrows.	50
3-23	Geometry of the benchmark problem incl. boundary conditions at the in- and outlet - Dimensions in μm	52
3-24	Initial levelset field with zero contour in white to depict initial geometry - The color legend shows the signed distance nodal levelset value.	53
4-1	Steady state solution XFEM model - Fluid velocity field and structural displacement with structural mesh	57
4-2	Steady state pressure field XFEM model	58
4-3	Steady state velocity and displacement field in x- and y-direction XFEM model	59
4-4	Steady state solutions - COMSOL	60
4-5	Steady state velocity and displacement field in x- and y-direction - COMSOL	61
4-6	A typical staggered solving process	62
4-7	Convergence plots of the staggered solve process	63
4-8	Norm of the single phase update vectors - The fluid update vector contains velocity and pressure terms and the structural update vector only displacements	65
4-9	Changed fluid DOFs after a fluid LSF update - Both the number and individual nodes changed.	70
4-10	The pressure field plotted right after the fluid levelset update in the second fluid update stage - A weird solution is visible in the area where the undeformed structure was located. Furthermore the pressure field has not adapted yet to the new configuration.	71
5-1	The exploded solution of the monolithic solver	74
5-2	Simplified FSI problems - Finite difference configurations with two elements	75
5-3	Spy plots and Frobenius norm of the complete analytic and FD Jacobian for two elements problem with left element intersected	76
5-4	Two elements, only left element intersected - Spy plot and Frobenius norm of the analytic structural Jacobian, FD Jacobian and FD traction Jacobian. The Frobenius norm of the FD traction Jacobian is negligible compared to the other Frob. norms.	77
5-5	Two elements, both intersections - Spy plot and Frobenius norm of the analytic structural \mathbf{J}_{fs} and the FD \mathbf{J}_{fs} . Both entries and Frob. norms differ.	78
5-6	A schematic overview of the two element Finite Difference setup of Figure 5-2b - For more information on the content of this figure the reader is referred to items 1 to 10	80
5-7	Two element setup with x-displacement of node 1 perturbed - Superscript * indicates a changed variable, due to a structural perturbation of node 1. The grey lines and dots in the background depict the Figure 5-6.	84
5-8	Mismatch between the projected intersection points and the zero contour intersection points for 2 element case - The projected intersection point \mathbf{P}_1^Γ does not coincide with the zero contour intersection point \mathbf{P}_1^ϕ , because the nodal absolute nodal levelset value d_2 is not equal to $(\mathbf{P}_2^n - \mathbf{P}_1^\Gamma)$	86

5-9	Zero contour and projection points mismatch at the top of the structural beam - The fluid residual relates to the purple interface, while the levelset field relates to the blue interface.	87
5-10	Symmetric two element setup with middle bottom node having equal orthogonal distance within both elements	89
5-11	The orthogonal distance to the closest interface from upper left node is depicted by the black dashed line, however, the min-operation sets it to the length of the green dashed arrow	89
6-1	Two beams fixed at the bottom wall	92
6-2	One beam fixed on bottom wall, one on top wall	92
6-3	FSI simulation of the capital letter T	92
E-1	Two elements, two intersections - Spy plot and Frobenius norm of the complete analytic Jacobian, FD Jacobian	116
E-2	Two elements, two intersections - Spy plot and Frobenius norm of the analytic structural Jacobian, FD structural Jacobian and FD traction Jacobian	117
E-3	Two elements, two intersections \mathbf{J}_{sf} - Spy plot and Frobenius norm of the analytic structural Jacobian and FD Jacobian	117
E-4	Two elements, two intersections \mathbf{J}_{ff} - Spy plot and Frobenius norm of the analytic structural Jacobian and FD Jacobian	118
F-1	Convergence plots - Relative to the residuals of the solution after the first fluid Newton-Raphson update	119

List of Tables

4-1	Results of mesh refinement	65
4-2	Under-relaxtion parameter α_s for the structure	66
4-3	The extreme values of the steady state solutions - The XFEM-results are dimensionalized according to App. A	67
5-1	Two elements, one intersection - Frobenius norms of the <i>difference</i> between the analytic and FD Jacobian of each individual part of the Jacobian (* The missing traction contribution has been accounted for, see Figure 5-4)	77
5-2	FD on levelset field for both elements - The sensitivities of nodal levelset values ϕ_i with respect to all structural displacements <i>in x-direction</i>	79
5-3	Analytic levelset field for element 1 - The sensitivities of nodal levelset values ϕ_i with respect to all structural displacements <i>in x-direction</i> . The red filled cells contain different numbers than the FD results of Table 5-2. The green cells are correct for element 1, but incorrect for element 2, see Table 5-4.	79
5-4	Analytic levelset field for element 2 - The sensitivities of nodal levelset values ϕ_i with respect to all structural displacements <i>in x-direction</i> . The red filled cells contain different numbers than the FD results of Table 5-2. The green cells are correct for element 2, but incorrect for element 1, see Table 5-3.	80
5-5	Nodal levelset dependencies on projected intersection points and structural displacements - The rows starting with the blue filled cells contain information the shared nodes of Figure 5-6. The projection intersection points and the nodal displacements have different parental elements for both nodes.	82

Acknowledgements

This Master's thesis is submitted in fulfillment of the requirements for the Degree of Master of Science in the subject of mechanical engineering at the Faculty of Mechanical, Maritime and Materials Engineering at Delft University of Technology. I worked on this thesis for approximately eleven months and was fortunate enough to spend the first six months in Boulder, Colorado, USA at the University of Colorado. I have met many nice people over these past eleven months, who I would like to thank for their support and contribution to the results and my personal development.

I wish to express my sincere appreciation to Prof.dr.ir. Fred van Keulen, professor and chair of the research group Structural Optimization and Computational Mechanics at the Department of Precision and Microsystems Engineering (PME), at the Faculty of Mechanical, Maritime and Materials Engineering at Delft University of Technology, for making this thesis work possible by introducing me to Prof. Kurt Maute. Furthermore, his constructive criticism, during the period that any satisfactory results of this work seemed impossible, challenged me to completely work out the details and figure out why the model did not work as supposed to.

I am very grateful to Prof. Kurt Maute, professor and Associate Dean for Research at the Center for Aerospace Structure (CAS) at the Aerospace Engineering Sciences Department at University of Colorado, for providing this very exciting and challenging thesis assignment, for having me over in Boulder for 6 months and for his help on this difficult research topic. I have learned a great deal on numerical modeling with XFEM and LSM, which would not have been possible without his support.

I would also like greatly acknowledge the support from Dr.ir. Matthijs Langelaar, assistant professor at the research group Structural Optimization and Computational Mechanics at the Department of Precision and Microsystems Engineering (PME), at the Faculty of Mechanical, Maritime and Materials Engineering at Delft University of Technology. His input during my work, especially in the beginning and end phase, was very fruitful and clarifying. Even though his agenda was tight, his reviews of my report were always extremely helpful.

My thesis committee will consist of Prof.dr.ir. Fred van Keulen (chair), Dr.ir. Matthijs Langelaar, Dr.ir. Wim-Paul Breugem and Dr.ir. Alexander van Zuijlen. Hereby, I would like to thank them for their time and effort to read through my report and for their participation in my defense.

Furthermore, I would like to thank all my colleagues at both research groups. In particular, Nick Jenkins and Evert Hooijkamp were always available for the short chat, whenever I got stuck and needed someone to share my thoughts with. They both provided me with very helpful reviews of my report. All the other colleagues have also been of great support and I am positive that without their help, this report would not have been possible.

Finally, I would like to thank family, friends and all other people that have supported me throughout my studies. I am very grateful for all opportunities given to me that allowed me to challenge and develop myself. It has been a great journey from high school to graduation and I am excited to discover more of what the future has to offer.

Delft, University of Technology
November 28, 2013

Thijs Bosma

Chapter 1

Introduction

Background Fluid structure interaction is the field of research working on the interaction of a movable or deformable structures with internal or external fluid flows. It is a very popular field of research and has a wide variety of practical implications. At one end of the FSI-spectrum lies *vibro-acoustics*, which looks at the interaction between a structure and linearized fluid flow (i.e. sound-propagation). At the other side of the spectrum one looks at large displacements of structures and non-linear fluid behavior. The latter problems, in particular, gained a lot of interest with the development of computational power. The whole spectrum of FSI is part of our every day live: airplane wings in air, blood flow in the human heart, vocal cords producing sounds and cable stayed bridges in the wind. Advanced FSI modeling is important to be able to accurately predict performance of systems that have become more and more complex over the years (aircrafts, bridges, micro-fluidic systems). In the old days, FSI problems have been analyzed using analytic or semi-analytic methods, but these methods are cumbersome, complex to use and based on strong assumptions, such as inviscid flow, harmonic motion and so on. This strongly limits the applicability of these methods to everyday problems. Today, methods rely more on approximation methods, such as Finite Element or Finite Volume Methods. Prediction and also optimization of FSI behavior, with the mentioned approximation methods, is a very hot research topic. This research is focused on FSI modeling with eXtended Finite Element Method (XFEM), which is a fairly new approach [Gerstenberger and Wall, 2008a, 2010]. A novel approach based on the XFEM and levelset method (LSM), using superposed meshes is implemented and is investigated to determine its characteristics. This particular approach has very promising characteristics for optimization procedures in terms of crisp interface definition. Optimization of FSI problems has mostly been done using density- or porosity-based methods that struggle with accurate physical behavior at the fluid-structure interface. Many different methods, such as the Arbitrary-Lagrangian-Eulerian method, have been developed in the past to successfully model FSI problems, but an XFEM approach has the potential to be more computationally efficient during optimization.

Goal The goal of this thesis to set up a monolithic solver to find the steady state solution of a FSI problem. A monolithic solver requires a numerically consistent Jacobian. By numerical

testing and characterization of the model, information is gathered to provide insights into how this model can be improved and used within an optimization framework in future research. Both the used methods and the implementation of these methods in MATLAB pose many difficult challenges and, hence, careful documentation on these challenges is very valuable for future work.

Methodology The basis for this research is provided by the Newton-Raphson method for solving non-linear systems of equations. Using the non-linear equations and the first derivative of these equations with respect to the solution, the solution is found in an iterative manner. We need to make sure all ingredients of the iterative process are correct, so different numerical setups are implemented to test the ingredients separately. The results of these different setups are presented and analyzed to be able to provide guidelines for improvements in future research.

Nomenclature In this report, the following notations are used: lower case bold symbols \mathbf{a} denote vectors and upper case \mathbf{A} denote matrices, unless explicitly defined otherwise. Non-bold lowercase symbols a indicate scalars. On occasion, the Einstein convention is used a_{ij} , where $i = 1..3$ and $j = 1..3$ and repeated indices mean summation. Both super- and subscript f and s indicate fluid and structure, respectively. Superscript and subscript n indicates a nodal value. For derivatives both d and ∂ are used to denote total and partial derivative, respectively. Sometimes the word phase is used to refer to either fluid or structure. Finally, the levelset based XFEM model for FSI problems implemented in MATLAB, that is the focus of this research, will be referred to as the ‘XFEM model’.

Report outline The report is structured as follows: Chapter 2 presents some general concepts to help the unfamiliar reader with understanding the complete XFEM model. The complete model is presented in Chapter 3. In Chapter 4, the results and analysis of simulation with staggered solver are presented to show that the FSI problem can be solved. Chapter 5 presents results and analysis on a monolithic approach for solving the system, which is needed in order to continue with future research on Topology Optimization using this XFEM model. Chapter 6 presents a short discussion, but most importantly illustrates the potential of the XFEM model for future research. The report ends with conclusions and recommendations in Chapter 7. Chapter 2 to Chapter 5 all end with a short summary to give an overview of what was presented in that chapter.

Chapter 2

General Concepts

This chapter presents some general concepts related to *levelset based modeling of fluid-structure interaction problems with the eXtended Finite Element Method*. These concepts are the basic ingredients of the complete XFEM model that will be presented Chapter 3. It is included help readers, who are unfamiliar with the concepts. Section 2-1 will present eXtended Finite Element Method (XFEM), as a numerical modeling technique. Section 2-2 explains how levelset method (LSM) is used to define geometries. The Newton-Rapshon method is presented in Section 2-3 as an iterative solver to find the solution of a non-linear problem. Section 2-4 presents some mathematical details on differentiation methods used in numerical modeling and optimization. The chapter is summarized in Section 2-5.

2-1 eXtended Finite Element Method (XFEM)

XFEM¹ is a versatile numerical tool for the analysis of problems characterized by discontinuities, localized deformations and complex geometries. Originally, it was proposed to model crack discontinuities [Belytschko and Black, 1999], but in recent years the method has also been used to model the discontinuities of grains [Moës and Belytschko, 2002], FSI problems [Gerstenberger and Wall, 2008a] and many more. In XFEM, discontinuities do not have to coincide with elements edges. Hence, they may lie within an element as shown in Figure 2-1a. The discontinuity cuts the element in different parts, introducing an interface Γ^+ within the element. This interface corresponds to the geometry of the Ω^- domain, which may contain a fluid or a structure. The interface Γ^+ is defined independently of the FE mesh, and its location within the FE mesh can also be changed independently of the mesh, i.e. the mesh does not have to be fitted to the edge of domain Ω^- and the topology of the mesh remains fixed. Suppose the interface location is updated (see Figure 2-1b), the mesh does not have to be changed to fit the new interface. In other words, the need for remeshing during a boundary update is omitted, saving computation time and numerical noise. This is advantageous,

¹In literature XFEM, the Partition of Unity (PUM) and Generalized Finite Element Method (GFEM) are often used to model discontinuities. All methods are based on the *Partition of Unity principle* and can be considered equivalent. These methods differ by the used strategy to introduce a discontinuity.

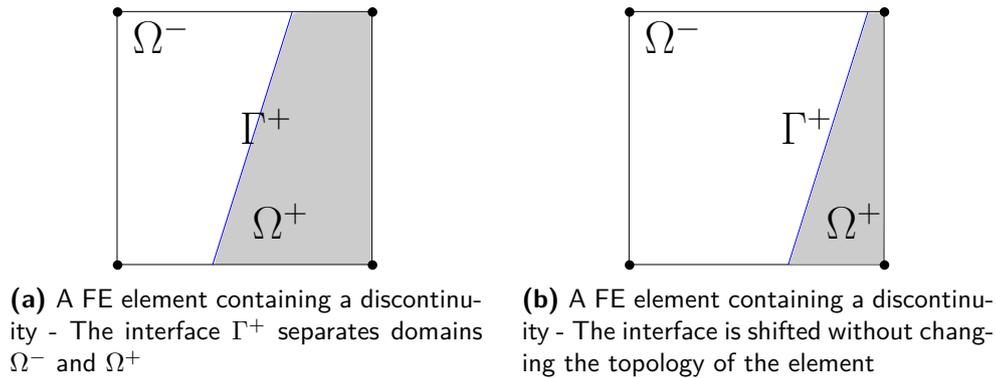


Figure 2-1: Discontinuity within a Finite Element

however, handling the discontinuities does require a more elaborate framework than FEM can provide. For details on boundary remeshing the reader is referred to Wall et al. [2006]. This section will start with a short introduction to the Finite Element Method (FEM) in general. Section 2-1-2 will present how the FEM-framework is extended to handle the discontinuities shown in Figure 2-1.

2-1-1 Introduction to FEM

Many engineering problems are represented by governing partial differential equations. The analytical solution of these governing equations is difficult, or even impossible, to find in most practical situations. A common reason is that the problem has a very complex geometry, see Figure 2-8. FEM was introduced to approximate the solution of these partial differential equations. FEM divides the problem spatially into sub-problems, called Finite Elements (FE), of which we are able to find the solution. These FE can have shapes like triangles, rectangles, tetrahedrons and cubes. The combination of the solution of all the sub-problems should give an approximate solution of the overall problem. FEM helps engineers to analyze and model a wide variety of engineering problems such as stress analysis, heat transfer, fluid flow and electromagnetics by computer simulation. FEM has a strong mathematical framework as will be shown below, but for more details on that the reader is referred to Fish and Belytschko [2007].

General approach

There are several ways to use FEM, but in general the FEM approach consists of the following five steps [Fish and Belytschko, 2007]:

1. Pre-processing: subdividing the domain Ω into finite elements Ω_e , i.e. mesh generation;
2. Processing element formulation: development of equations for elements to describe behavior at specific discrete points, called nodes;
3. Processing assembly: obtaining equations for the whole system based on the equations of the individual elements;

4. Processing solving the system of equations;
5. Post-processing: determining relevant quantities (the response), such as displacements, velocities or pressure and visualizations of the response.

At step 2 the behavior is usually described with partial differential equations, i.e. the strong form. The strong form holds for every mathematical point in the domain and therefore the solution is difficult to find in many practical situations. Approximations of the solution are possible, if the strong form is rewritten into a weak formulation. There are many different methods to approximate the solution of the weak form each having their own characteristics. In this research, a Galerkin approach has been used. The Galerkin approach makes use of the *partition of unity* principle, which is the foundation for the extension to XFEM.

Strong and weak formulation and Weighted Residuals

To illustrate the concept of the strong and weak formulation and the concept of *weighted minimal residual*, we setup a simple, arbitrary problem in strong form:

$$D(\mathbf{u}(\mathbf{x})) = \mathbf{f}(\mathbf{x}) \quad , \quad (2-1)$$

where D is a differential operator acting on a function \mathbf{u} to produce function \mathbf{f} . This is the simplest way to represent partial differential equations, which are used to model many different engineering problems. An example of such a problem is a linear structure subjected to external forces, such that it will deform according to $\mathbf{K}\mathbf{u} = \mathbf{f}$, where \mathbf{K} is the stiffness matrix, \mathbf{u} are the displacements and \mathbf{f} is the external force.

As we assume Eq. (2-1) does not have an analytical solution, we wish to approximate \mathbf{u} by \mathbf{u}^h , according to Eq. (2-2). The solution is then found only for a set discrete points, called nodes, within the domain and this turns weak formulation into a system of equations. The approximation is a linear combination of basis functions chosen from a linearly independent set:

$$\mathbf{u} \approx \mathbf{u}^h = \sum_{i=1}^n \mathbf{u}_i \mathbf{N}_i \quad , \quad (2-2)$$

where \mathbf{u}_i are nodal values and \mathbf{N}_i represents the interpolation between these nodes. Between the nodal points the solution is interpolated with the shape functions N_i . When Eq. (2-2) is plugged into Eq. (2-1) this will generally not give $\mathbf{f}(\mathbf{x})$. The error that is introduced by the approximation is called the *residual*:

$$\mathbf{R}(x) = D(\mathbf{u}^h(x)) - \mathbf{f}(x) \neq 0 \quad . \quad (2-3)$$

The notion of the Method of Weighted Residuals is to force this residual to zero in some average sense over the domain of computation Ω , using the weak formulation:

$$\int_{\Omega} \mathbf{R}(x) \mathbf{W}_i d\Omega = 0 \quad i = 1, 2, 3 \dots n \quad , \quad (2-4)$$

where \mathbf{W}_i are the weight functions and i is the number of unknown variables in \mathbf{u}^h . The result is a set of n algebraic equations. Depending on the choice for the weight function, the approximation methods differ. The Galerkin method uses the derivatives of the approximation

function with respect to the nodal values. If the approximating function is chosen as Eq. (2-2), this gives:

$$\mathbf{W}_i = \frac{\partial \mathbf{u}^h}{\partial \mathbf{u}_i} \quad . \quad (2-5)$$

This means that in the Galerkin method the weight functions are equal to basis functions, $\mathbf{W}_i = \mathbf{N}_i$. The Galerkin method is used in Finite Element Methods to approximate the solution of a partial differential equation. In FEM, the basis functions \mathbf{N}_i are usually called *shape functions* and the unknowns \mathbf{u}_i^h are the nodal values of the state variables. As mentioned the domain is subdivided into elements, which are associated to the nodes. For the linear structural problem introduced earlier, the approximation in Eq. (2-2) will represent the displacement field of the structure. To find this displacement field the Galerkin method is used to force the residual R to zero. An option to do so is to derive Eq. (2-4) with respect to the state variables \mathbf{u}_i , namely:

$$\mathbf{J} = \frac{d \int_{\Omega} \mathbf{R}(x) \mathbf{W}_i d\Omega}{d\mathbf{u}_i} \quad . \quad (2-6)$$

This derivative is called the Jacobian of the system and for linear structural problems is equal to the system's constant stiffness matrix \mathbf{K} . In case of non-linear equations, this derivative is valuable too. Even though it is not constant it can be used within a Newton-Raphson method to iteratively solve the non-linear equations, see Section 2-3 for more details. This is a very brief explanation on FEM, but it does serve as a preliminary to XFEM in Section 2-1-2. For more details on FEM for the Navier-Stokes (NS) equations used in this work, the reader is referred to Section 3-2.

Choice of elements

The domain in FEM is subdivided into elements that all together will give the approximate solution. Many different elements can be chosen, depending on the dimension of the domain. In this research, Q4-elements are used, which are quadrilateral elements with 4 nodes and bilinear interpolation (see Figure 2-2). The shape functions in reference coordinates for this type of element are presented in Eq. (2-7) to Eq. (2-10).

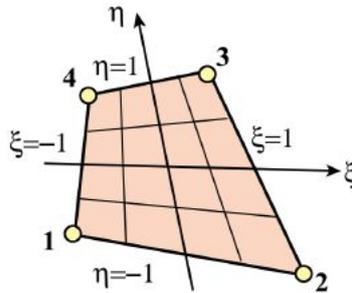


Figure 2-2: The 4-node bi-linear quadrilateral Q4 element [Felippa, 2013] - η and ξ are the natural coordinates and range from -1 to 1

$$N_1 = \frac{1}{4}(1 - \xi)(1 - \eta) \quad (2-7)$$

$$N_2 = \frac{1}{4}(1 + \xi)(1 - \eta) \quad (2-8)$$

$$N_3 = \frac{1}{4}(1 + \xi)(1 + \eta) \quad (2-9)$$

$$N_4 = \frac{1}{4}(1 - \xi)(1 + \eta) \quad (2-10)$$

2-1-2 Extending FEM to XFEM

XFEM is an extension of FEM related to the approximation is defined in Eq. (2-2). In short, XFEM enriches existing nodes with additional Degrees Of Freedom (DOFs) to be able to capture a discontinuity within an element, see Figure 2-3. The nature of this discontinuity may have different physical interpretations and may change location within the element during simulation and/or optimization, as was explained in Figure 2-1. The concept is discussed below using a simple 1-D example.

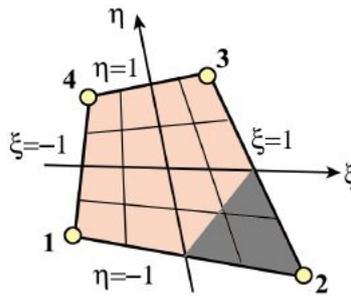


Figure 2-3: The bi-linear quadrilateral Q4 element with a discontinuity inside the element - The discontinuity may, for example, be a transition from solid material (grey area) to fluid material (red area).

Partition of Unity concept in XFEM

The foundation of XFEM is the *partition of unity* concept. A *partition of unity* in general is defined as a collection of functions $N_i(\mathbf{x})$ whose values sum up to unity at each point \mathbf{x} in the solution domain Ω :

$$\sum_{i=1}^n N_i(\mathbf{x}) = 1 \quad \forall \mathbf{x} \in \Omega \quad , \quad (2-11)$$

where n is the number of functions. In the context of FEM, the functions $N_i(\mathbf{x})$ are called shape functions and by definition all Lagrangian FE shape functions satisfy the partition of unity property. Actually, Eq. (2-11) is a mathematical representation of a more mechanical statement, namely that the FE shape functions should be able to represent rigid body translation [Belytschko et al., 2009]. In case of discontinuities, Eq. (2-2) is extended and a vector

field $\mathbf{u}^h(\mathbf{x})$ can be approximated by:

$$\mathbf{u}^h(\mathbf{x}) = \underbrace{\sum_{i=1}^n N_i(\mathbf{x})u_i}_{\text{FE interpolation}} + \underbrace{\sum_{i=1}^n N_i(\mathbf{x}) \sum_{j=1}^m E_j(\mathbf{x})q_{ij}}_{\text{enrichment}} \quad , \quad (2-12)$$

where superscript h denotes the approximate field, N_i the regular FEM shape functions, u_i are the regular nodal DOF values and q_i are the enrichment DOF values. The terms ‘regular’ and ‘enrichment’ make reference to the fact that the ‘regular’ interpolation field is considered as the background field upon which the ‘enrichment’ is added. $E_j(\mathbf{x})$ is called the enrichment function and in XFEM usually is a Heaviside function and m is number of terms in E_j needed to represent the discontinuity. Figure 2-6 illustrates the concept visually for a 1D truss element. Figure 2-4a shows a truss element and the following standard FEM shape functions:

$$\begin{aligned} N_1 &= \frac{x}{L}, \\ N_2 &= 1 - \frac{x}{L}. \end{aligned} \quad (2-13)$$

The shape functions in Figure 2-4a are related to the FE interpolation term of Eq. (2-12). From Eq. (2-13) and Figure 2-4a it is easily verified that these linear FE interpolation functions sum up to unity². Figure 2-5a shows what happens, if these shape functions are multiplied with a Heaviside function, which is defined as follows³:

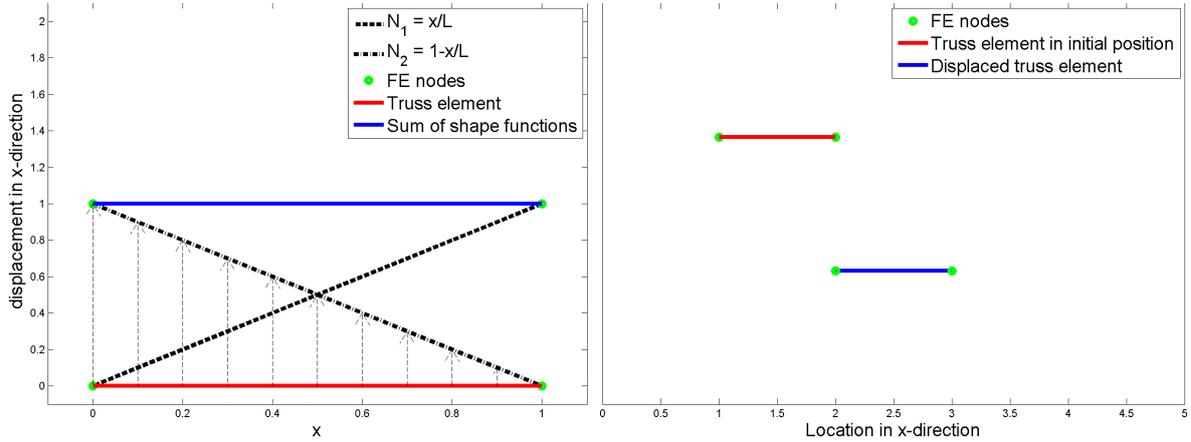
$$H = \begin{cases} 1 & \text{if } x \geq L/2 \\ 0 & \text{if } x < L/2 \end{cases} \quad . \quad (2-14)$$

The effect of the Heaviside enrichment is visualized in Figure 2-4 to Figure 2-6. The following remarks explain the figures.

1. Figure 2-4a shows a truss element with two nodes and the associated shape functions from Eq. (2-13). The blue line shows the sum of the shape functions, corresponding to Eq. (2-11). With unit displacement for \mathbf{u} and no enrichments, Eq. (2-12) will result in the normal displacement shown Figure 2-4b. The truss, in its initial position is depicted as the red truss, will undergo a unit rigid body displacement, which is depicted with the blue truss. Only one dimension is regarded, so the fact that blue truss lies lower than the red truss should be ignored. The latter holds also for Figure 2-5 and Figure 2-6.
2. The enrichment term of Eq. (2-12), using the Heaviside function of Eq. (2-14), is shown in Figure 2-5a. With unit displacement for \mathbf{q} , the physical situation will look as shown in Figure 2-5b. The Heaviside enrichment has introduced a gap with unit length. The truss element is separated into two parts. In this gap no material is present, i.e. void space.
3. If all terms in Eq. (2-12) are summed up the displacement field, with unit nodal values for \mathbf{u} and \mathbf{q} , looks as shown in Figure 2-6a. The physical situation is shown in Figure 2-6b.

²Note: The shape functions for the Q4-element also sum up to 1, see Eq. (2-7) to Eq. (2-10)

³For explanation purposes the domain where the Heaviside functions has a value 1 or 0 is arbitrarily chosen by the author. However, this domain is often defined by a LSM.



(a) Standard FEM element model incl. shape functions associated (b) Standard FEM element model with rigid body displacement

Figure 2-4: Standard FE interpolation term $\sum_{i=1}^n N_i(\mathbf{x})u_i$ - See also remark 1

Again the truss element is separated into two parts by the Heaviside enrichment. Both parts of the blue truss are displaced with unit distance, due to the FE term in Eq. (2-12), but the right part has additional displacement, due to the enrichment term. This illustrates how the standard FEM approach is enriched to incorporate discontinuities.

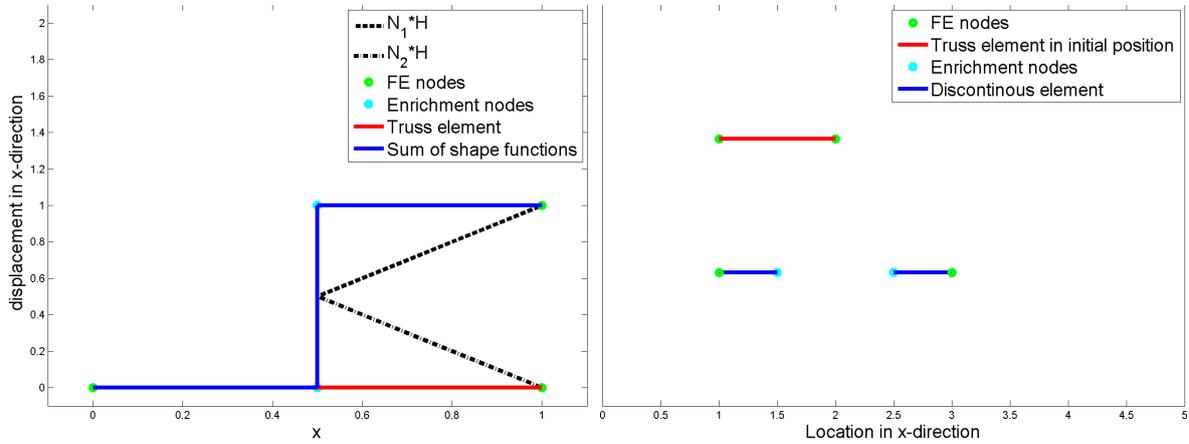
XFEM discretization In order to use XFEM to discretize the equations that describe your system, the approximation of a vector field of first order $l = 1$ over an n -noded enriched element, with all nodes enriched, is rewritten in FE matrix notation:

$$\mathbf{u}^h(\mathbf{x}) = \mathbf{N}(\mathbf{x})\mathbf{u} + \mathbf{N}_E(\mathbf{x})\mathbf{q} \quad , \quad (2-15)$$

where \mathbf{N} is a $l \times (l \times n)$ matrix containing the standard FE shape functions, \mathbf{N}_E is a $(l \times n) \times (l \times m \times n)$ matrix containing the extra basis terms for the enrichment, \mathbf{u} is a $l \times n \times 1$ vector with the standard DOFs and \mathbf{q} is a $(l \times m \times n) \times 1$ vector containing the extra DOFs [Simone, 2012]. When Eq. (2-15) is plugged into the weak formulation of the governing equations describing the system, one ends up with a matrix form suitable for solving with numerical algorithms [Kreissl and Maute, 2012; Gerstenberger and Wall, 2008a,b, 2010].

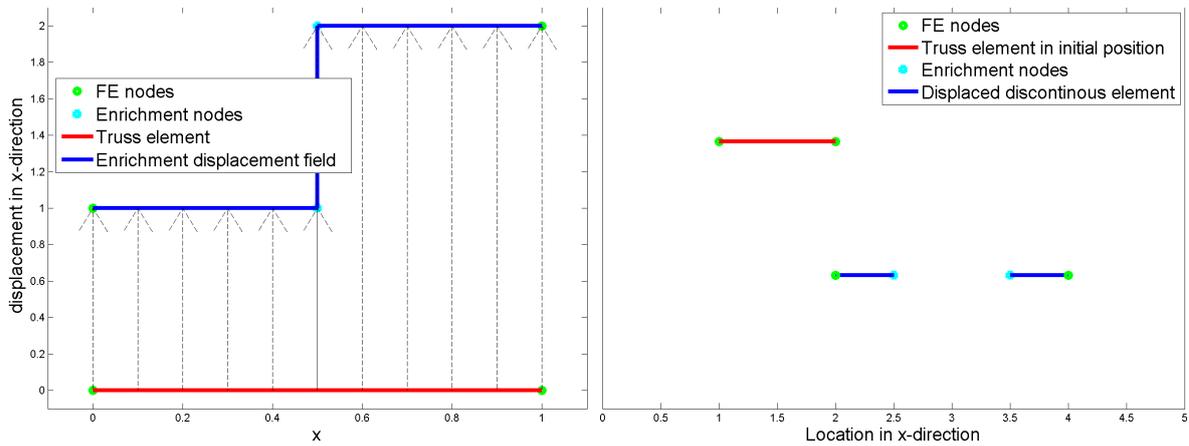
Problems with XFEM Unfortunately some problems need to be addressed when using XFEM. Below, these problems are listed without any details. For the details on these problems the reader is referred to the overview written by Fries and Belytschko [2010].

- The Kronecker-delta property is lost: in the FEM-approximation shape functions corresponding to a node should have value 1 at that node and value 0 at all other nodes. In XFEM, this no longer holds, due to the Heaviside enrichment in intersected elements, see the left node in Figure 2-5a. In this research, this problem is circumvented as the enrichment is used to turn parts of the elemental domain off, i.e. either Ω^- or Ω^+ is *deleted* from the solution and considered as void. Hence there is no need to deal with this as in crack modeling [Fries and Belytschko, 2010].



(a) Discontinuous shape functions of the enrichment term (b) Discontinuous element with unit Heaviside displacement

Figure 2-5: The enrichment term $\sum_{i=1}^n N_i(x)E(x)q_i$ with the Heaviside enrichment H - See also remark 2



(a) Discontinuous displacement field (b) Displaced discontinuous element

Figure 2-6: Illustration of Partition of Unity concept - The figures on the left show a truss element and associated discontinuous shape functions. The figures on the right show the physical situation based on unit Heaviside displacement and unit displacement fields - See also remark 3.

- Elements are cut by the discontinuity but still need to be integrated. The discontinuity changes the shape of the domain, see Figure 2-3. Gaussian quadrature can only be used for integration over standard elemental shapes like triangles and rectangles. The red area is not a standard shape and hence this area needs to be triangulated before integration is possible, see Section 3-2-3.
- The XFEM formulation could lead to ill-conditioned problems (near singular matrices), when the discontinuity leaves a very small part of an element). The use of a preconditioner helps with this problem.

2-2 Levelset method (LSM)

LSMs are numerical techniques to track interfaces and geometries. They were originally proposed by Osher and Sethian [1988] to model moving interfaces in multi-phase flows in crystal growth and flame propagation problems. Unlike density-based methods, LSMs use an implicit description of interfaces to parametrize the geometry. This implicit description allows easy treatment of topological changes of the geometry [Van Dijk et al., 2013]. This section will shortly introduce the concept of LSMs. The actual LSM, used in this research, is presented in Section 3-1.

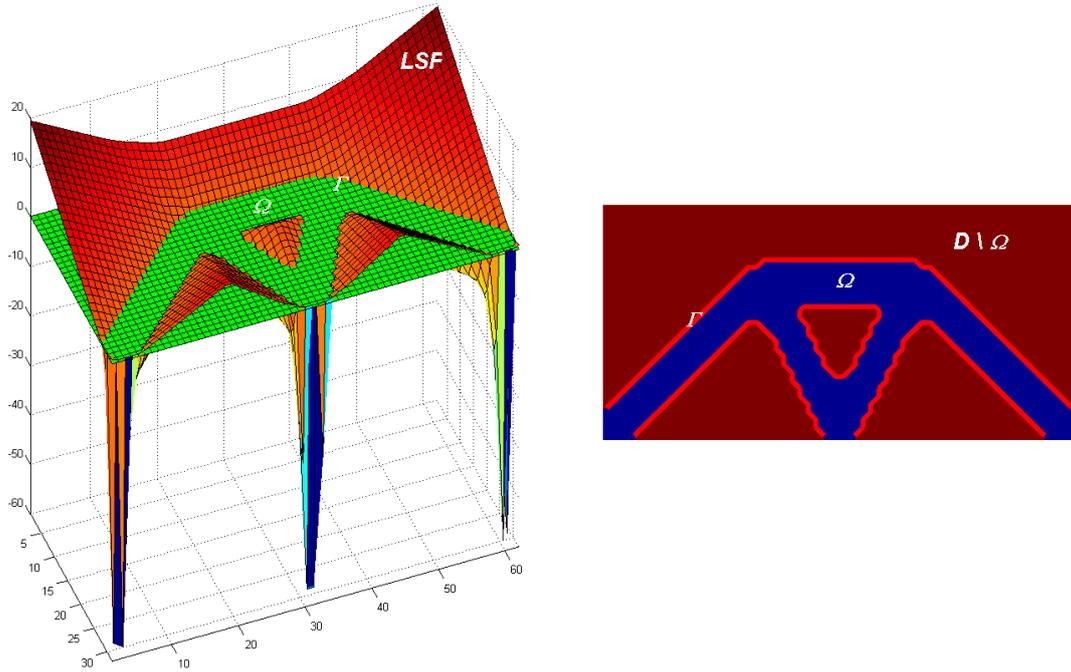
2-2-1 The concept of levelset methods

A crisp description of the interface between two material phases can be defined, by means of the iso-contour lines of a levelset function. Commonly the LSMs divide the computational domain in three parts: the material domain Ω , the void domain ($D \setminus \Omega$) and the material interface Γ as:

$$\begin{aligned}\phi(\mathbf{x}) < c &\Leftrightarrow \mathbf{x} \in \Omega \text{ (material)} \quad , \\ \phi(\mathbf{x}) = c &\Leftrightarrow \mathbf{x} \in \Gamma \text{ (interface)} \quad , \\ \phi(\mathbf{x}) > c &\Leftrightarrow \mathbf{x} \in D \setminus \Omega \text{ (void)} \quad ,\end{aligned}\tag{2-16}$$

where ϕ is the levelset field (LSF), c is the constant iso-contour interface value (usually $c = 0$) and \mathbf{x} is a point in the design domain D . Void, in this context, means that there is no material present, or it is ignored and not part of the physical solution. Figure 2-7 shows an example of a LSF with a zero contour geometry description of a bridge like structure. By changing the LSF the geometry is changed. When combined with a suitable modeling approach and geometry mapping, LSMs allow for a more accurate and physically correct mechanical model in the vicinity of the interfaces [Van Dijk et al., 2013]. This is a big benefit compared to modeling with *density based methods*, where intermediate densities around the fluid-structure interface give nonphysical behavior. This is illustrated in Figure 2-8. For fluid-structure interaction problems in particular, a crisp description of the interface is needed, as this interface is used to couple the solid domain with the fluid domain. Another important remark to remember is that one of the arguments to use the XFEM, is that this discretization technique also allows for a sharply defined interface, even though the interface does not coincide with element edges.

XFEM and LSM fit together perfectly regarding the definition and handling of crisp interfaces. This is particularly valuable in models where the interface is part of the physical problem, for



(a) A levelset function - The intersection between the levelset function and the green zero iso-plane describes the contour of the geometry

(b) The red contour Γ is the zero contour of the levelset function in Figure 2-7a - The domain is separated in three parts according to Eq. (2-16)

Figure 2-7: Illustration of the levelset method (LSM) to describe geometries

instance, in FSI problems. Moreover, this flexible way of describing the geometry with LSMs has a lot of potential in topology optimization [Kreissl and Maute, 2012].

2-3 Non-linear solver - Newton-Raphson

Engineering problems are often described by a set of non-linear equations. If the weak formulation of the problem is discretized according to Section 2-1, a non-linear solver is applied to iteratively find the solution. In FEM, a widely used non-linear solver is the Newton-Raphson solver. It is used to find zeros of some function. Within the FEM context, this means to find an approximated solution of the problem, that gives a residual close to zero, see Eq. (2-4). The method aims to successively find better solutions for the problem based on first order Taylor approximations. For systems of equation the Newton-Raphson process can be written as follows:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \frac{\mathbf{f}(\mathbf{x}_n)}{\mathbf{F}'(\mathbf{x}_n)} \quad . \quad (2-17)$$

In Eq. (2-17), \mathbf{x}_n denotes the n-th approximation of the solution of function \mathbf{f} . In FEM context, the function \mathbf{f} is the residual and \mathbf{F}' the first order derivative of \mathbf{f} with respect to the solution \mathbf{x}_n , i.e. the Jacobian, see also Section 2-1-1. In essence, the non-linear problem is solved by successively solving a linearization of the non-linear problem. The update process is illustrated for a 1D example in Figure 2-9. This solver process has inspired a wide variety of



Figure 2-8: Optimized geometries for a classic L-bracket problem - Levelset based interface on the left, density based on the right with unphysical intermediate densities [James and Martins, 2012]

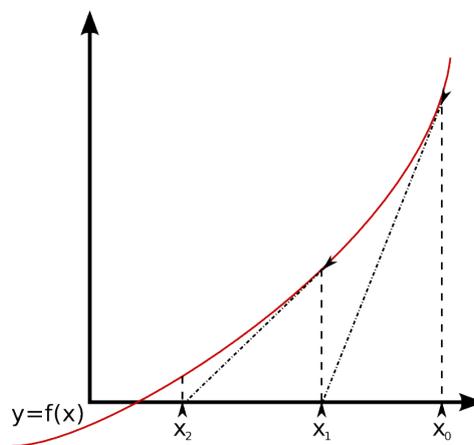


Figure 2-9: A 1D Newton-Raphson update process - http://commons.wikimedia.org/wiki/File:Newton-Raphson_method.png

gradient-based solvers, each being more suitable for particular problems. Without presenting much more details, some characteristics of this method are:

- Quadratic convergence in the neighborhood of the solution;
- Success of the solver strongly depends on the initial guess;
- The possibility exists that the solver overshoots the desired solution;
- The derivative in stationary points is zero thus the process terminates.

There are multiple techniques to handle (some of) the problems mentioned above. These techniques, however, require that the residual and Jacobian are correct. After this has been verified, one can look into these tuning methods.

2-4 Differentiation in numerical modeling

Derivatives, in the most general sense, are measures of how a function changes as its input changes, i.e. how sensitive the function is to a change in input. This concept is of crucial

importance in calculus and numerical modeling. If correctly used, the derivative information leads the way to find a (local) minimum or maximum. Although other methods, such as the bi-section method, can also be used, derivatives usually are efficient and accurate in finding the extremes. As mentioned, one aims to reduce the residual error in FEM modeling to zero, by using the Jacobian to update the solution. In optimization, one aims to minimize some objective function by changing some design variables, using the derivative of the objective function with respect to the design variables. To find the derivatives of a function $f(x)$, one has several options and below Finite Differences and analytical derivatives will be presented.

2-4-1 Finite Differences

The derivative in FD is defined as the ratio between the difference of the perturbed system and the original system, and the step size. If the step size goes to zero in the limit, one ends up with an analytic derivative. Below only first order derivatives are presented for simplicity:

$$f' = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} \approx \frac{f(x + \Delta x) - f(x)}{\Delta x} \rightarrow \text{Forward difference} \quad , \quad (2-18)$$

$$f' \approx \frac{f(x) - f(x - \Delta x)}{\Delta x} \rightarrow \text{Backward difference} \quad , \quad (2-19)$$

$$f' \approx \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} \rightarrow \text{Central difference} \quad , \quad (2-20)$$

where $f(x)$ is the function describing your system, x is the variable and Δx is the step size. The differences between these three methods are illustrated in Figure 2-10. Although finite

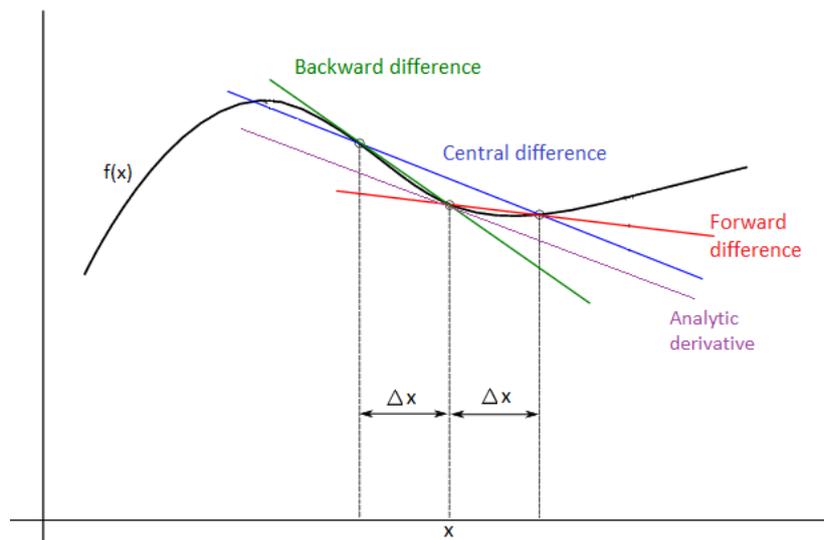


Figure 2-10: Finite Differences - The purple line shows the slope of analytic derivative $df(x)/dx$ at point x , the green line the slope according to backward differences in Eq. (2-19), the blue line according to central differences in Eq. (2-20) and the red line according to forward differences in Eq. (2-18).

difference methods are very easy to apply, one should take care during application, because these methods have shortcomings. Choosing step size is difficult but crucial because large

step sizes can suffer from truncation errors and small sizes can lead to condition errors (for instance, as a result of round-off errors). Computational efficiency, accuracy and consistency depend heavily on the type of solvers used to solve the discretized system, but in general finite differences are costly because for every variable at least one (two for central finite differences) extra evaluation of the system response is needed [Van Keulen et al., 2005]. The error introduced by the approximations in Eq. (2-18) to Eq. (2-20) is an order lower for central differences, which is supported by Figure 2-10. It shows that central differences gives the best approximation of the slope, since the purple and blue line are approximately parallel. Analytical derivatives are preferred over Finite Differences (FD) from an accuracy point of view, but unfortunately sometimes a lack of analytical relation forces one to use FD.

2-4-2 Discrete analytical derivatives - Adjoint formulation

For analytical derivatives it is important to include all dependencies on all (intermediate) variables in the *chain rule*. They are used in both modeling and optimization algorithms. Implementation of the analytical derivatives in FEM is usually a rather difficult, but rewarding process from an accuracy and efficiency point of view. In some optimization procedures, another technique can be used to make the computation of the sensitivity more efficient, i.e. the derivative of the system's response function with respect to design variables⁴. The adjoint method is an efficient method to do sensitivity analysis when the number of response functions is less than the number of design variables [Van Keulen et al., 2005]⁵. In essence, the adjoint method uses a Lagrange multiplier to get rid of the state variable vector sensitivity, which is expensive to compute. Starting with a standard response function z , which is a function of the design variables \mathbf{s} and the state variables $\mathbf{u}(\mathbf{s})$ it can be augmented as follows:

$$z^*(\mathbf{s}, \mathbf{u}(\mathbf{s})) = z(\mathbf{s}, \mathbf{u}(\mathbf{s})) + \boldsymbol{\lambda}^T (\mathbf{f}(\mathbf{s}) - \mathbf{K}(\mathbf{s})\mathbf{u}(\mathbf{s})) \quad . \quad (2-21)$$

The term after the Lagrange Multiplier $\boldsymbol{\lambda}$ may be recognized as the residual formulation for linear elastic structural problem from Eq. (2-3). When this equation is derived with respect to the design variables \mathbf{s} , rearranged and rewritten in index notation, the result will lead to a particular choice for $\boldsymbol{\lambda}$ that avoids computation of the state sensitivity $d\mathbf{u}/ds_j$:

$$\frac{dz_i^*}{ds_j} = \frac{\partial z_i}{\partial s_j} + \boldsymbol{\lambda}_i^T \left(\frac{\partial \mathbf{f}}{\partial s_j} - \frac{\partial \mathbf{K}}{\partial s_j} \mathbf{u} \right) + \left(\frac{\partial z}{\partial \mathbf{u}} - \boldsymbol{\lambda}_i^T \mathbf{K} \right) \frac{d\mathbf{u}}{ds_j} \quad . \quad (2-22)$$

The last term on the right hand side in Eq. (2-22) determines the $\boldsymbol{\lambda}$ that cancels whole term itself, leaving only:

$$\frac{dz_i^*}{ds_j} = \frac{\partial z_i}{\partial s_j} + \underbrace{\mathbf{K}^{-T} \left(\frac{\partial z_i}{\partial \mathbf{u}} \right)^T}_{\boldsymbol{\lambda}_i^T} \left(\frac{\partial \mathbf{f}}{\partial s_j} - \frac{\partial \mathbf{K}}{\partial s_j} \mathbf{u} \right) \quad . \quad (2-23)$$

This last equation shows that the Lagrange Multiplier uses the stiffness matrix of a linear elastic structural problem. If the external force \mathbf{f} is a constant, the stiffness matrix equals the

⁴The topic of Topology Optimization has not been touched in this research, however, it will be part of future research with this XFEM model. This section is included to emphasize the importance of a numerically consistent Jacobian.

⁵If the number of response functions is more than the number of the design variables, the so called *direct method* is used.

system's Jacobian. Hence, one needs the system's Jacobian for the adjoint formulation. The state sensitivity in FEM models is expensive to compute and avoiding this with the adjoint formulation is a smart choice in (X)FEM-based optimization.

2-5 Summary

The general concepts in this chapter provide background to obtain a better understanding of the complete XFEM model, presented in Chapter 3. In this research, an FSI problem is modeled with spatial XFEM discretization in combination with LSM geometry description. XFEM is based on regular FEM, but allows the modeling of discontinuities with relative ease. The location of these discontinuities can be determined by a LSM. With both FEM and XFEM, the goal is to find an approximated solution of the governing partial differential equations. The method of weighted residuals and the Galerkin approach provide the framework to iteratively find the approximated solution of non-linear problems.

The LSM is a flexible way to describe the geometry of a system by taking the zero-contour of the LSF. It provides a crisp interface, which is suitable for problems where the interface is a crucial part of the problem. FSI problems are a good example of problems where physics at the interface determine the behavior of the whole system. Additionally, due to the flexible geometry description, the method is also useful in optimization, where the initial geometry is altered to improve the system's performance.

The solution of the partial differential equations describing the system can be found iteratively with a Newton-Raphson method. This method solves a non-linear problem by solving a sequence of linearized problems using first order derivatives. In FEM context, the solver reduces the residual by updating the solution based on the system's Jacobian.

First order derivatives can be computed both numerically and analytically. A numerical method is the FD method. In this method the derivatives are approximated by the difference of the perturbed response and the non-perturbed response divided by the perturbation size. It is a simple concept, but there are some practical complications, such as choosing the correct step size. The analytical derivatives use the chain rule and are more accurate and computationally efficient in most situations. Implementational effort is higher, but it is usually worth the effort. In optimization problems involving many design variables and few responses, the adjoint formulation is the best approach to compute the system's sensitivity with respect to the design variables efficiently. As with the Newton-Raphson method, the adjoint formulation requires a numerically consistent Jacobian of the system. This emphasizes the need to build a correct residual and to find a numerically consistent Jacobian.

The complete XFEM model

In this research, a 2 dimensional fluid-structure interaction (FSI) problem is modeled with LSM and XFEM. This approach is particularly interesting to investigate from the point of view of optimization, because of the flexible geometry description and crisp interfaces. As was already mentioned in Chapter 2, both solving the system and the adjoint formulation for optimization benefit from a numerically consistent Jacobian. This chapter explains how the Jacobian is built. It starts with explaining the geometry description, which is done using a levelset method in Section 3-1. The physics of the system and the residuals are presented in Section 3-2. Both Section 3-1 and Section 3-2 contain information on how to build a numerically consistent Jacobian, which is the topic of Section 3-3. Section 3-4 presents how the Jacobian is used to solve the system with a staggered and monolithic approach. Section 3-5 presents the actual physical problem that is modeled. The chapter ends with a summary.

3-1 Levelset procedure

The geometries of both fluid and structure in our XFEM model are determined by a zero contour of the levelset field (LSF). The zero contour intersects with element edges on both fluid and structural mesh. This separates the mesh into a material and a void domain, as explained in Section 2-2, describing the geometry as such. The lines separating these two domains are the material to void interfaces. Along these interfaces the fluid-structure coupling will take place and this is where the levelset feature of crisp interfaces proves its potential. This section will discuss the use of the two meshes in Section 3-1-1, the LSF initialization in Section 3-1-2, the tracking of the structural interface in Section 3-1-3, the fluid levelset update procedure in Section 3-1-4 and it will conclude with some reflection on the current implementation of the method.

3-1-1 Separate meshes - 3-field setup

The idea of this XFEM-based FSI model is to use two separate, superposed meshes for the two phases (fluid and structure) in the computational domain, with a third field to transfer

information between the two meshes. Figure 3-1 illustrates the 3-field setup. Superposed, in this context, means that the structural mesh is projected onto the fluid mesh. Using the

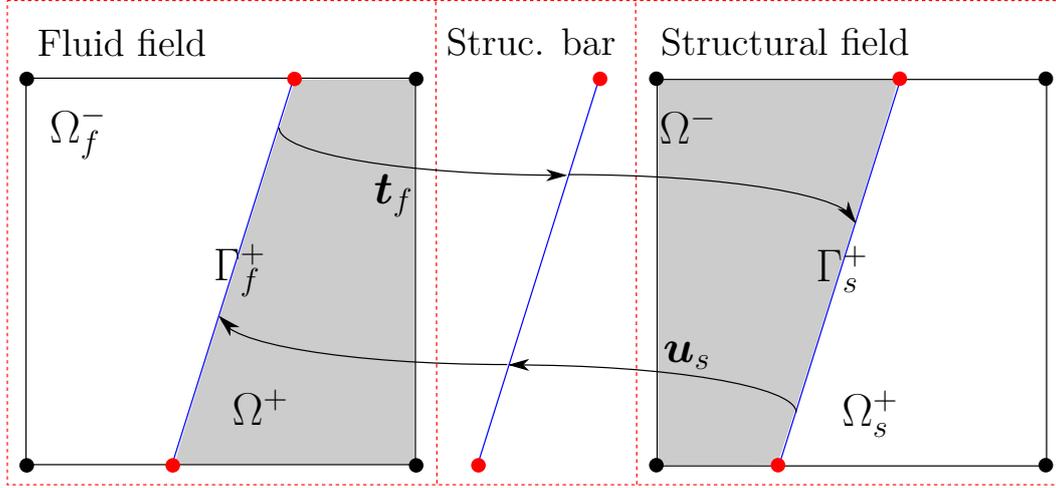


Figure 3-1: 3-field setup - The fluid domain is on the left, the structural domain on the right. The arrows indicate the transfer of FSI coupling information from one domain to the other through the structural bars; \mathbf{t}_f and \mathbf{u}_s traction and structural displacements and velocities, respectively. The grey areas represent voids.

eXtended Finite Element Method (XFEM)-feature to model discontinuities and turn off parts of a mesh, the structure is considered as a void in the fluid mesh and vice versa, see grey areas in Figure 3-1. The geometry of these voids is determined by a zero levelset contour and the interfaces between phase and void are the interfaces where the fluid-structure coupling takes place, according to the FSI conditions presented in Section 3-2. The fluid problem is solved on a fixed Eulerian mesh and the structural problem is solved on a deformable Lagrangian mesh. This is a common approach in FSI, since the governing partial differential equations for fluids and structures exhibit different mathematical characteristics. In general, structural constitutive behavior is given in terms of material coordinates, which corresponds to the Lagrangian representation. In fluid modeling, the relevant physical quantities are generally not known in terms of material coordinates and hence a choice is made for the Eulerian representation.

Figure 3-2 to Figure 3-4 illustrate the approach. Figure 3-2 shows the undeformed Eulerian fluid mesh with an arbitrary initial fluid-structure interface. Figure 3-3 shows an arbitrarily deformed fluid-structure interface on an accordingly deformed mesh. The fluid-structure interface, as shown in Figure 3-2 and Figure 3-3, is defined by the zero levelset contour. From the XFEM perspective: the interfaces indicate the edge of the void in the mesh. The same sign convention is used for both the fluid and structural level set field, namely positive nodes ($\phi_i > 0$) belong to the solid phase (void in the fluid mesh), negative nodes ($\phi_i < 0$) belong to the fluid phase (void in the structural mesh). This sign convention is important to remember, when doing Finite Element (FE) interpolation. Since two meshes are used, two levelset fields are defined. They are initialized in the undeformed configuration, where fluid and structural mesh coincide, see Section 3-1-2. However, as soon as the structure starts to deform and the structural solution is projected onto the fluid mesh, the projection changes and the result is as shown in Figure 3-4. The structural bars mentioned earlier are the key component in

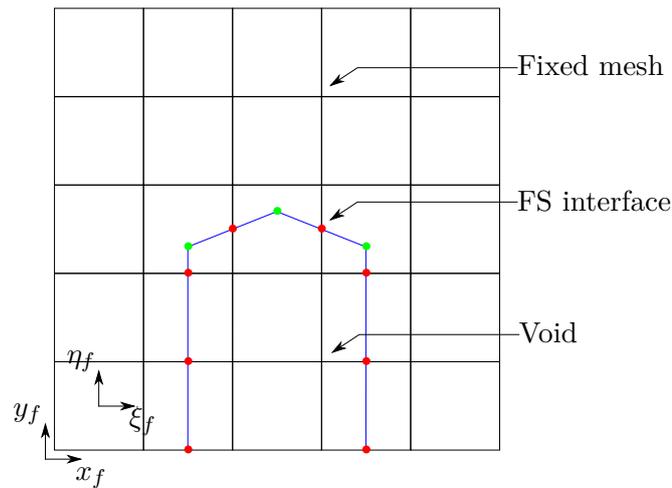


Figure 3-2: Fixed fluid mesh with zero levelset contour in blue lines to indicate fluid to void edge as fluid-structure interface - Inside the contour lies the void domain.

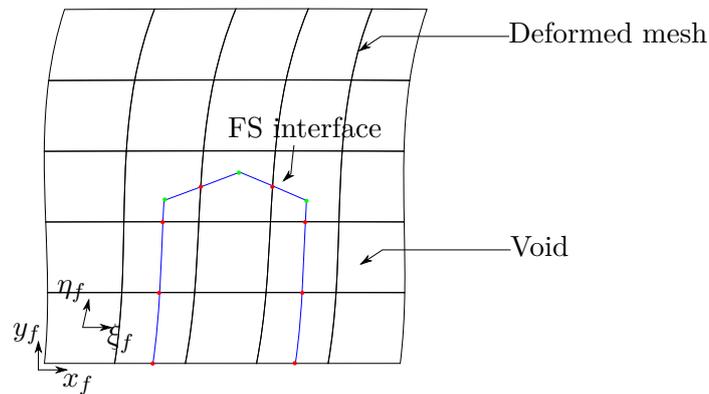


Figure 3-3: Deformed structural mesh with zero levelset contour in blue lines to indicate structure to void edge as fluid-structure interface - Outside the contour lies the void domain

this structural projection onto the fluid mesh. The red dots indicate an intersection between deformed structure and fixed fluid mesh, and these intersections are computed using the structural bars. The intersections between the deformed structure and the fixed fluid mesh provide the information the XFEM solver needs to determine which nodes to enrich and what part of the domain to consider as void or material.

3-1-2 Levelset initialization

The LSF is determined by a *signed distance function*. In essence, this function computes the Euclidian distance from a node to an interface, which gives the absolute nodal levelset value. Then according to the sign convention mentioned earlier, the actual levelset value is given a positive sign in the structural domain and a negative sign in the fluid domain. In order to initialize the LSF, we define the zero levelset contour as a function the dimensions of the structure shown in Figure 3-5. A loop over all nodes in both meshes determines whether the y -coordinate of a node is smaller than the height of the beam without semicircular top

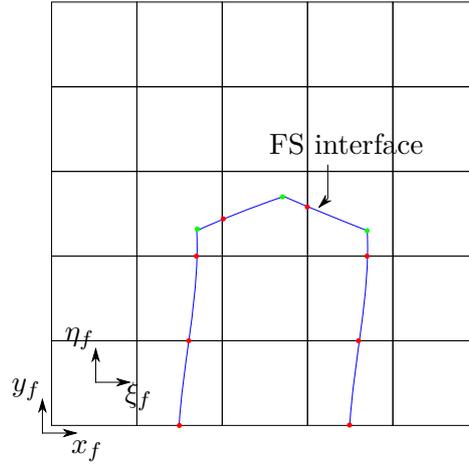


Figure 3-4: Deformed structure projected onto fixed fluid mesh, such that all void areas are filled with the other phase and physical FSI domain is complete.

$h - r$. If this is the case, the nodes left of the structure are given the Euclidean/horizontal distance $\|d_1\|$ as difference between the nodal x-coordinate and the x-coordinate of the left vertical part of the interface. For nodes right of the structure the difference between the nodal x-coordinate and the x-coordinate of the right vertical part of the interface is used. For nodes between the two vertical parts of interface the minimum of the horizontal distances to the two interfaces is used. If the nodal y-coordinate is bigger than $h - r$, the Euclidean distance $\|d_2\|$ is defined as the distance from the node to the center point cp of the semi-circle minus the radius r . The sign is determined by whether a node lies in- or outside the levelset contour, see the + and - in Figure 3-5. This procedure is straightforward, but it does have some implications. The fluid levelset update procedure gives a slightly different result as will be explained in Section 3-1-4. Another important remark to add is that the blue bars connecting the intersection points, depicted by the red dots in Figure 3-5, are *structural bars* and define the third field. Hence, the structural bars are determined by initialization of the LSF and the undeformed meshes. The bars are used to keep track of the interface of the structure in the Eulerian reference frame. The initialization is equal for both the fluid and the structural mesh as the structural mesh is not deformed yet, implying that both meshes will contain intersected elements. Because the structure is modeled on a Lagrangian mesh, the structural LSF is fixed, in contrast to the fluid LSF.

3-1-3 Tracking structural interface deformation

The structural displacements at the interface are tracked using the third field, called the *structural bars*. The structural bars are defined between the initial zero-contour and the Eulerian mesh as shown in Figure 3-5. This third field does not have any physical contribution, i.e. it is solely defined to transfer information from one mesh to the other, as is illustrated in Figure 3-1. These bars are displaced using the nodal displacement field of the previous Newton-Raphson iteration¹ at the zero contour of the structural mesh. This setup allows

¹More details on the solver are presented Section 3-4

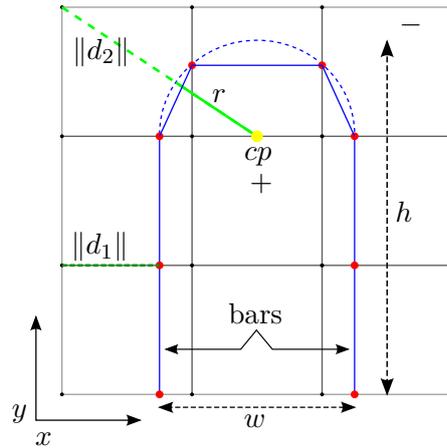


Figure 3-5: Initialization of the levelset field - Dashed blue line is the actual geometry, the continuous blue line is the discretized geometry, which also determines the structural bars between the intersection points (the red dots). Parameters h , w , $r = w/2$ and cp indicate the height, the width, the radius and center point of the top of the structural beam, respectively. See also Figure 3-23 for the complete problem setup.

to track the Lagrangian interface in an Eulerian reference frame and is similar to the 3-field setup of Gerstenberger and Wall [2008a].

A standard intersection configuration

As mentioned the initialization of the LSF fixes the interface for the structure and determines the structural bars. We consider a standard intersection configuration, see Figure 3-6, to illustrate how the Lagrangian structural interface is tracked. The coordinates of the intersec-

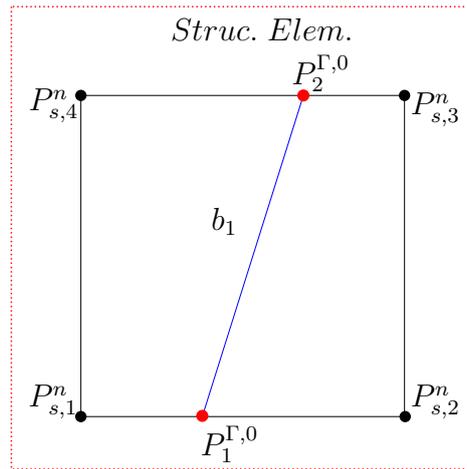


Figure 3-6: An intersected *structural* element considered in undeformed and initial configuration - The red dots are the intersection points between the zero contour and the structural element edges. These red dots will also be defined as the bar ending points.

tion points $P_1^{\Gamma,0}$ and $P_2^{\Gamma,0}$ are determined by intersections between the zero contour of the

structural LSF ϕ_s^n and the element edges, and can be written as:

$$\mathbf{P}_1^{\Gamma,0} = \begin{bmatrix} x_1^\Gamma \\ y_1^\Gamma \end{bmatrix} \quad \mathbf{P}_2^{\Gamma,0} = \begin{bmatrix} x_2^\Gamma \\ y_2^\Gamma \end{bmatrix} . \quad (3-1)$$

As we are considering the undeformed configuration, the points $\mathbf{P}_1^{\Gamma,0}$ and $\mathbf{P}_2^{\Gamma,0}$, actually, also determine the *bar ending points*, which will be called \mathbf{P}_1^b and \mathbf{P}_2^b . The line connecting the bar ending points is the structural bar. In the undeformed configuration the fluid and structural mesh coincide, but as soon as the structure starts to deform this is no longer true. To keep track of the bar ending points in the Eulerian reference frame, the displacements at the bar ending points are interpolated using the nodal solutions for the displacements \mathbf{u}_x^n and \mathbf{u}_y^n as well as the shape functions \mathbf{N}^+ , evaluated at the bar ending points \mathbf{P}_1^b and \mathbf{P}_2^b , which gives:

$$\begin{aligned} u_{x1}^b &= \mathbf{N}_{\mathbf{P}_1^b}^+ \cdot \mathbf{u}_x^n , \\ u_{y1}^b &= \mathbf{N}_{\mathbf{P}_1^b}^+ \cdot \mathbf{u}_y^n , \\ u_{x2}^b &= \mathbf{N}_{\mathbf{P}_2^b}^+ \cdot \mathbf{u}_x^n , \\ u_{y2}^b &= \mathbf{N}_{\mathbf{P}_2^b}^+ \cdot \mathbf{u}_y^n . \end{aligned} \quad (3-2)$$

Using these displacements the coordinates of the bar ending points is updated from their initial position in Eq. (3-1) as follows;

$$\mathbf{P}_1^b = \begin{bmatrix} x_1^\Gamma + u_{x1}^b \\ y_1^\Gamma + u_{y1}^b \end{bmatrix} \quad \mathbf{P}_2^b = \begin{bmatrix} x_2^\Gamma + u_{x2}^b \\ y_2^\Gamma + u_{y2}^b \end{bmatrix} . \quad (3-3)$$

Eq. (3-3) represents the global coordinates of the bar ending points in the *Eulerian reference frame*. In Figure 3-7, the updated configuration from Eq. (3-3) is illustrated. Figure 3-7 shows

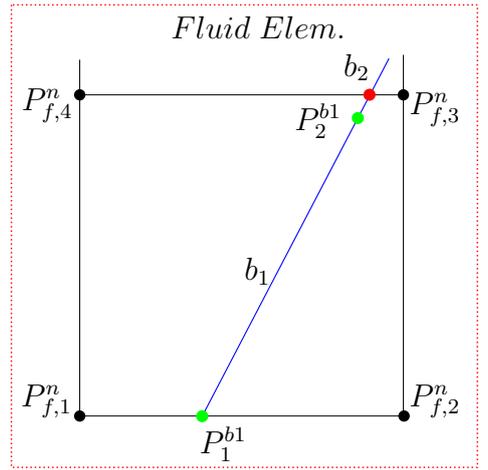


Figure 3-7: The updated location of the structural bar within the *fluid* element that coincided with the undeformed structural element from Figure 3-6 - The bar ending points have moved (green dots) and hence also the intersection points (red dots).

the displaced bar b_1 , the bar endpoints P_1^{b1} and P_2^{b1} , and the fluid element containing the bar.

One can see that in this deformed state the endpoint \mathbf{P}_2^{b1} no longer lies on the fluid element edge and the distinction between $\mathbf{P}_2^{\Gamma,0}$ and \mathbf{P}_2^{b1} becomes clear. The bar b_2 was originally located in the element sharing the top element edge with the element shown in Figure 3-7. Because of the new displacement field, this bar has also moved and now intersects with the element edge to give a changed intersection point (the red dot).

3-1-4 Fluid levelset update

With the updated location of the bars, representing the actual location of the structure to void interface in the Eulerian reference frame, the intersection points between the bars and the Eulerian mesh have changed. These updated intersection points are the ingredients for the fluid levelset update.

Structural projection and intersections

In the previous section, it was explained how the structural interface was tracked within the Eulerian mesh, using the structural bars. The intersections between the structural bars and the fluid element edges, called *projected intersection points*, will provide the information for the fluid levelset update procedure to update the LSF and hence the void area in the fluid mesh. The following steps are part of this projection:

1. Compute the projected intersection points between the bars and the fluid element edges in global and local coordinates;
2. Divide the bars into *sub-bars* at the intersections points and compute location of the sub-bar endpoints in global and local coordinates;
3. Compute the outward normal to the sub-bars in global coordinates, pointing from structure to fluid;
4. Compute Gauss points and weights on the sub-bars in global and local coordinates.

The last three steps are done for Gaussian integration purposes of the traction, discussed in Section 3-2-3. The process is illustrated by Figure 3-8.

The fluid LS update procedure

With the projection of the structure onto the fluid mesh, it is possible to calculate the new fluid levelset field. For each node, connected to an intersected element, the perpendicular distance to the closest interface is calculated, with the interface defined as the straight line between two intersection points (see red dashed line Figure 3-8). The intersection points are found by straight forward vector computation, as will be shown in this section. Element edges are defined between the nodal coordinates defined below:

$$\mathbf{P}_1^n = \begin{bmatrix} x_1^n \\ y_1^n \end{bmatrix}, \quad \mathbf{P}_2^n = \begin{bmatrix} x_2^n \\ y_2^n \end{bmatrix}, \quad \mathbf{P}_3^n = \begin{bmatrix} x_3^n \\ y_3^n \end{bmatrix}, \quad \mathbf{P}_4^n = \begin{bmatrix} x_4^n \\ y_4^n \end{bmatrix} \quad . \quad (3-4)$$

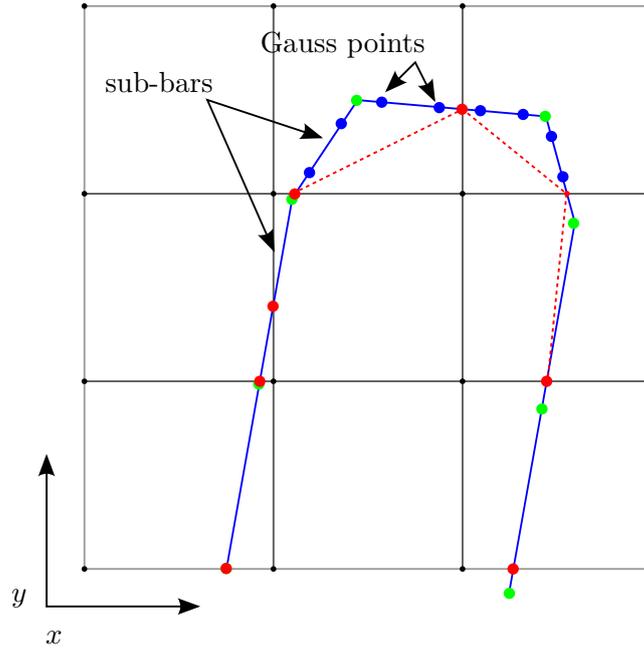


Figure 3-8: Projected deformed structure on fluid mesh with bar endpoints as green dots, intersections as red dots, Gauss points as blue dots and the red dashed line as the interface between the projected intersection points

The bars are defined between the bar endpoints in Eq. (3-3) and may be located with respect to an element edge as shown in Figure 3-9. To solve for the local coordinates r and s , a system of equations can be set up that represents the location of the intersection point based on both line segments, namely:

$$\mathbf{P}_1^b + (\mathbf{P}_2^b - \mathbf{P}_1^b) \cdot s = \mathbf{P}_1^n + (\mathbf{P}_2^n - \mathbf{P}_1^n) \cdot r \quad , \quad (3-5)$$

which can be rewritten to:

$$\underbrace{\begin{bmatrix} (\mathbf{P}_2^b - \mathbf{P}_1^b) & (\mathbf{P}_1^n - \mathbf{P}_2^n) \end{bmatrix}}_{\mathbf{A}} \cdot \underbrace{\begin{bmatrix} s \\ r \end{bmatrix}}_{\mathbf{x}} = \underbrace{\mathbf{P}_1^n - \mathbf{P}_1^b}_{\mathbf{b}} \quad . \quad (3-6)$$

The previous equation has the form of $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$, which is helpful when determining the derivatives later on in this section. The location of the intersection point \mathbf{P}_1^Γ in global coordinates can then be found after Eq. (3-6) is solved, with the following formula:

$$\mathbf{P}_1^\Gamma = \mathbf{P}_1^n + (\mathbf{P}_2^n - \mathbf{P}_1^n) \cdot r \quad . \quad (3-7)$$

In this research, the mesh size compared to size of the structure is chosen such that only one interface per element is possible. This means each intersected fluid element will have 2 intersection points with the structural bars computed according to Eq. (3-6) and Eq. (3-7). With these two *projected intersection points*, the perpendicular distance from each node i to the interface can be found as follows:

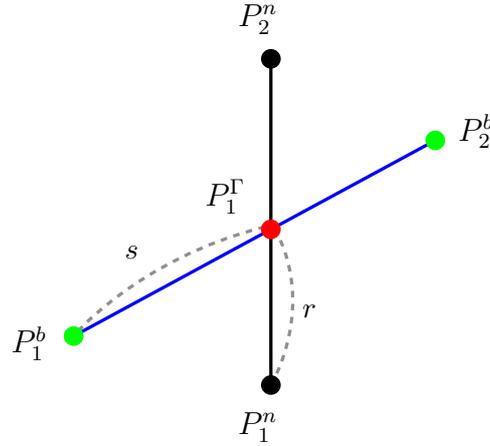


Figure 3-9: Bar intersection with fluid element edge - s and r are local coordinates of the intersection point along the element edge and the bar, respectively. The configuration is chosen arbitrarily, just to illustrate how the structural bars may intersect 1 of the 4 element edges.

1. Define the interface vector \mathbf{m} between in the projected intersection points, i.e. the red dashed line in Figure 3-10:

$$\mathbf{m} = \mathbf{P}_2^\Gamma - \mathbf{P}_1^\Gamma = \begin{bmatrix} x_2^\Gamma \\ y_2^\Gamma \end{bmatrix} - \begin{bmatrix} x_1^\Gamma \\ y_1^\Gamma \end{bmatrix} = \begin{bmatrix} x_m^\Gamma \\ y_m^\Gamma \end{bmatrix} ; \quad (3-8)$$

2. Define the normal \mathbf{n}_m to the interface vector \mathbf{m} :

$$\mathbf{m}^\perp = \mathbf{n}_m = \begin{bmatrix} -y_m^\Gamma \\ x_m^\Gamma \end{bmatrix} ; \quad (3-9)$$

3. Assemble into a matrix $\mathbf{B} = [\mathbf{m} \quad -\mathbf{n}_m]$ to represent a new interface/normal reference frame. Node i has nodal coordinates \mathbf{P}_i^n and local coordinates/ratios w.r.t the new interface/normal frame \mathbf{o}_i :

$$\mathbf{o}_i = \begin{bmatrix} o_1 \\ o_2 \end{bmatrix}_i = \mathbf{B}^{-1} \cdot [\mathbf{P}_i^n - \mathbf{P}_1^\Gamma] ; \quad (3-10)$$

4. The perpendicular vector from \mathbf{P}_i^n to the interface is defined as:

$$\mathbf{d}_i = (\mathbf{P}_1^\Gamma + \mathbf{m} \cdot o_{1i}) - \mathbf{P}_i^n , \quad (3-11)$$

with $\mathbf{P}_1^\Gamma + \mathbf{m} \cdot o_1$ being the global coordinates in the Eulerian mesh of the orthogonally projected point \mathbf{op} on the interface, see Figure 3-10;

5. The perpendicular distance and hence the absolute levelset value is then given as $\|\mathbf{d}_i\|$.

As we know the outward normal \mathbf{n}_m of the structural bars, these will be used to give the correct sign to the new nodal levelset values on an elemental level. For example, the lower element in Figure 3-10 has nodal levelset values as follows:

$$\text{sign}(\phi_f^e) = [\phi_1^{\text{sign}} \quad \phi_2^{\text{sign}} \quad \phi_3^{\text{sign}} \quad \phi_4^{\text{sign}}] = [-1 \quad 1 \quad 1 \quad -1] . \quad (3-12)$$

This procedure is done in a loop over all nodes connected to intersected elements. Each node is connected to multiple elements and all these elements are checked on intersections. In each connected element, a perpendicular distance is computed, if it exists inside the element. Otherwise the distance to the closest intersection point is used. From all calculated distances for that node, the absolute nodal levelset value is set to minimum calculated distance. This is illustrated in Figure 3-10 at node P_4^n , where the blue dashed line is the distance to the closest intersection point in the top element and the green dashed line is the actual absolute levelset value, given by the bottom element. The updated nodal levelset values per element can then be found as:

$$\phi_f^e = \begin{bmatrix} \phi_1^{sign} \cdot \|d_1\| \\ \phi_2^{sign} \cdot \|d_2\| \\ \phi_3^{sign} \cdot \|d_3\| \\ \phi_4^{sign} \cdot \|d_4\| \end{bmatrix} . \quad (3-13)$$

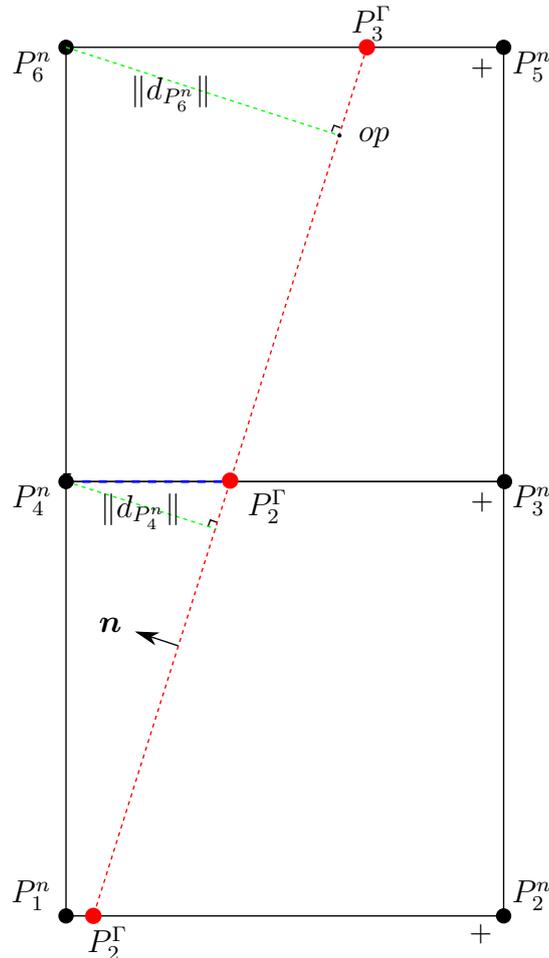


Figure 3-10: Nodal perpendicular distances (green lines) to the interfaces between the projected intersection points.

Visualization of the update process

An overview of the fluid levelset update process is shown in Figure 3-11. On the left, the undeformed configuration is shown, which defines the structural bars between the intersection points of the zero contour and the fixed element edges. The blue bar is actually located between the two red intersection points, which is denoted by the black dashed lines between the green bar ending points and red intersection points. On the right, the location of the structure with respect to fluid mesh has been updated according to the structural displacement field and hence the location of the structural bar with respect to the fluid element has changed. The intersections between bar and element edges determine the nodal levelset field, denoted by the green dashed arrow in the fluid element.

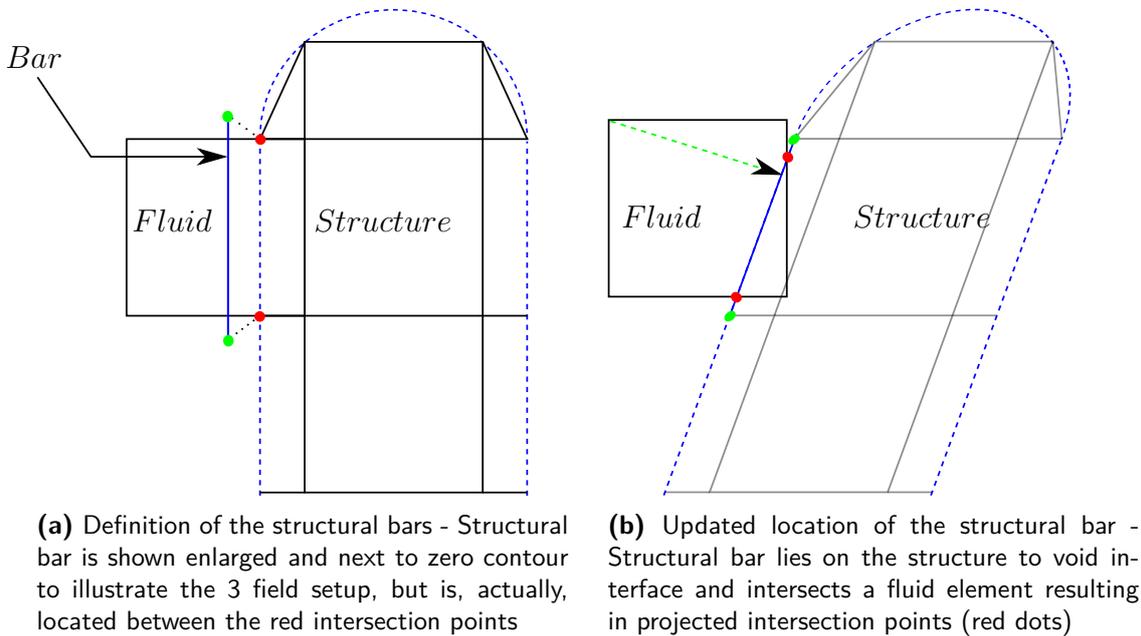


Figure 3-11: Overview update process of the fluid levelset field - The undeformed configuration on the left and a deformed configuration on the right, where the same intersected fluid element is shown but the structural projection has changed

3-1-5 Fluid levelset field derivatives

In Section 3-3, it will be explained that for the Jacobian to be consistent we need, amongst other contributions, the derivative of the fluid LSF with respect to the structural state variables. Section 3-1-4 explains that the fluid levelset field is function of the projected intersection points and that the projected intersection points are a function of the structural displacements. These dependencies are important for computing analytical derivatives as is shown in Eq. (3-14):

$$\frac{d\phi_i^f \left(\mathbf{P}_{1,2}^\Gamma(\mathbf{u}_s^n) \right)}{d\mathbf{u}_s^n} = \frac{\partial \phi_i^f}{\partial \mathbf{P}_{1,2}^\Gamma} \frac{d\mathbf{P}_{1,2}^\Gamma}{d\mathbf{u}_s^n}, \quad (3-14)$$

where \mathbf{u}_s^n are the nodal displacements from the structural mesh in both x- and y-direction used in Eq. (3-2), ϕ_i^f is the nodal levelset value of the i -th fluid node and $\mathbf{P}_{1,2}^\Gamma$ both projected intersection points within the fluid element giving the orthogonal distance for node i . Due to the geometric vector setup of the LSM described in Section 3-1-4, there is an analytical relation for Eq. (3-14).

Derivatives - Nodal levelset value with respect to projected intersection points

The absolute nodal levelset value is given by $\|\mathbf{d}_i\|$ and therefore the derivative of this with respect to projected intersection points looks as follows:

$$\frac{\partial \|\mathbf{d}_i\|}{\partial \mathbf{P}_{1,2}^\Gamma} = \frac{1}{\|\mathbf{d}_i\|} \frac{\partial \mathbf{d}_i}{\partial \mathbf{P}_{1,2}^\Gamma} = \begin{bmatrix} \frac{\partial \mathbf{d}_i}{\partial x_1^\Gamma} & \frac{\partial \mathbf{d}_i}{\partial y_1^\Gamma} \\ \frac{\partial \mathbf{d}_i}{\partial x_2^\Gamma} & \frac{\partial \mathbf{d}_i}{\partial y_2^\Gamma} \end{bmatrix}, \quad (3-15)$$

where \mathbf{d}_i is defined Eq. (3-11). Using the chain rule again gives:

$$\frac{\partial \mathbf{d}_i}{\partial \mathbf{P}_{1,2}^\Gamma} = \frac{\partial \mathbf{P}_1^\Gamma}{\partial \mathbf{P}_{1,2}^\Gamma} + \frac{\partial \mathbf{m}}{\partial \mathbf{P}_{1,2}^\Gamma} o_{1i} + \mathbf{m} \frac{\partial o_{1i}}{\partial \mathbf{P}_{1,2}^\Gamma}, \quad (3-16)$$

where the individual derivative terms look as follows:²

$$\frac{\partial \mathbf{P}_1^\Gamma}{\partial \mathbf{P}_{1,2}^\Gamma} = \begin{bmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \end{bmatrix}, \quad (3-17)$$

$$\frac{\partial \mathbf{m}}{\partial \mathbf{P}_{1,2}^\Gamma} = \begin{bmatrix} \begin{pmatrix} -1 \\ 0 \\ 1 \\ 0 \end{pmatrix} & \begin{pmatrix} 0 \\ -1 \\ 0 \\ 1 \end{pmatrix} \end{bmatrix}, \quad (3-18)$$

$$\frac{\partial o_{1i}}{\partial \mathbf{P}_{1,2}^\Gamma} = \begin{bmatrix} \mathbf{B}^{-1} \begin{pmatrix} o_1 - 1 \\ -o_2 \end{pmatrix} & \mathbf{B}^{-1} \begin{pmatrix} o_2 \\ o_1 - 1 \end{pmatrix} \\ \mathbf{B}^{-1} \begin{pmatrix} -o_1 \\ o_2 \end{pmatrix} & \mathbf{B}^{-1} \begin{pmatrix} -o_2 \\ -o_1 \end{pmatrix} \end{bmatrix}. \quad (3-19)$$

After all these terms are plugged into Eq. (3-15), the final step is to give the derivatives the correct sign according to ϕ_i^{sign} . The levelset derivative with respect to the projected intersection points is written as follows:

$$\frac{\partial \phi_i^f}{\partial \mathbf{P}_{1,2}^\Gamma} = \phi_{f,i}^{sign} \frac{\partial \|\mathbf{d}_i\|}{\partial \mathbf{P}_{1,2}^\Gamma}. \quad (3-20)$$

This method can be extended to multiple intersections quite easily, however, in this research we do not consider cases with multiple intersections, as this poses difficulties for integration of the residual contributions.

²The entries are presented in similar order regarding the denominator, as in Eq. (3-15)

Derivatives - Projected intersections points with respect to structural displacements

For the second term on the right hand side in Eq. (3-14) we need the derivatives of the projected intersection points with respect to nodal displacements. Using Figure 3-9 again, these derivatives can be found as described below. If we differentiate Eq. (3-7) with respect to a dummy variable \tilde{u} , which is later to be substituted by all four coordinates of the two bar ending points, we get the following:

$$\frac{d\mathbf{P}_{1,2}^\Gamma}{d\tilde{u}} = (\mathbf{P}_2^n - \mathbf{P}_1^n) \cdot \frac{dr}{d\tilde{u}} \quad . \quad (3-21)$$

The $\partial r/\partial\tilde{u}$ can be found differentiating Eq. (3-6) with respect to \tilde{u} :

$$\frac{\partial \mathbf{A}}{\partial \tilde{u}} \cdot \mathbf{x} + \mathbf{A} \cdot \frac{\partial \mathbf{x}}{\partial \tilde{u}} - \frac{\partial \mathbf{b}}{\partial \tilde{u}} = 0 \quad , \quad (3-22)$$

which can be rewritten as:

$$\frac{d\mathbf{x}}{d\tilde{u}} = \begin{bmatrix} \frac{ds}{d\tilde{u}} \\ \frac{dr}{d\tilde{u}} \\ \frac{d\tilde{u}}{d\tilde{u}} \end{bmatrix} = \mathbf{A}^{-1} \left[\frac{\partial \mathbf{b}}{\partial \tilde{u}} - \frac{\partial \mathbf{A}}{\partial \tilde{u}} \cdot \begin{bmatrix} s \\ r \end{bmatrix} \right] \quad . \quad (3-23)$$

Evaluating all terms in Eq. (3-23) with the structural displacements at the *bar ending points* gives the following:

- For $\tilde{u} = u_{x1}^b$:

$$\frac{\partial \mathbf{b}}{\partial u_{x1}^b} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, \quad \frac{\partial \mathbf{A}}{\partial u_{x1}^b} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad . \quad (3-24)$$

- For $\tilde{u} = u_{y1}^b$:

$$\frac{\partial \mathbf{b}}{\partial u_{y1}^b} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \quad \frac{\partial \mathbf{A}}{\partial u_{y1}^b} = \begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix} \quad . \quad (3-25)$$

- For $\tilde{u} = u_{x2}^b$:

$$\frac{\partial \mathbf{b}}{\partial u_{x2}^b} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \frac{\partial \mathbf{A}}{\partial u_{x2}^b} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad . \quad (3-26)$$

- For $\tilde{u} = u_{y2}^b$:

$$\frac{\partial \mathbf{b}}{\partial u_{y2}^b} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \frac{\partial \mathbf{A}}{\partial u_{y2}^b} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad . \quad (3-27)$$

This leads to the following solutions for Eq. (3-23):

- For $\tilde{u} = u_{x1}^b$:

$$\begin{bmatrix} \frac{ds}{d\tilde{u}} \\ \frac{du_{x1}^b}{d\tilde{u}} \\ \frac{dr}{d\tilde{u}} \\ \frac{du_{x1}^b}{d\tilde{u}} \end{bmatrix} = \mathbf{A}^{-1} \begin{bmatrix} -1 + s \\ 0 \end{bmatrix} \quad . \quad (3-28)$$

- For $\tilde{u} = u_{y1}^b$:

$$\begin{bmatrix} \frac{ds}{du_{y1}^b} \\ \frac{ds}{dr} \\ \frac{ds}{du_{y1}^b} \end{bmatrix} = \mathbf{A}^{-1} \begin{bmatrix} -s \\ 0 \end{bmatrix} . \quad (3-29)$$

- For $\tilde{u} = u_{x2}^b$:

$$\begin{bmatrix} \frac{ds}{du_{x2}^b} \\ \frac{ds}{dr} \\ \frac{ds}{du_{x2}^b} \end{bmatrix} = \mathbf{A}^{-1} \begin{bmatrix} 0 \\ -1 + r \end{bmatrix} . \quad (3-30)$$

- For $\tilde{u} = u_{y2}^b$:

$$\begin{bmatrix} \frac{ds}{du_{y2}^b} \\ \frac{ds}{dr} \\ \frac{ds}{\partial u_{y2}^b} \end{bmatrix} = \mathbf{A}^{-1} \begin{bmatrix} 0 \\ -r \end{bmatrix} . \quad (3-31)$$

The bottom rows of these last four equations can be plugged into Eq. (3-21). The derivatives of the intersection point coordinates with respect to the nodal structural displacement are then found, according to Eq. (3-2), by post-multiplication with the shape functions \mathbf{N}^+ evaluated at the bar ending points:

$$\begin{aligned} \frac{d\mathbf{P}_{1,2}^\Gamma}{du_{xj}^n} &= \frac{d\mathbf{P}_{1,2}^\Gamma}{du_{xk}^b} \mathbf{N}^+(P_k^b) , \\ \frac{d\mathbf{P}_{1,2}^\Gamma}{du_{yj}^n} &= \frac{d\mathbf{P}_{1,2}^\Gamma}{du_{yk}^b} \mathbf{N}^+(P_k^b) , \\ \frac{d\mathbf{P}_{1,2}^\Gamma}{du_s^n} &= \frac{d\mathbf{P}_{1,2}^\Gamma}{du_{xj}^n} \cup \frac{d\mathbf{P}_{1,2}^\Gamma}{\partial u_{yj}^n} , \end{aligned} \quad (3-32)$$

where $_{1,2}$ denotes the both projected intersection points, j the j -th node of the structural element, k the k -th bar endpoint and n the n -th structural element containing bar b . Section 3-3 will present how all these terms are incorporated to find a consistent Jacobian.

Levelset field consistency

A problem with this levelset method (LSM), that needs to be addressed, occurs in cases of large displacement of the structure, within a Newton-Raphson iteration. In that case, an element initially belonging to the void phase can be found completely immersed in the material phase and vice versa. Before and after the update the element did not contain any intersections and so the procedure described above does not work to update the nodal levelset values. This results in unprocessed elements, which gives local errors in the levelset field. These unprocessed elements will be erroneous void or material elements.

The solution to this problem is a *floodfill algorithm*³. This algorithm essentially sets all nodal fluid levelset values to either 1 or -1 depending on the domain, except for the nodes that are

³The idea of the floodfill is based on floodfill algorithms used in paint-programs http://en.wikipedia.org/wiki/Flood_fill

connected to intersected elements. The latter will be handled by the LSM presented earlier. Every node is flagged to indicate whether it has been processed by the signed distance function setup. A loop over all elements determines the correct phase using material information from the neighboring elements, starting with elements connected to intersected elements as they hold correct phase information. When all the nodes are processed the algorithm is done. For more details and illustrations on this algorithm, the reader is referred to Aloun [2012].

3-1-6 Reflection on proposed levelset method

The proposed LSM has been designed to compute analytical derivatives used in the Jacobian, which is desired from a computational efficiency standpoint, see Section 3-3. Without presenting any specific details, it should be noted that the current MATLAB implementation has some shortcomings and characteristics that one has to be aware of. These shortcomings and characteristics should be taken into account when the results are analyzed later on in Chapter 4. A characteristic of this method is that the fluid levelset is unable to perfectly track the structural projection. The consequences are loss of interface information and intersections that are impossible to handle in the current setup (code break down). These characteristics and shortcomings are presented below, as they have been encountered several times during this research and reduce robustness of the XFEM model.

Characteristic - Discrepancy between projection and fluid levelset

The LSF of the structure is determined by the procedure described in Section 3-1-2 and is fixed, due to the Lagrangian setup. The fluid levelset is updated according to the method described in Section 3-1-4 and, hence, there will always be a discrepancy between the two meshes. This is definitely not uncommon in FSI and it implies the use of approximating projection techniques to transfer information from one mesh to the other, see Section 3-2-3. With more complex structural shapes this discrepancy increases and will reduce accuracy of the results. The discrepancy is illustrated by Figure 3-12⁴.

Shortcomings - Handling different intersection configurations

An essential part of this XFEM model is handling all potential intersection configurations. This is, however, not easy in all cases. Below two problems will be discussed that negatively influence the results, reduce the freedom of choosing mesh coarseness and should be fixed for future research to make the code more robust and generally applicable to other geometries.

Multiple intersections along one element edge In this research, the mesh is chosen such that multiple interfaces within a fluid element do not occur; the geometry of the structure is big compared to the size of the fluid elements. However, the first shortcoming relates to the case where a sharp corner of the structure is projected onto the shared edge between two fluid elements. The configuration is illustrated by Figure 3-13 within the red circle. The figure shows a corner of the projected structure that has just crossed an fluid element

⁴In Section 5-3-3 at Problem 3, the explanation for the discrepancy is presented.

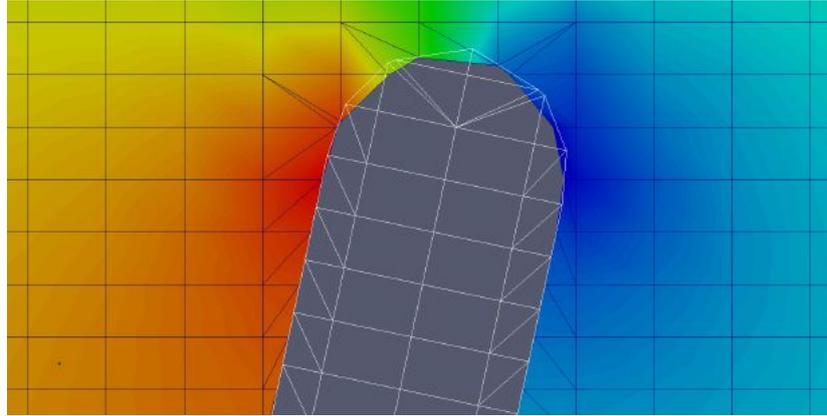


Figure 3-12: Mesh discrepancy - Grey is void in fluid mesh and does not align perfectly with the white wire frame of the structure. The discrepancy is clearly visible, especially at the top of the beam, where the structural details are finer.

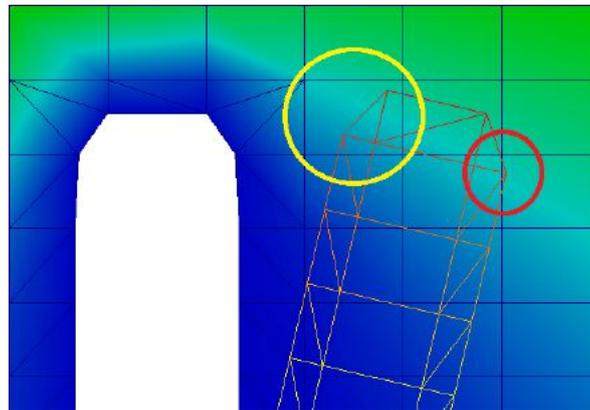


Figure 3-13: Problematic intersections - Double intersections on one element edge in red circle and submersed bar in yellow circle

edge. The element edge is intersected twice, such that the left element has two interfaces inside the element and the right element has two intersections on the same element edge. The intersection configuration of the left element can in theory be handled by XFEM with additional enrichments, but the current MATLAB implementation is not ready for that. The element on the right gives trouble, as intersection points on one edge give an fluid interface aligned with that edge. Local remeshing of the two concerned elements, if the situation occurs, is the most elegant way to handle this configuration, but this solution is too advanced for this research [Gerstenberger and Wall, 2008b]. However, it is recommended for future research. In this research, this configuration is prevented by using a fine mesh. The mesh refinement should aim to reduce sharp corners of the structure on the right. An example of fine mesh is depicted in Figure 3-14. Prevention is not the most robust way to handle this problem, since it uses information on the geometry of this particular problem.

Submersed bars - Traction loss The second situation that gives problems, occurs when a structural bar is completely submersed in a fluid element and does not intersect any fluid

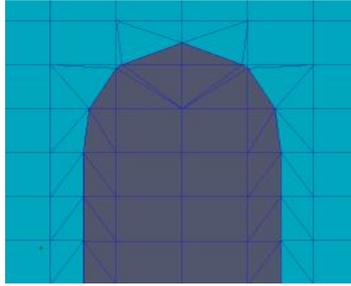


Figure 3-14: Refined mesh to handle multiple intersection case - The structure has a smoother interface, which is less likely to create a problematic intersection configuration.

element edges. The routine that projects the fluid traction onto the structural interface, see Section 3-2-3, only projects traction information onto the bars that actually intersect the elements edges. This is a minor problem, but it does result in loss of physical information. The configuration is also shown in Figure 3-13 within the yellow circle. If one wants to handle this problem, again local mesh refinement would be a good option.

3-2 Fluid Structure Interaction

Fluid-structure interaction is the physical field on which this research is focused. As mentioned in the introduction, the domain in FSI problems contains both a structure and a fluid interacting with each other at a shared interface. The LSM from Section 3-1 determines the geometry of the system and FSI provides the governing partial differential equations that describe the physical behavior of separated parts of the domain and the interaction between the two. This section is built up as follows: Section 3-2-1 and Section 3-2-2 present the equations that describe the physics of the system. Section 3-2-3 explains how the fluid and structure are coupled to complete the FSI model.

3-2-1 Mechanics

To derive the governing equations that describe the physics of a system, one has to choose for a reference frame. In FSI, there are three choices 1) *Eulerian/spatial* 2) *Lagrangian/material* or 3) *Arbitrary Lagrangian Eulerian (ALE)* description. In this work, we have chosen an Eulerian description for the fluid and a Lagrangian description for the structure. Eulerian means that the reference frame of the observer is fixed in space and Lagrangian means that the observer travels with the material. This choice implies the use of two different meshes, which is a key feature in this work mentioned in Section 3-1-1. The ALE description is very popular in FSI and characterized by an observer moving with the fluid-structure interface and a deformable mesh, which is remeshed to fit the updated interface. XFEM has the advantage of not having to re-mesh, saving computational effort and reducing numerical noise. It is possible to use an ALE description within the framework presented in this research, but that would make the XFEM model more complicated. Nevertheless Gerstenberger and Wall [2010] did propose an XFEM-ALE model, which proves to be advantageous in complex fluid-structure interaction cases (especially complex geometries). The choice for one of these reference frames

is important for the mathematics (convective terms), but all three should describe the same physical behavior.

Fluid mechanics

The complete Navier-Stokes equations consist of conservation of mass, conservation of momentum and conservation of energy. These equations are complemented with Navier's law, Fourier's law and the equations of state to fully determine the state of the fluid. The equations of conservation look as follows:

$$\begin{aligned} \frac{\partial \rho_f}{\partial t} + \rho_f \frac{\partial v_i}{\partial x_i} &= 0 \rightarrow \text{conservation of mass} \quad , \\ \frac{\partial \rho_f v_i}{\partial t} + \frac{\partial \rho_f v_i v_j}{\partial x_j} &= \frac{\partial p \delta_{ij}}{\partial x_j} + \frac{\partial \tau_{ij}}{\partial x_j} + \rho_f f_i \rightarrow \text{conservation of momentum} \quad , \\ \frac{\partial \rho_f E}{\partial t} + \frac{\partial \rho_f E v_i}{\partial x_i} + \frac{\partial p v_i}{\partial x_i} - \frac{\partial \tau_{ij} v_i}{\partial x_j} + \frac{\partial q_i}{\partial x_i} &= \rho_f f_i v_i \rightarrow \text{conservation of energy} \quad , \end{aligned} \quad (3-33)$$

where ρ_f is density of the fluid, v_i is fluid velocity, p is fluid pressure, τ_{ij} is the tensor of viscous stresses, f_i are external body forces, E is internal energy and q_i is heat flux due to conduction. Important to note is that in Eq. (3-2-1) there are 5 equations to determine ρ_f , v_i and E , but there are 10 more unknowns: p , 6 shear stresses $\tau_{ij} = \tau_{ji}, i \neq j$ and 3 flux components of q_i . The following equations are introduced to determine the remaining unknowns:

$$\tau_{ij} = \tau_{ji} = \mu \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) - \frac{2}{3} \delta_{ij} \mu \frac{\partial v_k}{\partial x_k} \rightarrow \text{Navier's law} \quad , \quad (3-34)$$

$$q_i = k \frac{\partial T}{\partial x_i} \rightarrow \text{Fourier's law} \quad , \quad (3-35)$$

$$\rho_f = \text{constant}, \quad E = cT \rightarrow \text{Equations of state} \quad . \quad (3-36)$$

Here, the Navier-Stokes (NS) equations are presented in local/differential and dimensional form using the the Einstein convention in a Cartesian and Eulerian reference frame. Depending on assumptions on, for instance, compressibility or low Reynolds numbers the equations above can be simplified. This is the topic of the next section.

Incompressible Navier Stokes equations

The NS equations can be simplified by assuming the following

1. Temperature and energy are constant
2. Density is independent of time and location
3. No external/body forces

The first assumption gets rid of the conservation of energy equation. The second assumption drops the first term in the conservation of mass equation. This means that state variables in the fluid will only be velocities and pressure. The mass equation is reduced to an incompressibility condition. In other words, the velocity field is divergence free. This on its turn

drops the last term in Navier's law. Thus the incompressible NS⁵ equations can be written as follows:

$$\underbrace{\rho_f \frac{\partial v_i}{\partial t}}_{\text{Unsteady acceleration}} + \underbrace{\rho_f v_j \frac{\partial v_i}{\partial x_j}}_{\text{Convective acceleration}} = \underbrace{\frac{\partial \sigma_{ij}}{\partial x_j}}_{\text{Pressure gradient and Viscosity}} \rightarrow \text{momentum balance} \quad , \quad (3-37)$$

$$\frac{\partial v_i}{\partial x_i} = 0 \rightarrow \text{mass balance/incompressibility condtion} \quad , \quad (3-38)$$

where $\sigma_{ij} = -p\delta_{ij} + \mu(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i})$. Note that the relation for the stress tensor σ_{ij} is *Stokes' law* and that μ is the dynamic viscosity⁶. Since there is a linear relation between stress tensor and the strain rate tensor, we are considering a Newtonian fluid. *Additionally, in this research, we solve for the static steady state solution, i.e. time integration is dropped*⁷. The steady state solution is found by solving the equations in Eq. (3-39) and Eq. (3-40):

$$\rho_f v_j \frac{\partial v_i}{\partial x_j} - \frac{\partial \sigma_{ij}}{\partial x_j} = 0 \quad , \quad (3-39)$$

$$\frac{\partial v_i}{\partial x_i} = 0 \quad , \quad (3-40)$$

where again $\sigma_{ij} = -p\delta_{ij} + \mu(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i})$ and the time dependent terms are dropped. Modeling these incompressible NS equations with Finite Element Method (FEM) has been studied extensively over the last 30 years. Although this research does not focus the mathematical details on this particular topic, some interesting remarks are [Donea and Huerta, 2004]:

1. The presence of non-linear and non-symmetric convective terms in the momentum equation can give rise to numerical instabilities. Multiple options to solve this problem have been proposed, but in this research Streamline-Upwind/Petrov Galerkin (SUPG) has been used, which introduces artificial diffusion in convection dominated cases.
2. The incompressibility condition also can give rise to numerical instabilities. Pressure in the incompressible NS equations is not related to any constitutive relation. Its presence in the momentum equation has the purpose of introducing an additional DOF to satisfy the incompressibility constraint. The role of the pressure variable is to adjust itself instantaneously, in order to satisfy the condition of divergence-free velocity. Pressure acts as a Lagrange Multiplier of the incompressibility constraint and thus there is *coupling* between the velocity and the pressure unknowns, which could lead to instabilities when equal order interpolation is used. This problem is handled by Pressure-Stabilized/Petrov Galerkin (PSPG) stabilization.
3. In Eq. (3-38), one sees that *only spatial derivatives* of the pressure are present in the momentum equation. This could lead to negative pressures in the solution of the model.

⁵To get an idea of incompressible flow check out: http://www.youtube.com/watch?v=p08_KlTKP50

⁶In the full NS equations the stress tensor σ_{ij} is defined as $\sigma_{ij} = -p\delta_{ij} + \tau_{ij}$. In this notation, the isotropic part $-p\delta_{ij}$ and the non-isotropic or *deviatoric* part τ_{ij} are separated.

⁷The MATLAB code is setup for time integration but with a very large time step, we only solve for the steady state solution

Physically, these negative pressure do not have any relevance, but instead one should look at the pressure differences. Suppose one adds a constant reference pressure to the pressure in Eq. (3-37), the derivative operator will kick this constant out of the equation and hence this should not affect to solution for the velocity field. However, in this way it is able to get rid of the negative pressure, which intuitively makes no sense at first.

Both these stabilization techniques are based on perturbing the weight functions in the weak formulation. These stabilization techniques are standard procedure in FEM modeling for FSI and hence are considered as a black box by the author. Details on these stabilization techniques are beyond the scope of this work and for that the reader is referred to, for instance, Donea and Huerta [2004] and Fries and Matthies [2005].

Static non dimensional weak formulation - Fluid

It is quite common to work with non dimensional parameters. This makes the equations simpler and highlights the most important/dominant terms. The non dimensional variables that have been defined in this research are presented in App. A. Choices for the particular reference parameters depend on what one expects to be the most dominant physical phenomenon. This is important, especially for time dependent values in the NS equations. The choices in this research are mostly based on dynamics of the fluid. Plugging these non dimensional variables into Eq. (3-39) and Eq. (3-40), using the *Gauss Theorem* and *integration by parts* gives the following formula for the residual defined in Section 2-1-1:

$$R_f^\Omega = \left[\begin{array}{c} \int_{\Omega} \left(\delta v_i \rho_f \left(\frac{\partial v_i}{\partial x_j} v_j \right) + \frac{1}{2} \left(\frac{\partial \delta v_i}{\partial x_j} + \frac{\partial \delta v_j}{\partial x_i} \right) \sigma_{ij} \right) d\Omega \\ \int_{\Omega} \delta p \frac{\partial v_i}{\partial x_i} d\Omega \end{array} \right] \quad (3-41)$$

and

$$R_f^\Gamma = \left[\begin{array}{c} \int_{\Gamma} \delta v_i n_j \left(-p \delta_{ij} + \mu \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \right) d\Gamma \\ 0 \end{array} \right] , \quad (3-42)$$

with $\mu = 1/Re$.⁸ The total elemental static residual is than given as:

$$R_f = R_f^\Omega \cup R_f^\Gamma . \quad (3-43)$$

The equations above are presented in non-dimensional variables and throughout the rest of the report the equations are dimensionless unless stated otherwise. The derivation of the residual equations can be found in App. A. One can see that the total elemental residual is a combination of both volume and boundary residuals, which is a direct result of the *Gauss theorem* and integration by parts. These boundary integrals make it possible to incorporate boundary conditions into the system (see next Section 3-2-3). Note the *static* indicates that residual is time *independent* and only regards the steady state solution. This residual is numerically computed using *Gaussian quadrature*.

⁸*Re* is the non-dimensional Reynolds number. The Reynolds number is the ratio between inertia and viscous forces.

Structural mechanics

For the structure we choose a linear elastic model. We consider a structural model that neglects inertia terms and damping terms, i.e. no time dependent terms, and looks as follows:

$$-\frac{\partial \sigma_{ij}}{\partial x_j} = 0 \quad , \quad (3-44)$$

where $\sigma_{ij} = \mathbf{C} : \boldsymbol{\epsilon} = C_{ijkl}\epsilon_{kl}$ with C the fourth order Right Cauchy-Green tensor and $\boldsymbol{\epsilon}$ is the Euler Almansi strain tensor to denote Hooke's law for continuous media. Actually, Eq. (3-44) is the abstract representation of the constitutive equation stating stiffness matrix times displacement equals external forces (here zero). Neglecting the time dependent terms is motivated by a low Reynolds number that indicates laminar flow. In this research on 2D FSI problems, a plane strain model is chosen for the structure. This model assumes that the normal and shear strains related to the third dimension z are zero. Linear elasticity theory assumes small deformations and a linear relation between stress and strain.

Static non dimensional weak formulation - Structure

Without presenting intermediate steps the nodal static residual for the structure is:

$$\begin{aligned} R_s &= R_s^\Omega \cup R_s^\Gamma = \int_{\Omega} \delta u_i \left(-\frac{\partial \sigma_{ij}}{\partial x_j} \right) d\Omega \\ &= \int_{\Omega} \delta u_i \left(\sigma_{ij} \frac{\partial \delta u_i}{\partial x_j} \right) d\Omega - \int_{\Gamma} \sigma_{ij} \delta u_i n_j d\Gamma \quad , \quad (3-45) \end{aligned}$$

where $\sigma_{ij} = \sigma_{ji}$. The residuals for fluid and structure can be discretized in space using XFEM, see Section 2-1. This discretization turns all the equations above into vector equations.

3-2-2 Initialization and boundary conditions

In order to solve the governing equations in the previous section, an initial solution and boundary conditions (BCs) need to be specified. The boundary conditions are directly incorporated into the residual by means of the boundary terms, see Eq. (3-43) and Eq. (3-45). Initialization determines the state of the system at the start of the solving process. The initialization in this research has the following form:

$$v_i^f = v_{i,0}^f \rightarrow \text{velocity for fluid in } \Omega \quad , \quad (3-46)$$

$$p^f = p_0^f \rightarrow \text{pressure for fluid in } \Omega \quad , \quad (3-47)$$

$$u_i^s = u_{i,0}^s \rightarrow \text{displacement for structure in } \Omega \quad , \quad (3-48)$$

$$\dot{u}_i^s = \dot{u}_{i,0}^s \rightarrow \text{velocity for structure in } \Omega \quad . \quad (3-49)$$

Two remarks on these pretty straightforward equations are:

1. The pressure initialization can be used to prevent negative pressures in the solution;

2. We consider \dot{u}_i also as a independent variable for the structure in a state space representation, which comes in handy when the fluid and structure are coupled through no-slip boundary conditions. The structural velocities in initial and steady state configuration are zero, as if the structure was a rigid wall. If the system is also solved for time, the structural velocities at time steps prior to steady state will not be zero. The relation between u_i and \dot{u}_i needs to be defined for time integration purposes and for details on this the reader is referred to Aloun [2012]. In this research, the relation is not needed, as the the structural velocities are set to zero.

Boundary conditions need to be introduced to cope with the spatial derivatives. These conditions can be subdivided into *Dirichlet* or *Neumann* boundary conditions. Dirichlet boundary conditions specify the solution of the problem on (a certain part of) the boundary Γ , while Neumann boundary conditions specify the derivative of the solution of the problem on (a certain part of) the boundary Γ :

$$v_i = v_{i,\text{in/out}} \rightarrow \text{fluid velocity at in or outlet on } \Gamma \quad , \quad (3-50)$$

$$v_i = v_{i,\text{rigid wall}} = 0 \rightarrow \text{fluid velocity at rigid wall on } \Gamma \quad , \quad (3-51)$$

$$p = p_{\text{in/out}} \rightarrow \text{fluid pressure at in or outlet on } \Gamma \quad , \quad (3-52)$$

$$u_i = u_{i,\text{fixed}} \rightarrow \text{fixed structural displacement } \Gamma \quad . \quad (3-53)$$

Eq. (3-50) can represent a parabolic velocity profile at the inlet and Eq. (3-51) represents the *no-slip* condition. Eq. (3-52) can represent setting the outlet pressure equal to the pressure of the environment outside the domain. Eq. (3-53) may denote a clamping condition for the structure. Application of these regular boundary conditions is common practice in FEM modeling of fluids and therefore considered as a black-box by the author. In addition to these boundary conditions at the standard boundaries of domain, boundary conditions at the fluid structure interface are defined.

Fluid structure interface conditions

In FSI, the communication between the structure and fluid is key. Two relations are responsible for the FSI coupling, namely:

$$v_i = \dot{u}_i \rightarrow \text{no-slip on } \Gamma_{sf} \quad , \quad (3-54)$$

$$\sigma_{ij}^f n_i = \sigma_{ij}^s n_i \rightarrow \text{traction balance on } \Gamma_{sf} \quad . \quad (3-55)$$

The first equation states that the fluid has to follow the structure with equal velocity, i.e. the fluid is not allowed to leave or penetrate the structure, and the second equation represents a traction balance between fluid and structure. In case of using XFEM, the interface between fluid and structure lies within intersected elements. This makes the coupling more complicated and Gerstenberger and Wall [2010] have developed a method to work with these interface conditions, which will be the topic of Section 3-2-3.

3-2-3 FSI coupling in XFEM

The projection of the deformed structure onto the fluid mesh results in intersected elements for the fluid, as can be seen in Figure 3-15. These intersected elements are partially turned

off by XFEM and at the material to void interface the actual FSI takes place according to Eq. (3-54) and Eq. (3-55). These interface boundary conditions give a direct contribution to the residuals in Eq. (3-43) and Eq. (3-45). However, these residuals contain boundary terms related to the regular element boundary Γ , while the intersected boundary elements have a *embedded* boundary, called Γ^+ . This means the Dirichlet boundary condition cannot be applied directly. The elemental contributions from intersected elements (the residuals

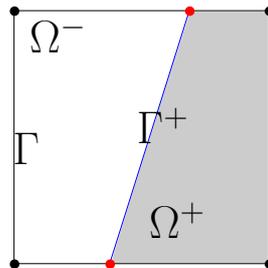
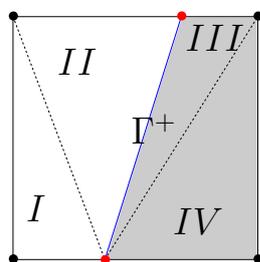


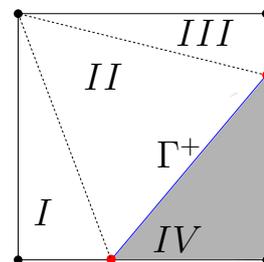
Figure 3-15: Intersected element with embedded boundary Γ^+

without boundary terms) are approximated by Gaussian quadrature after triangulation of the active part of the element based on the zero contour of the LSF. The contribution of each triangle is computed using 7 Gauss points. The void part of an element does not give a contribution to the integration of the residual. Depending on the intersection configuration the triangulation can look as shown in Figure 3-16. Two interesting notes to make here are:

1. The use of XFEM and LSM essentially shifts the effort in FEM modeling from remeshing to numerical integration in the intersected elements. The domain that is turned off is deleted from the solution as mentioned in Section 2-1-2. The shape functions in the void domain can be considered as zero, see also Kreissl and Maute [2012].
2. To compute the solutions at the Gauss points, the standard FEM nodes are used. The enriched nodes from Eq. (2-12) are only used to visualize the solution at the interface. This only works for models, where parts of the domain are turned off.



(a) Triangulation with opposing intersected element edges - Area of integration is area I plus area II



(b) Triangulation with neighbouring intersected element edges - Area of integration is area I plus area II plus area III.

Figure 3-16: Triangulation of intersected elements into 4 areas for integration purposes

No-slip condition at the fluid-structure interface

The no-slip condition at the fluid-structure interface, i.e. Eq. (3-54), gives residual contributions to the fluid via the structural velocities at the interface. This follows from the weak formulation of the Dirichlet boundary condition. For an embedded interface this boundary condition looks as follows:

$$(v_i n_i)_{\Gamma^+} = (\dot{u}_i n_i)_{\Gamma^+} \quad , \quad (3-56)$$

where v_i are the fluid velocities in x- and y-direction, \dot{u}_i the structural velocities in x- and y-direction and n_i is the normal to the interface Γ^+ . As mentioned, a state space representation is used, where the structural velocities are introduced independently. They are not part of the solution, but they do give contributions to the system at the embedded interface. Gerstenberger and Wall [2010] have developed a method to handle the no-slip boundary constraint at the embedded interface. *The main concept of this method is to introduce a new stress field σ_{ij}^σ as an unknown and use this new stress field to define a Lagrange Multiplier to enforce the Dirichlet boundary condition.* The stress field and displacement field are coupled by matching the strain tensors computed from each velocity field to condense out the contribution of additional stress field. The procedure looks as follows:

1. In the NS equations, the Cauchy stress tensor is defined as:

$$\sigma_{ij}^f = -p\delta_{ij} + \mu \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \quad . \quad (3-57)$$

2. The strain based on the fluid velocities is:

$$\epsilon_{ij}^v = \frac{1}{2} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \quad . \quad (3-58)$$

3. The strain based on the new unknown stress field σ_{ij}^σ is defined as:

$$\epsilon_{ij}^\sigma = \frac{1}{2\mu} (\sigma_{ij}^\sigma + p\delta_{ij}) \quad . \quad (3-59)$$

4. And the Lagrange Multiplier is defined as:

$$\lambda_i = n_j \sigma_{ij}^\sigma \quad . \quad (3-60)$$

With the equations above the total weak formulation for an intersected element can be explicitly written as:

$$\begin{aligned} & \int_{\Omega^-} \left(\delta v_i \rho_f \left(\frac{\partial v_i}{\partial t} + v_j \frac{\partial v_i}{\partial x_j} \right) + \frac{1}{2} \left(\frac{\partial \delta v_i}{\partial x_j} + \frac{\partial \delta v_j}{\partial x_i} \right) \sigma_{ij}^f + \delta p \frac{\partial v_i}{\partial x_i} \right) d\Omega^- - \int_{\Gamma} \delta v_i n_j \sigma_{ij}^f d\Gamma \\ & - \int_{\Gamma^+} \delta v_i n_j \sigma_{ij}^\sigma d\Gamma^+ \rightarrow \text{boundary term analogous to } \Gamma \\ & - k \int_{\Omega^-} \gamma_{ij} (\epsilon_{ij}^\sigma - \epsilon_{ij}^v) d\Omega^- \rightarrow \text{compatibility condition} \\ & - \int_{\Gamma^+} \gamma_{ij} n_j (v_i - \dot{u}_i) d\Gamma^+ \rightarrow \text{boundary condition} \quad , \end{aligned} \quad (3-61)$$

where γ_{ij} is the weighing function associated with σ_{ij}^σ , k is a constant compatibility scaling factor to improve numerical stability and the boundary condition corresponds to Eq. (3-56). Since we have introduced σ_{ij}^σ as an unknown, we need to find a solution for it. We continue by condensing out the new stress field, on an element level, by solving:

$$k \int_{\Omega^-} \gamma_{ij} (\epsilon_{ij}^\sigma - \epsilon_{ij}^v) d\Omega + \int_{\Gamma^+} \gamma_{ij} n_j (v_i - \dot{u}_i) d\Gamma^+ = 0 \quad . \quad (3-62)$$

Once σ_{ij}^σ is determined it can be substituted into:

$$- \int_{\Gamma^+} \delta v_i n_j \sigma_{ij}^\sigma d\Gamma^+ \quad (3-63)$$

to give the elemental contribution to the residual for an intersected element. The implementation of the system of equations of Eq. (3-62), after discretization, can be found in Appendix B⁹.

Numerical integration with Gaussian quadrature

The integral in Eq. (3-63) to determine the residual is approximated using a three-point Gaussian integration, as shown in Figure 3-17. After the coordinates and the weights of the Gauss points are determined with respect to fluid element, the structural velocities \dot{u}_i at the Gauss points are interpolated. Using the shape functions of the positive phase N_i^+ of the structural mesh and the nodal values of the structural element, associated with the structural bar, the structural velocities are determined at the interface and transferred to the fluid to give both residual and Jacobian contributions. In Figure 3-17, the fluid-structure interface is

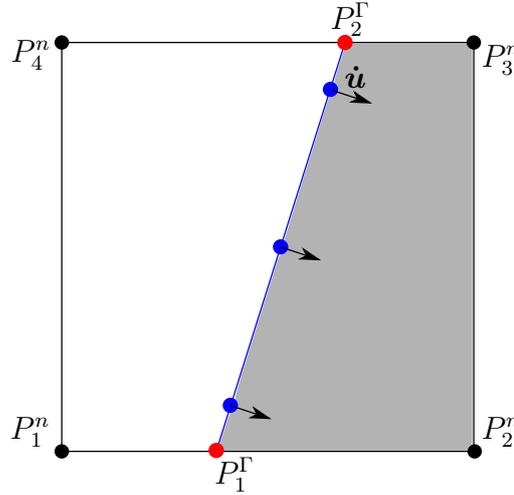


Figure 3-17: Gauss points (blue dots) on the interface for numerical integration of the no-slip condition in fluid elements.

perceived identically by the fluid and structural mesh, but it was already shown in Section 3-1 the implemented method of projecting the deformed structural mesh onto the fixed fluid mesh can result in intersection case as shown in Figure 3-18. The dashed red line depicts the fluid-structure surface as perceived by the fluid, the blue lines represent structural bars

⁹The author has not done any work on this particular part of the model and hence cannot be considered as an expert. This section is included as it is an essential part of the model, but far from trivial.

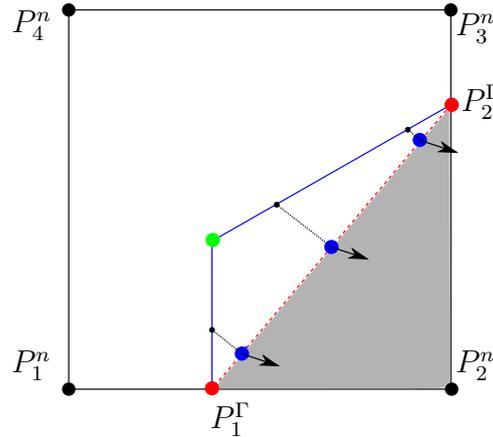


Figure 3-18: Projected Gauss points (blue dots) on the interface for numerical integration of the no-slip condition in fluid elements with a discrepancy between the fluid interface and the actual location of the structural bars.

with the green dot as bar endpoint. From this surface, orthogonal projection is done onto the fluid-structure interface of the structural mesh. This orthogonal projection gives the point on the other mesh closest to the Gauss point. This projection point holds the information, found by interpolation from the bar ending points, on structural velocities and is transferred to the fluid mesh. This discrepancy between the two meshes is a source of inaccuracy, but when the geometric details of the structure (corners) are sufficiently bulky (fine mesh) compared to the fluid elements, the error will be less significant.

Traction condition at the fluid-structure interface

The traction boundary condition is handled in similar fashion as the no-slip boundary condition, except a two-point integration is used. The fluid traction is applied as a residual contribution to the whole system. By means of Lagrange Multipliers, the fluid traction is applied to the structure as an external force, according to the setup of Maute et al. [2003]. The Jacobian contribution is determined by the derivative of the structural residual with respect to the fluid state variables. Figure 3-19 shows a structural sub-bar in an intersected fluid element. Again in this configuration the fluid and structure perceive the interface identically. The sub-bar is associated to a structural element, but the structural element is not shown in Figure 3-19. The traction vector at a Gauss point on the interface is defined as:

$$\mathbf{t}^{G,p} = (\boldsymbol{\sigma}_f \cdot \mathbf{n})^{G,p} \quad , \quad (3-64)$$

where superscript G,p denotes the p-th Gauss point on this sub-bar, $\boldsymbol{\sigma}_f$ denotes the regular fluid stress tensor, defined in Eq. (3-57), and \mathbf{n} the normal vector to the interface. The traction vector is visualized in Figure 3-19. As was shown in Figure 3-8, every structural bar, associated with a particular structural element, was subdivided into sub-bars using the projected intersection points. Effectively, this means that sub-bars lie within different fluid elements and therefore the residual contribution is computed per sub-bar. The residual

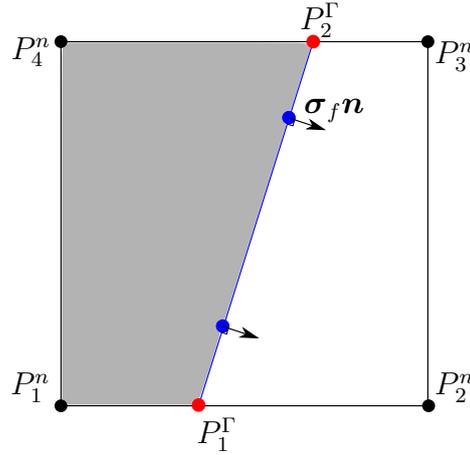


Figure 3-19: Gauss points (blue dots) on structural bar and the bar is defined between the intersections points P_1^Γ and P_2^Γ within a fluid element

contribution of the traction of one sub-bar can be written as follows:

$$\mathbf{R}_s^{SB} = \int_{\Gamma^+} \delta u \cdot \mathbf{t}^f d\Gamma^+ = \sum_p^k \mathbf{N}_{G,p}^- \cdot \mathbf{t}^{G,p} \cdot w^{G,p} \quad , \quad (3-65)$$

where $w^{G,p}$ denote the weight of the p-th Gauss point of the sub-bar and k is the number of Gauss points on the sub-bar. In order to compute this stress tensor, the pressure and velocities at the Gauss points need to be interpolated from the nodal values at nodes $P_1^n \dots P_4^n$ on the fluid mesh, again using the FE element interpolation with shape functions \mathbf{N}^- from the negative phase, i.e. the fluid domain. This looks as follows:

$$\mathbf{v}^{G,p} = (\mathbf{N}^- \cdot \mathbf{v})^{G,p} \quad , \quad (3-66)$$

$$p^{G,p} = (\mathbf{N}^- \cdot p)^{G,p} \quad , \quad (3-67)$$

where \mathbf{v} and p are the nodal values for fluid velocity and pressure, respectively. For the fluid traction vector we need the spatial derivatives of the velocities defined above:

$$\left(\frac{\partial \mathbf{v}}{\partial x_i} \right)^{G,p} = \left(\frac{\partial \mathbf{N}^-}{\partial x_i} \cdot \mathbf{v} \right)^{G,p} \quad , \quad (3-68)$$

where x_i , with $i = 1..2$, indicates x and y , respectively. When the equations above are plugged into the traction vector Eq. (3-57) and derived with respect to the fluid Degrees Of Freedom (DOFs) the result is the following:

$$\left(\frac{\partial \mathbf{t}}{\partial p} \right)^{G,p} = \begin{bmatrix} -n_1 \mathbf{N}^- \\ -n_2 \mathbf{N}^- \end{bmatrix}^{G,p} \quad , \quad (3-69)$$

$$\left(\frac{\partial \mathbf{t}}{\partial v_x} \right)^{G,p} = \begin{bmatrix} 2\mu \frac{\partial \mathbf{N}^-}{\partial x} n_1 + \mu \frac{\partial \mathbf{N}^-}{\partial y} n_2 \\ \mu \frac{\partial \mathbf{N}^-}{\partial y} n_1 \end{bmatrix}^{G,p} \quad , \quad (3-70)$$

$$\left(\frac{\partial \mathbf{t}}{\partial v_y} \right)^{G,p} = \begin{bmatrix} \mu \frac{\partial \mathbf{N}^-}{\partial x} n_2 \\ 2\mu \frac{\partial \mathbf{N}^-}{\partial y} n_2 + \mu \frac{\partial \mathbf{N}^-}{\partial x} n_1 \end{bmatrix}^{G,p} . \quad (3-71)$$

The terms above are needed to find the Jacobian contribution of the traction. Again we also consider the case where the perceived interfaces are different. Figure 3-20 depicts the situation.

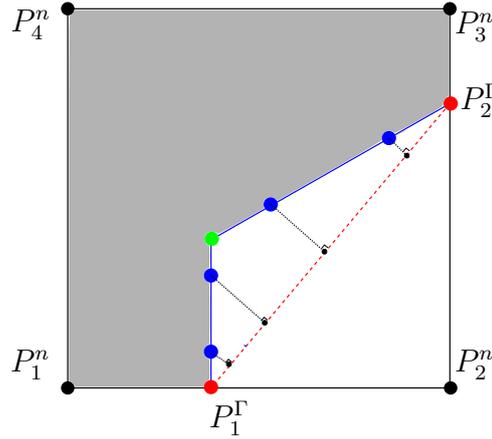


Figure 3-20: Projected Gauss points (blue dots) on the structural bars for numerical integration of the traction condition in structural elements with a discrepancy between the fluid interface and the actual location of the structural bars.

Projected Gauss points (blue dots) on the interface for numerical integration of the no-slip-condition in fluid elements with a discrepancy between the fluid interface and the actual location of the structural bars.

Essentially, the procedure is exactly the same as for the no-slip-condition, but in this case the Gauss points are computed on the sub-bars and projected onto the fluid mesh. Again these projection points provide the information velocities and pressure to calculate the fluid stress tensor and henceforth the structural residual and Jacobian contribution.

Numerical integration with Gaussian quadrature

Eq. (3-65) showed that the residual is approximated by Gaussian quadrature as follows:

$$\mathbf{R}_s^{SB} = \sum_p^k \mathbf{N}_{G,p}^- \cdot \mathbf{t}^{G,p} \cdot w^{G,p} , \quad (3-72)$$

where $w^{G,p}$ denotes the weight of the p-th Gauss point of the sub-bar and k is the number of Gauss points on the sub-bar. The residual for the parental structural element is found by summation of the all sub-bars in the element:

$$\mathbf{R}_s^{e+} = \sum_i^{ns} \mathbf{R}_{s,i}^{SB} , \quad (3-73)$$

where ns is the number of sub-bars of the parental structural bar. The Jacobian contributions from the sub-bars to the parental element follow from the residuals as follows:

$$\left(\frac{\partial \mathbf{R}_s^{SB}}{\partial v_x} \right) = \sum_p^k \mathbf{N}_{G,p}^- \cdot \left(\frac{\partial \mathbf{t}}{\partial v_x} \right)^{G,p} \cdot w^{G,p} \quad , \quad (3-74)$$

$$\left(\frac{\partial \mathbf{R}_s^{SB}}{\partial v_y} \right) = \sum_p^k \mathbf{N}_{G,p}^- \cdot \left(\frac{\partial \mathbf{t}}{\partial v_y} \right)^{G,p} \cdot w^{G,p} \quad , \quad (3-75)$$

$$\left(\frac{\partial \mathbf{R}_s^{SB}}{\partial p} \right) = \sum_p^k \mathbf{N}_{G,p}^- \cdot \left(\frac{\partial \mathbf{t}}{\partial p} \right)^{G,p} \cdot w^{G,p} \quad . \quad (3-76)$$

This leads to the elemental Jacobian contribution as follows:

$$\left(\frac{\partial \mathbf{R}_s^e}{\partial v_x} \right) = \sum_i^k \left(\frac{\partial \mathbf{R}_{s,i}^{SB}}{\partial v_x} \right) \quad , \quad (3-77)$$

$$\left(\frac{\partial \mathbf{R}_s^e}{\partial v_y} \right) = \sum_i^k \left(\frac{\partial \mathbf{R}_{s,i}^{SB}}{\partial v_y} \right) \quad , \quad (3-78)$$

$$\left(\frac{\partial \mathbf{R}_s^e}{\partial p} \right) = \sum_i^k \left(\frac{\partial \mathbf{R}_{s,i}^{SB}}{\partial p} \right) \quad , \quad (3-79)$$

where again k is the number of sub-bars in the parental element and $\mathbf{R}_{s,i}^{SB}$ is the structural residual of the i -th sub-bar. It cannot be emphasized enough that the sub-bars are always associated with same parental structural element e , but the fluid element containing them differs depending on the structural projection. A structural bar can have multiple sub-bars located within different fluid elements, but all contribute to the same structural element. These Jacobian contributions are part of the complete Jacobian presented in Section 3-3.

3-3 Consistent Jacobian

As mentioned earlier, the goal is to solve the FSI system with a monolithic setup. To do so, a numerically consistent Jacobian is desired. A Jacobian, in mathematical terms, is the matrix of all first-order partial derivatives of a vector-valued function. In this research, this relates to the partial derivatives of the residual with respect to state variables of the spatially discretized problem. Numerically consistent means that the Jacobian matrix contains the partial derivatives, that represent the slope of the residual function at the particular solution for the state variable (remember Figure 2-10). The motivation to find this numerically consistent Jacobian is that it provides high convergence rates for *Newton-Raphson-like* non-linear solvers and high accuracy for sensitivity analysis prior to optimization. In the previous sections, we have already built parts of the Jacobian of the system, according to the elemental and boundary condition contributions. This section will present an overview of the Jacobian contributions and some additional terms related to the fluid changing LSF.

3-3-1 Jacobian overview

In general and abstract notation, we can define our system in the following way:

$$\mathbf{R}_{tot} = \begin{bmatrix} \mathbf{R}_s \\ \mathbf{R}_f \end{bmatrix}, \quad \mathbf{u}^* = \begin{bmatrix} \mathbf{u}_s \\ \dot{\mathbf{u}}_s \\ \mathbf{u}_f \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}_s \\ \mathbf{u}_f \end{bmatrix}, \quad (3-80)$$

where \mathbf{R}_{tot} represents the system's total residual vector and the individual terms in \mathbf{u}^* and \mathbf{u} look as follows:

$$\mathbf{u}_s = \begin{bmatrix} \mathbf{u}_x \\ \mathbf{u}_y \end{bmatrix}, \quad \dot{\mathbf{u}}_s = \begin{bmatrix} \dot{\mathbf{u}}_x \\ \dot{\mathbf{u}}_y \end{bmatrix}, \quad \mathbf{u}_f = \begin{bmatrix} p \\ \mathbf{v}_x \\ \mathbf{v}_y \end{bmatrix}, \quad (3-81)$$

indicating structural displacements in x- and y-direction, structural velocities in x- and y-direction, fluid pressure, fluid velocity in x- and y-direction, respectively. The vector of state variables is represented by \mathbf{u} and it contains the discretized solution of the system. The asterisk * indicates state space representation, i.e. in \mathbf{u}^* , we have independently introduced the structural velocities $\dot{\mathbf{u}}_s$ (see item 2 in Section 3-2-2) to incorporate the no-slip condition at the embedded fluid-structure interface. The no-slip condition, at the embedded interface for steady state, implies:

$$\begin{bmatrix} \dot{\mathbf{u}}_x \\ \dot{\mathbf{u}}_y \end{bmatrix}_s = \begin{bmatrix} \mathbf{v}_x \\ \mathbf{v}_y \end{bmatrix}_f = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad (3-82)$$

omitting the need to define the relation between \mathbf{u}_s and $\dot{\mathbf{u}}_s$. In Figure 3-18 and Figure 3-20, it is shown that, due to the mismatch between the two meshes, a projection method is needed to transfer physical information from one material to the other. The structural residual contains elemental, regular boundary and fluid dependent traction contributions, and the fluid residual contains elemental, regular boundary and structural dependent no-slip condition contributions. This means that both fluid and structural residual depend on structural and fluid variables, but also on the fluid LSF¹⁰. This is represented in the following formula:

$$\mathbf{R}_{tot} = \begin{bmatrix} \mathbf{R}_s(\mathbf{u}_s, \mathbf{u}_f, \phi_f(\mathbf{u}_s)) \\ \mathbf{R}_f(\dot{\mathbf{u}}_s, \mathbf{u}_f, \phi_f(\mathbf{u}_s)) \end{bmatrix}. \quad (3-83)$$

Remembering that the fluid LSF ϕ_f changes as a function function of the structural displacements, we can define the Jacobian in state space using the chain-rule as follows:

$$\mathbf{J}^* = \frac{d\mathbf{R}_{tot}}{d\mathbf{u}^*} = \begin{bmatrix} \frac{d\mathbf{R}_s}{d\mathbf{u}_s} & \frac{d\mathbf{R}_s}{d\dot{\mathbf{u}}_s} & \frac{d\mathbf{R}_s}{d\mathbf{u}_f} \\ \frac{d\mathbf{R}_f}{d\mathbf{u}_s} & \frac{d\mathbf{R}_f}{d\dot{\mathbf{u}}_s} & \frac{d\mathbf{R}_f}{d\mathbf{u}_f} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{R}_s}{\partial \mathbf{u}_s} + \frac{\partial \mathbf{R}_s}{\partial \phi_f} \frac{d\phi_f}{d\mathbf{u}_s} & \frac{d\mathbf{R}_s}{d\dot{\mathbf{u}}_s} & \frac{d\mathbf{R}_s}{d\mathbf{u}_f} \\ \frac{\partial \mathbf{R}_f}{\partial \phi_f} \frac{d\phi_f}{d\mathbf{u}_s} & \frac{d\mathbf{R}_f}{d\dot{\mathbf{u}}_s} & \frac{d\mathbf{R}_f}{d\mathbf{u}_f} \end{bmatrix}, \quad (3-84)$$

where the different terms of the Jacobian are:

1. $\frac{\partial \mathbf{R}_s}{\partial \mathbf{u}_s}$ is the linear stiffness matrix of the structure;

¹⁰They also depend on the structural LSF, but this is fixed and therefore does not show up explicitly in the Jacobian

2. $\frac{\partial \mathbf{R}_s}{\partial \phi_f} \frac{d\phi_f}{d\mathbf{u}_s}$ is the Jacobian contribution due to the dependence of the traction on the changing fluid LSF (projection);
3. $\frac{d\mathbf{R}_s}{d\dot{\mathbf{u}}_s} = 0$, since the structural residual does not depend on the structural velocities;
4. $\frac{d\mathbf{R}_s}{d\mathbf{u}_f}$ is the Jacobian contribution of the traction interface condition, defined in Eq. (3-77) to Eq. (3-79);
5. $\frac{\partial \mathbf{R}_f}{\partial \phi_f} \frac{d\phi_f}{d\mathbf{u}_s}$ is the Jacobian contribution due to the dependence of the no-slip condition (projection) and the intersected fluid elements on the changing fluid LSF;
6. $\frac{d\mathbf{R}_f}{d\dot{\mathbf{u}}_s}$ is the Jacobian contribution of the no-slip condition, described in Section 3-2-3, where $\dot{\mathbf{u}}_s = \mathbf{v}_f$ from (3-82);
7. $\frac{d\mathbf{R}_f}{d\mathbf{u}_f}$ is the non-linear stiffness matrix of the fluid.

The consequences of the implemented LSM are clearly visible in Eq. (3-84) as two LSF dependent terms show up, namely term 2 and 5. For a numerically consistent Jacobian all these terms need to be incorporated. The most interesting terms in Eq. (3-84) are terms 2 and 5 and will be referred to as the LSF Jacobian terms (LSFJ). Focusing on term 5, it written out as follows:

$$\frac{\partial \mathbf{R}_f(\dot{\mathbf{u}}_s, \mathbf{u}_f, \phi_f(\mathbf{u}_s))}{\partial \mathbf{u}_s} = \frac{\partial \mathbf{R}_f}{\partial \phi_f} \frac{d\phi_f}{d\mathbf{u}_s} \quad (3-85)$$

The first part of the term on the right-hand side Eq. (3-85) is determined using finite differences due to a lack of analytical dependencies. This finite differencing is straightforward, however, one should take care of preventing sign changes of the nodal levelset value, due to the perturbation. A sign change would mean that the node in question has moved from fluid to void domain or vice versa and that should be prevented. Prevention is done by switching between forward, backward and central differencing, if a sign change occurs. The second part of the term is determined analytically according to the method described in Section 3-3-2, using the results from Section 3-1-4 and can be regarded as the Jacobian of the fluid LSF.

In this research, term 6 is ignored and hence will not be part of the analytic Jacobian. In Section 5-3, this term will be computed with finite differences for a simple test problem and it will be shown that this term contributes little to the Jacobian, compared to the stiffness matrix. This motivates the choice to ignore it and it is expected that the consequences for the monolithic solving process will be minimal. Strictly speaking, however, this does imply that a numerically consistent Jacobian is not built in this work. The Jacobian, in state space, can now be rewritten to:

$$\mathbf{J}^* = \begin{bmatrix} \frac{\partial \mathbf{R}_s}{\partial \mathbf{u}_s} + \frac{\partial \mathbf{R}_s}{\partial \phi_f} \frac{d\phi_f}{d\mathbf{u}_s} & \frac{d\mathbf{R}_s}{d\dot{\mathbf{u}}_s} & \frac{d\mathbf{R}_s}{d\mathbf{u}_f} \\ \frac{\partial \mathbf{R}_f}{\partial \phi_f} \frac{d\phi_f}{d\mathbf{u}_s} & \frac{d\mathbf{R}_f}{d\dot{\mathbf{u}}_s} & \frac{d\mathbf{R}_f}{d\mathbf{u}_f} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{R}_s}{\partial \mathbf{u}_s} & 0 & \frac{d\mathbf{R}_s}{d\mathbf{u}_f} \\ \frac{\partial \mathbf{R}_f}{\partial \phi_f} \frac{d\phi_f}{d\mathbf{u}_s} & \frac{d\mathbf{R}_f}{d\dot{\mathbf{u}}_s} & \frac{d\mathbf{R}_f}{d\mathbf{u}_f} \end{bmatrix} \quad (3-86)$$

Since we are only interested in the structural displacements and not in the structural velocities, the state space representation needs to be dropped. For this we make use of Eq. (3-82), where

an explicit relation between velocities is shown, which allows us to add the no-slip condition contribution to fluid velocity related terms, as follows:

$$\mathbf{J} = \frac{d\mathbf{R}_{tot}}{d\mathbf{u}} \begin{bmatrix} \frac{d\mathbf{R}_s}{d\mathbf{u}_s} & \frac{d\mathbf{R}_s}{d\mathbf{u}_f} \\ \frac{\partial \mathbf{R}_f}{\partial \phi_f} \frac{d\phi_f}{d\mathbf{u}_s} & \frac{d\mathbf{R}_f}{d\dot{\mathbf{u}}_s} + \frac{d\mathbf{R}_f}{d\mathbf{u}_f} \end{bmatrix}, \quad (3-87)$$

where, according to Eq. (3-80) the lower right term, in detail, looks as follows:

$$\frac{d\mathbf{R}_f}{d\dot{\mathbf{u}}_s} + \frac{d\mathbf{R}_f}{d\mathbf{u}_f} = \left[\frac{d\mathbf{R}_f}{d\mathbf{p}} \left(\frac{d\mathbf{R}_f}{d\mathbf{v}_x} + \frac{d\mathbf{R}_f}{d\dot{\mathbf{u}}_x} \right) \left(\frac{d\mathbf{R}_f}{d\mathbf{v}_y} + \frac{d\mathbf{R}_f}{d\dot{\mathbf{u}}_y} \right) \right], \quad (3-88)$$

with Eq. (3-82) to complete the Jacobian. The terms between the curved brackets consist of an elemental and no-slip condition Jacobian contribution.

3-3-2 Analytical terms

As mentioned, the Jacobian of the fluid LSF is computed analytically, because it can enhance numerical performance. The fluid levelset field setup, as presented in Section 3-1-4, allows to find $\partial\phi_f/\partial\mathbf{u}_s$ analytically. The fluid LSF is a function of the intersection points in an element as was shown in Figure 3-10. The intersection points are a function of the structural displacements. The derivative of ϕ_f with respect to the structural variables is rewritten as:

$$\left(\frac{d\phi_f}{d\mathbf{u}_s} \right)^e = \left(\frac{\partial\phi_f}{\partial \mathbf{P}_{1,2}^\Gamma} \frac{d\mathbf{P}_{1,2}^\Gamma}{d\mathbf{u}_s} \right)^e, \quad (3-89)$$

where i denotes the i -th intersection point and e denotes the e -th fluid element. Both terms on the right-hand side have been determined in Section 3-1-5. Per fluid element this derivative information is stored as follows: each element has 4 nodal levelset values for the fluid and 8 structural nodal displacements (2 spatial directions for the structural displacements). This gives two matrices of 4×4 for the fluid LS derivative with respect to the structural displacements:

$$\frac{\partial\phi_f}{\partial\mathbf{u}_x^s} = \begin{bmatrix} \frac{\partial\phi_1}{\partial u_{x1}^s} & \frac{\partial\phi_1}{\partial u_{x2}^s} & \frac{\partial\phi_1}{\partial u_{x3}^s} & \frac{\partial\phi_1}{\partial u_{x4}^s} \\ \frac{\partial\phi_2}{\partial u_{x1}^s} & \frac{\partial\phi_2}{\partial u_{x2}^s} & \frac{\partial\phi_2}{\partial u_{x3}^s} & \frac{\partial\phi_2}{\partial u_{x4}^s} \\ \frac{\partial\phi_3}{\partial u_{x1}^s} & \frac{\partial\phi_3}{\partial u_{x2}^s} & \frac{\partial\phi_3}{\partial u_{x3}^s} & \frac{\partial\phi_3}{\partial u_{x4}^s} \\ \frac{\partial\phi_4}{\partial u_{x1}^s} & \frac{\partial\phi_4}{\partial u_{x2}^s} & \frac{\partial\phi_4}{\partial u_{x3}^s} & \frac{\partial\phi_4}{\partial u_{x4}^s} \end{bmatrix} \quad (3-90)$$

and

$$\frac{\partial\phi_f}{\partial\mathbf{u}_y^s} = \begin{bmatrix} \frac{\partial\phi_1}{\partial u_{y1}^s} & \frac{\partial\phi_1}{\partial u_{y2}^s} & \frac{\partial\phi_1}{\partial u_{y3}^s} & \frac{\partial\phi_1}{\partial u_{y4}^s} \\ \frac{\partial\phi_2}{\partial u_{y1}^s} & \frac{\partial\phi_2}{\partial u_{y2}^s} & \frac{\partial\phi_2}{\partial u_{y3}^s} & \frac{\partial\phi_2}{\partial u_{y4}^s} \\ \frac{\partial\phi_3}{\partial u_{y1}^s} & \frac{\partial\phi_3}{\partial u_{y2}^s} & \frac{\partial\phi_3}{\partial u_{y3}^s} & \frac{\partial\phi_3}{\partial u_{y4}^s} \\ \frac{\partial\phi_4}{\partial u_{y1}^s} & \frac{\partial\phi_4}{\partial u_{y2}^s} & \frac{\partial\phi_4}{\partial u_{y3}^s} & \frac{\partial\phi_4}{\partial u_{y4}^s} \end{bmatrix}, \quad (3-91)$$

where $\partial\phi_i/\partial u_{yy}^s$ denotes the derivative of nodal levelset value of the i -th node with respect to y -displacement of the j -th structural node. For each element the matrices above can be found with Eq. (3-89). These matrices are multiplied with finite differenced term for the fluid residual to complete the Jacobian.

3-4 FSI solvers

In this research, two different solvers have been implemented and investigated, a *monolithic* and a *staggered* solver. Both solvers try to find the approximated FEM solution at steady state. They differ in which individual Jacobian terms are used and how this information is used to update the solution. The monolithic setup uses the complete Jacobian in Eq. (3-87) and updates fluid and structure at the same time. The staggered solver ignores the off-diagonal terms in Eq. (3-87) and updates the fluid and structure separately. The residuals, however, include all information in both solvers. The coupling information is the most challenging to implement and ignoring this information allows one to check other parts of the model without running into problems related to the ignored terms. A complete Jacobian, however, containing all information on the phases and coupling between the phases, is needed in the sensitivity analysis for optimization. In other words, if one wants to use this model for optimization purposes, the monolithic setup needs to be used. The staggered setup, on the other hand, provides useful insights into the model and can be used to benchmark the problem, if the residual build-up is correct. In Chapter 4, the results of the staggered solver are presented and in Chapter 5 the results of the monolithic setup are presented. This section is written to introduce the two different solvers and their characteristics, see Section 3-4-1 and Section 3-4-2.

3-4-1 Monolithic solver

A monolithic solver is the same as the classical Newton-Raphson solver presented in Section 2-3 for multivariable functions. Monolithic refers to the fact that the complete system of equations is solved at once. The complete Jacobian in Eq. (3-87) is used and it updates both phases simultaneously. It is a widely used solver in FEM and the challenge lies in finding the numerically consistent Jacobian semi-analytically. The next section presents the overview of the MATLAB-code, that has been used in this work.

Code overview - Monolithic

The structure of the code that has been implemented follows the steps presented in Section 2-1-1. In summary, the code looks as is shown in Figure 3-21. The model is set up in the pre-processing stage. In this stage, all the required information is gathered and defined to start Newton-Raphson solver. Within the Newton-Raphson solver the fluid levelset field is updated, the residual and Jacobian are computed and unless convergence is reached the solution is updated. As mentioned earlier, the fluid levelset field is updated according to the displacements of the previous solution. This results in a lag between the void in the fluid mesh and the structure, until convergence is reached. After convergence is reached the solution is visualized with *Paraview* for further analysis.

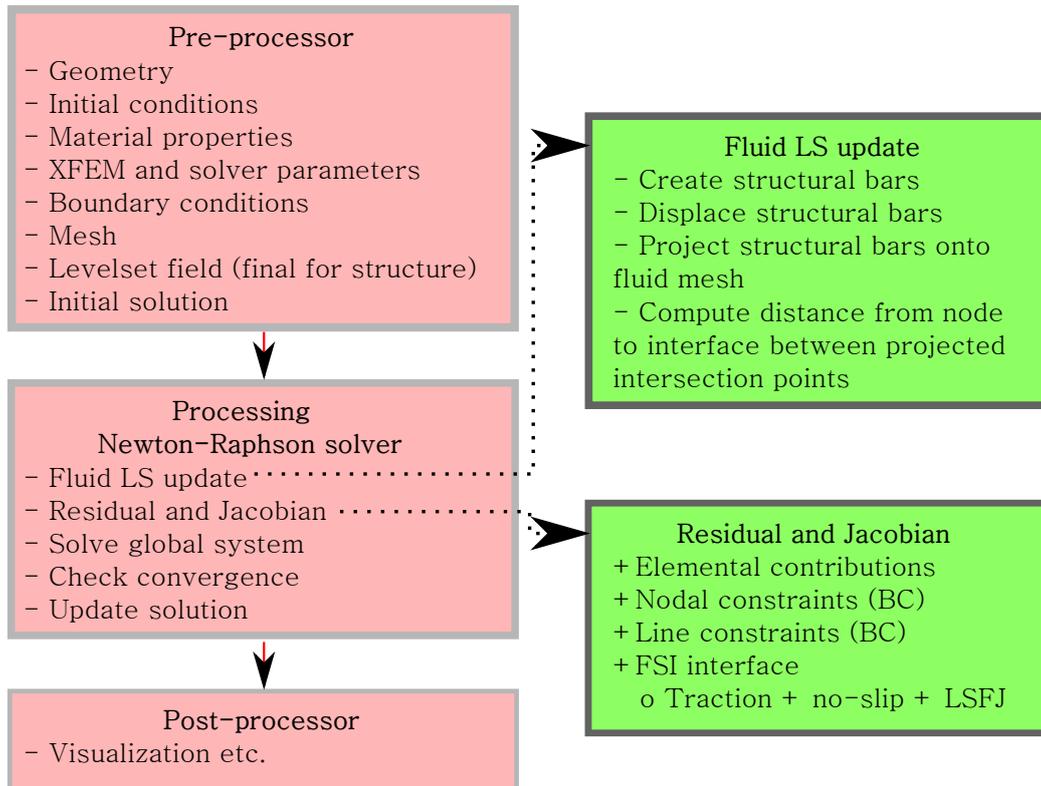


Figure 3-21: Code overview monolithic setup - The great majority of this MATLAB code was developed at the research group of Prof. Kurt Maute at the University of Colorado, Boulder, CO, USA

3-4-2 Staggered solver

The other solver used in this work is a *spatially staggered Newton-Raphson solver*. A staggered setup is a common method to solve FSI problems. Staggered schemes in general introduce an asynchronous solution between the flow and structure. In this research, we do not introduce a time shift, but a spatial shift. The fluid and structural problem are solved separately. The solver scheme is illustrated by Figure 3-22. In Figure 3-22, superscript ns indicates the Newton-Raphson update stage of the solver, where a fluid stage consists of multiple classical Newton-Raphson updates for the fluid and a structural stage of one Newton-Raphson update for the structure. Each Newton-Raphson update is computed with a direct solve. The dashed arrows indicate a transfer of information and the solid arrows a Newton-Raphson update stage. The small dots along the solid arrows represent a Newton-Raphson update. The process can be described as follows; at arrow 1 information on the location/displacements of the structure is transferred to be input for fluid Newton-Raphson solve stage at arrow 2. The results from stage 2 are then transferred with arrow 3 as an externally applied traction force to structure for the structure Newton-Raphson update stage at arrow 4. At arrow 5 a partial update for the structure is done and the process repeats itself until convergence. It takes two solve stages to update both fluid and structure and each solve stage consists of 1 to 3 Newton-Raphson updates. The Jacobian terms used for these Newton-Raphson updates

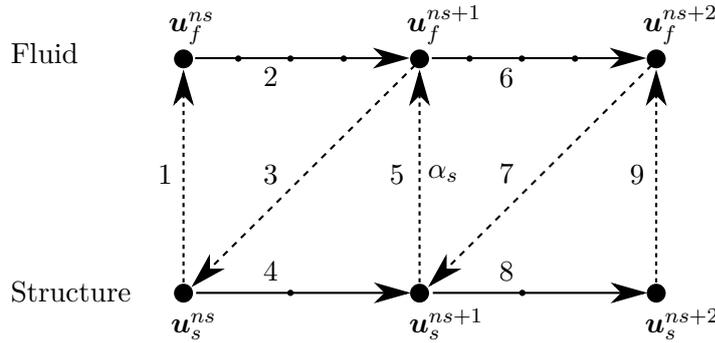


Figure 3-22: Staggered setup - Solid arrows indicate solution updates, dashed arrows indicate coupling information transfer. Individual Newton-Raphson updates are indicated with the small dots along the solid arrows.

are the diagonal terms in Eq. (3-87)¹¹. The partial update is done according to the following formula:

$$\mathbf{u}_s^{n+1} = \alpha_s \cdot \mathbf{u}_s^n + (1 - \alpha_s) \cdot \mathbf{u}_s^{n-1} \quad . \quad (3-92)$$

Eq. (3-92) essentially shows that the new structural solution is combination of the two previous solutions. This smooths the structural update process and prevents wiggling around the steady state solution, boosting convergence. Motivation to start with updating the fluid first is motivated by the fact that the initialization of the fluid has no physical relevance and hence neither does the applied traction. The result of the scheme is a converged steady state solution. The next section will give more details on how the code is, actually, structured from start to end of the solver.

Code structure - Staggered

The framework of the monolithic solver in Section 3-4-1 is used to set up the staggered solver. It is important to note that within the staggered Newton-Raphson solver, there is a switch that determines whether the fluid or the structure is updated, such that the update scheme is represented by Figure 3-22. Explicitly and chronologically written out the staggered Newton-Raphson solver looks as is shown in Alg. 1 in Appendix C. Each fluid stage multiple Newton-Raphson updates are done, until the fluid residual is smaller than ϵ_{th} . For the linear structure this ϵ_{th} threshold is reached after one Newton-Raphson update. The overall convergence is reached when the norm of the total residual and the structural update vector is small. This combination of convergence criteria is needed to make the two separated problems ‘communicate’. The criterion on the structural displacements also relates to the fluid problem, since the LSM presented in Section 3-1 depends on the structural displacements. The two problems may have small residual norms individually, but that does not guarantee a fluid levelset field that fits the structural projection. The staggered setup is not the most efficient FSI solver, but it does allow to track the update process and see the characteristics of the staggered solver. The difference between the staggered setup and the code presented in

¹¹The staggered solver is similar to the *block-Gauss-Seidell method*, which is often used in FSI modeling. Standard block-Gauss-Seidell methods, however, include all coupling terms in contrast to this staggered approach.

Figure 3-21 is that the staggered setup contains an additional loop over the Newton-Raphson solver and a switch to determine to update either the fluid or the structural solution.

3-5 The problem setup

In this research, the modeled physical problem has been taken from the COMSOL Multi-physics model library. The results from COMSOL are computed using the ALE method for FSI problems, which has proved itself in the past. The 2D geometry of the problem is shown in Figure 3-23. The third dimension has unit length. The problem consists of a horizontal flow channel with a vertical structural beam fixed at the bottom near the middle of the tunnel. The fluid flows from left to right and it imposes a force on the deformable structure, resulting from viscous forces and fluid pressure¹².

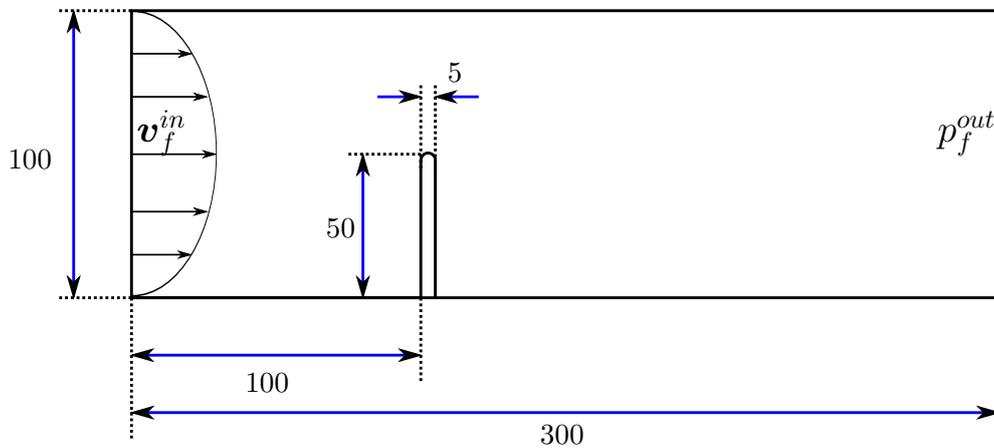


Figure 3-23: Geometry of the benchmark problem incl. boundary conditions at the in- and outlet
- Dimensions in μm

3-5-1 Boundary conditions

In order solve differential equations, boundary conditions need to be defined. These boundary conditions have a clear physical interpretation.

Fluid

The boundary conditions for the fluid are the following:

- Parabolic inlet conditions - On the left of the domain the fluid flows in with a parabolic velocity profile, see Figure 3-23;
- Zero pressure outlet - On the right of the domain the outlet pressure is zero¹³;

¹²A very cool video on this problem including optimization by Nick Jenkins (CU Boulder): <http://www.youtube.com/watch?v=qTBe3XoXmP4&feature=c4-overview&list=UU5UIdaNroLAvI1Fbk4XGfDg>

¹³This problem is part of the COMSOL Multiphysics library, but it is slightly adapted to be able to model the similar problem in MATLAB. The outlet pressure boundary condition is only zero pressure and does not include zero traction.

- No-slip condition walls - The fluid velocity is zero at the bottom and top horizontal walls, $\mathbf{v}_f = 0$ [m/s], to prevent the fluid from escaping the domain in the vertical direction;
- No-slip condition fluid-structure interface - The fluid velocity at the fluid-structure interface is equal to velocity of the structure, $\mathbf{v}_f = \dot{\mathbf{u}}_s$ [m/s], as the fluid is not allowed to leave or penetrate the structure.

Structure

The boundary conditions for the structure are the following:

- Clamped beam - The structure is clamped at the bottom fixing all DOFs, i.e. $\mathbf{u}_s^{wall} = 0$;
- Traction condition fluid-structure interface - The fluid applies an external force $\boldsymbol{\sigma}_f \cdot \mathbf{n}$ to the structure at the fluid-structure interface.

3-5-2 Material parameters

The material parameters describe the characteristics of the materials used to model the problem. In the XFEM code, all these parameters are non-dimensionalized, see also App. A.

Fluid

The fluid parameters are the following:

- Fluid density - $\rho_f = 1000$ [kg/m³];
- Dynamic viscosity - $\mu = 0.001$ [Pa · s];
- Velocity mean of parabolic inlet - $v_f^{mean} = 0.0333$ [m/s];
- Reynolds number - $Re = 1.11$ [-];

Structure

The structural parameters are the following:

- Young's modulus - $E = 2 \cdot 10^5$ [Pa];
- Poisson's ratio - $\nu = 0.33$ [-].

As mentioned the structure is linear elastic with a plane strain model. This Young's modulus is very low (very flexible material) as it is 10^6 times lower than that of steel. The Poisson's ratio is equal to that of steel.

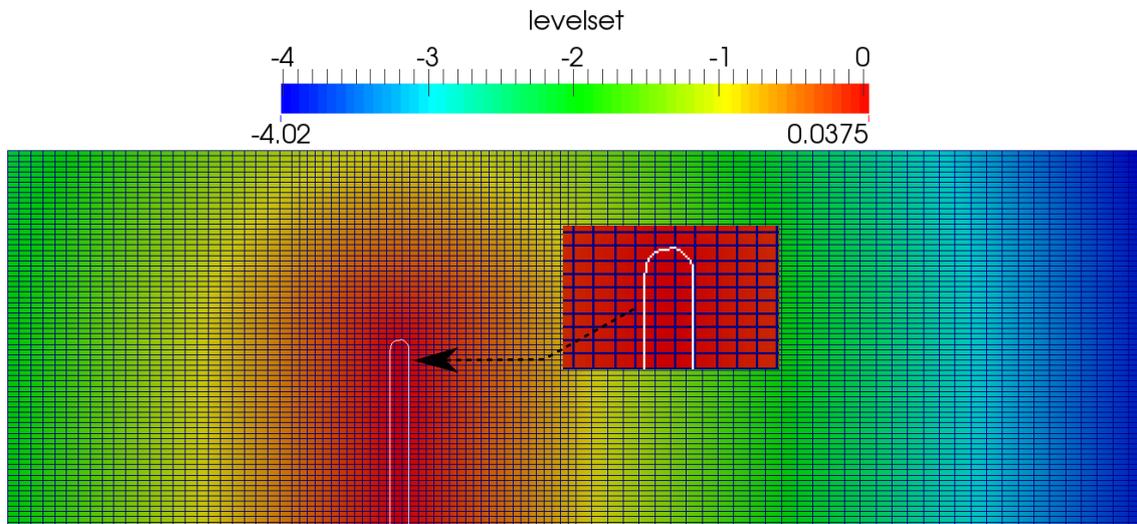


Figure 3-24: Initial levelset field with zero contour in white to depict initial geometry - The color legend shows the signed distance nodal levelset value.

3-5-3 Initial configuration

The geometry description, in this XFEM model, is done with a levelset method and the initialization of the levelset field determines the geometry. The initialization of the levelset field is shown Figure 3-24. The nodal levelset values are shown together with the levelset zero contour in white to depict the undeformed structure. One might have noticed the mesh is divided into three sub-areas with the middle part containing the structure. This middle part has smallest elements, i.e. finest mesh, because the most interesting physics occur around structure and therefore a finer mesh is desired to give more accurate results. The beam is located just left to the middle of the middle area, as it will deform to the right. In ALE-methods for FSI, the body fitted mesh usually is finer and finer near the fluid-structure interface for similar reasons. In XFEM, the freedom to choose the refined area is big, but should be tuned towards the expected physical phenomena.

3-6 Summary

In this research, a 2D FSI problem is modeled, using XFEM and LSM. The combination of XFEM and LSM gives a flexible geometry description, which is beneficial, especially, for topology optimization. Together LSM, XFEM and the governing equations provide the framework to build a numerically consistent Jacobian. Both solution and optimization procedures benefit from a numerically consistent Jacobian and therefore much effort is put into finding it.

The geometry of the system is described by the zero-contour of the LSF and this gives a crisp and physically relevant fluid-structure interface. The Lagrangian structural domain is projected onto the Eulerian fluid domain and a third field is defined to determine the intersections between the two domains. One material domain is considered void in the mesh of the other domain by exploiting the XFEM-feature of turning parts of the mesh off. The

third field has no physical interpretation, it is only used to transfer coupling information from one mesh to the other. It is defined by structural bars, which track the structure to void interface in the Eulerian reference frame. While the structural LSF is constant, the LSF for the fluid is defined as the orthogonal distance from node to fluid to void interface. This interface is determined by the intersections of the structural bars with the fluid element edges. This rather elaborate setup has the advantage that analytical derivatives can be determined, which are needed for the numerically consistent Jacobian. The current implementation of this LSM has some negative consequences regarding mesh discrepancy and unmanageable projection configurations.

In this research, FSI is modeled by the incompressible steady state Navier-Stokes equations for the fluid and a linear elastic material in plane strain for the structure. These governing equations are rewritten into the weak formulation to allow approximations with the Galerkin approach and incorporation of boundary conditions. At the interface between fluid and structure no-slip and traction conditions are applied. Since in XFEM the fluid-structure interface lies within intersected elements, the advanced method developed by Gerstenberger and Wall [2010] is implemented to handle the no-slip condition. To approximate the integrals of the weak formulation, Gaussian quadrature is used.

Due to the introduction of the LSF geometry description, finding the numerically consistent Jacobian is a more complex process, as the fluid LSF depends on the structural displacements. This gives additional terms that are important, especially for the fluid. The fluid residual is a function of the fluid LSF and the state variables, such that the fluid residual derived with respect to structural DOFs can be separated into two terms. The first being the fluid residual with respect to fluid LSF and the second the fluid LSF with respect to the structural displacements, see Eq. (3-85). The first term is computed with Finite Differences and the second is computed analytically using the setup of Section 3-1.

With the Jacobian, the system can be solved to find the steady state solution. Two different solvers are used in this research, which use the Jacobian differently. The staggered setup solves the fluid and structure problem separately and ignores several coupling terms in the Jacobian. The monolithic setup uses the complete Jacobian. The residuals built during both solving processes are equal. Using two different solvers, that use different information, helps to pinpoint potential problems with the model.

The actual physical FSI problem solved in this work is a 2D tunnel in which a fluid flows from left to right. Near the middle of the tunnel a beam is fixed at the bottom. This problem is part of the COMSOL model library and the results from the MATLAB model are compared to the results that COMSOL produces using an ALE method for FSI problems.

Model verification - Staggered setup

This chapter presents the results and analysis of the staggered setup. This staggered setup will provide information on whether the fluid levelset update procedure works correctly, on convergence rates of the separate domains and on the current residual build-up to see whether it is capable of finding a steady state solution. The ultimate goal is to be able to conclude that the residual build-up is done correctly such that the steady state solution found with this XFEM model is similar to that of the COMSOL benchmark. Relating this back to Section 2-3 means finding out whether the function we define is able to describe the solution of the physical problem we investigate. The residuals in both the monolithic and staggered solver include all elemental, boundary and FSI contributions, but the Jacobians differ, as was explained in Section 3-4. Previous experience with the used XFEM model showed that single fluid or structural problems produced correct results. In Section 4-1, the steady state solution of the XFEM model and the COMSOL model are presented. In Section 4-2, some characteristics of the staggered solver are presented. In Section 4-3, analysis of these results is presented.

4-1 Results - Staggered setup

In this section, the steady state solutions from the XFEM model and COMSOL are presented. Some first observations are made, which will be used in the analysis in Section 4-3. The problem setup for these results was presented in Section 3-5. The mesh used for these results is: left 20×79 , middle 60×79 , right 20×79 elements per sub-area.

4-1-1 Steady state solution - XFEM model

At steady state, the displacement field of the structure and the pressure and velocity field of the fluid no longer change. The external force applied to the structure by the fluid is in equilibrium with the structural displacement field times the structural stiffness matrix. This situation is shown in Figure 4-1 to Figure 4-2b with the non-dimensional numbers denoting

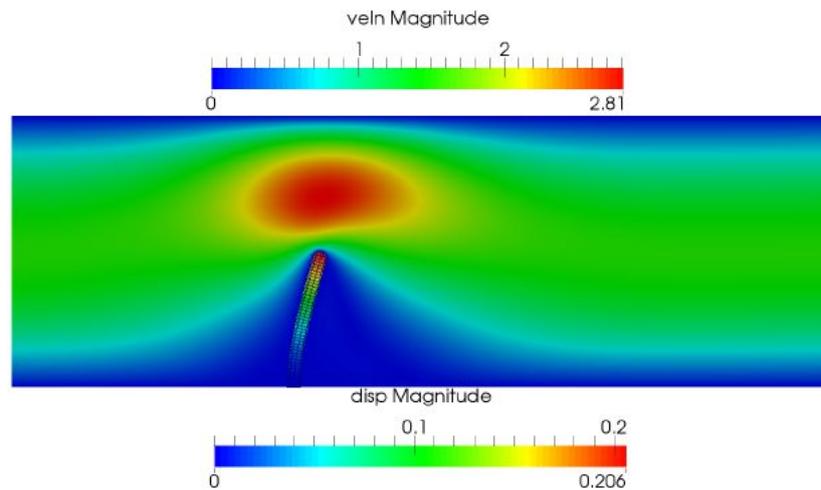
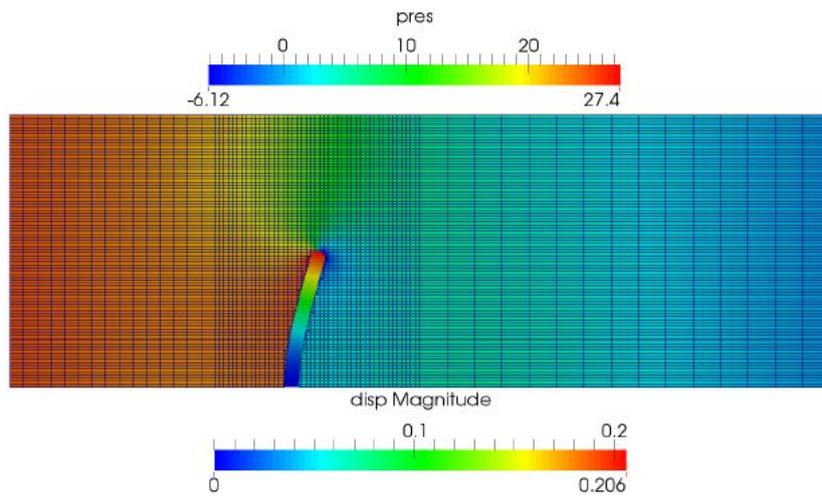


Figure 4-1: Steady state solution XFEM model - Fluid velocity field and structural displacement with structural mesh

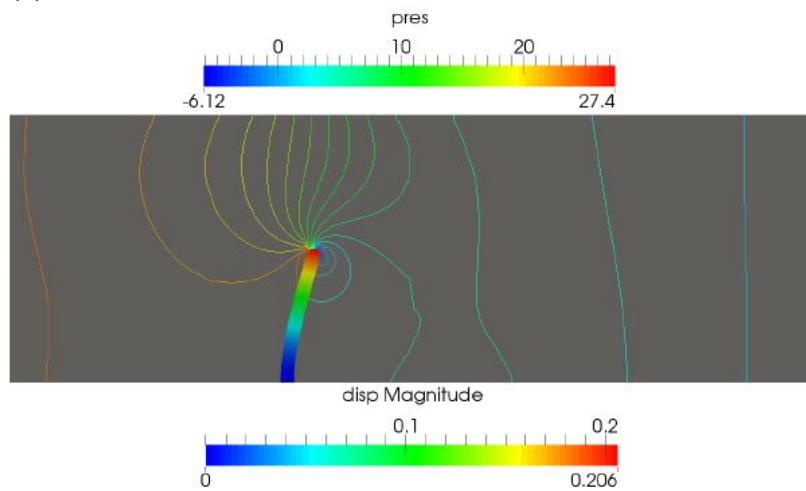
the magnitudes of the nodal values. Figure 4-3 shows the velocity and displacement field separated into x- and y-direction.

Fluid velocity Figure 4-1 shows the deformed structure with the velocity field. The highest velocities are found directly above the top of the beam. The *incompressible* fluid is squeezed through the narrow passage and hence the highest velocities should be found there. This ‘squeezing’ is also nicely illustrated by Figure 4-3b, where left of the structure the fluid moves up (positive velocities in y-direction) and right of the structure the fluid moves down. The undeformed length of the beam was half the height of the domain, such that in the deformed configuration the passage above the beam will be slightly higher than half the height of the domain. This would mean that, if we presume a parabolic flow profile between the top of the beam and the upper wall, the maximum velocity should be less than twice the maximum velocity at the inlet. This is the case, namely 1.5 at the inlet versus 2.81 above the beam. Along the upper and lower wall and the fluid-structure interface, the velocities are zero indicated with the dark blue, according to the no-slip condition. The deformed Lagrangian mesh is also visible within the structure.

Fluid pressure Figure 4-2a shows the deformed structure with the pressure field. In Section 3-2, it was mentioned that pressure essentially acts as a Lagrange Multiplier on incompressible condition, i.e. the pressure gradient acts as a force to prevent volume change of the fluid. This supports the observation that on the left side of the domain the pressures are highest and lowest on the right. The fluid is squeezed through the passage above the beam and afterwards the available flow area increases again. Figure 4-2b shows 20 pressure contour lines, which help to relate the pressure more accurately to the location within the domain, making the comparison with the COMSOL results easier. Within the pressure field the mesh is also visible to illustrate the refinement around the structure.



(a) Fluid pressure field field with fluid mesh and structural displacement



(b) Fluid pressure contours and structural displacement

Figure 4-2: Steady state pressure field XFEM model

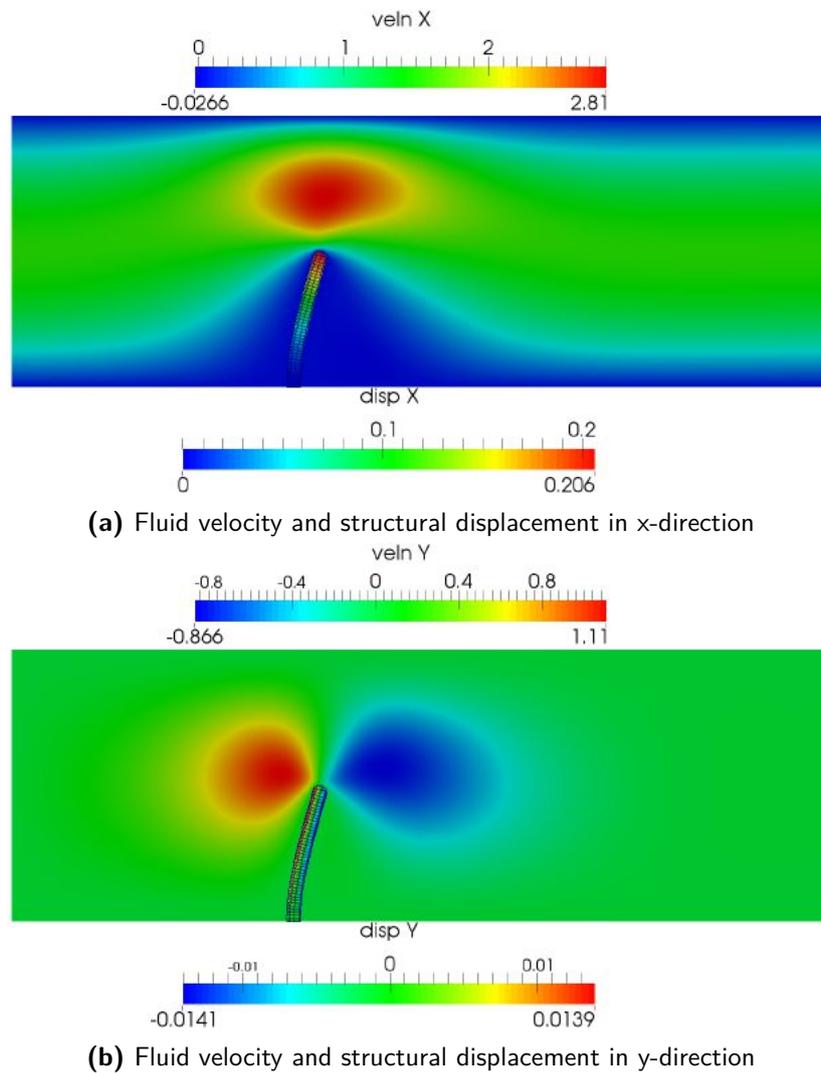


Figure 4-3: Steady state velocity and displacement field in x- and y-direction XFEM model

Structural displacements Both Figure 4-1 and Figure 4-2a show the deformed structure and the colors indicate the magnitude of the displacement. The biggest displacements occur at the top of the beam with a maximum of $0.206 \times 50 \approx 10[\mu\text{m}]$. This displacement is rather big compared to the length of the structure, indicating that the assumption of small deformations is no longer valid. Physically, these results do not have any relevance, since a linear elastic material was used, but *this highlights the ability of XFEM model to cope with large displacements*. In future research, a non-linear material model may be implemented to improve the physical relevance of the results.

4-1-2 Steady state solution - COMSOL benchmark

In this section, the results of the COMSOL simulation of the problem setup from Section 3-5 are presented. As mentioned, this particular problem is part of the COMSOL model library. Take note that the default outlet condition in COMSOL includes a traction-free outlet

condition, which should be changed to match Section 3-5. In Figure 4-4a, the magnitude of the velocity field is plotted together with the deformed structure. Figure 4-4b shows 20 pressure contours and the extremes of the displacements. Figure 4-5 show the velocity and displacement field separated in x- and y-direction.

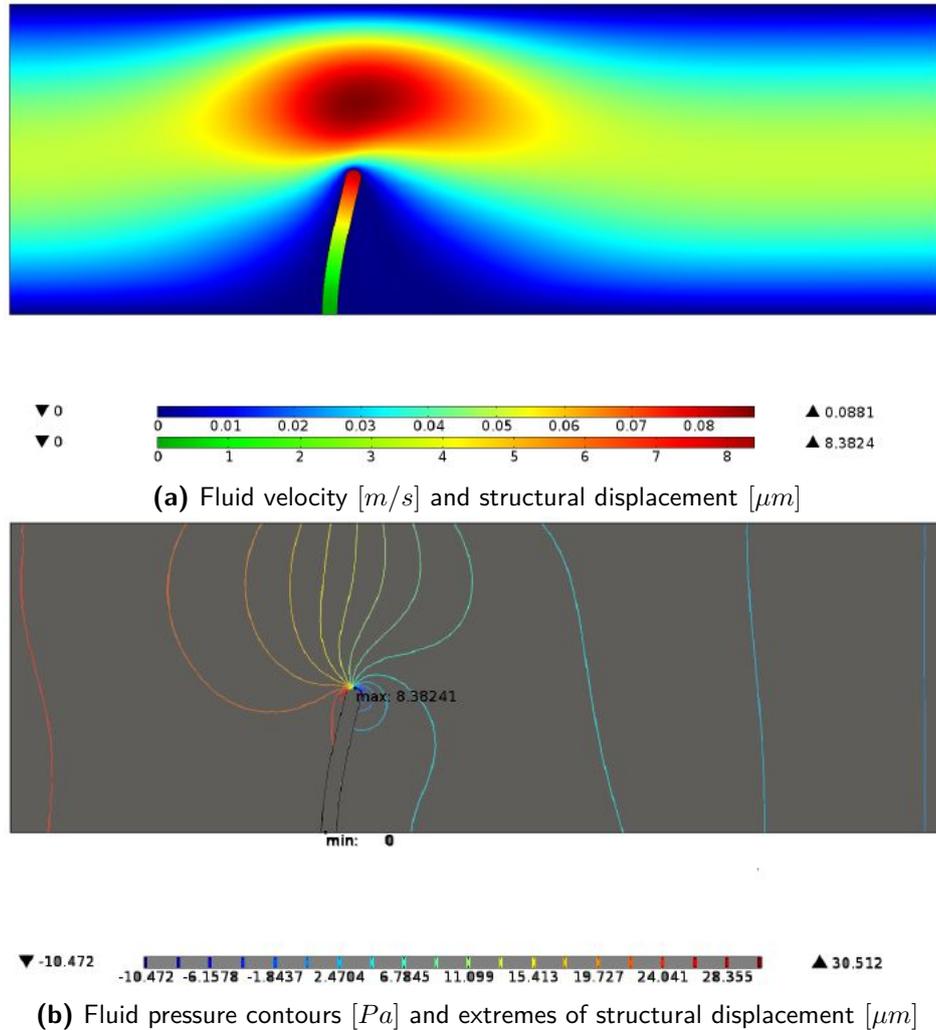


Figure 4-4: Steady state solutions - COMSOL

4-2 Solver studies

This section presents some studies on the characteristics of the staggered solver. These studies have been performed on a different problem setup than presented in Section 3-5¹. However,

¹The differences with the problem setup in Section 3-5 are 1) a plane stress model is used for the structure in contrast to a plane strain model 2) the mean inlet velocity at the left boundary was 22.2 [mm/s] instead of 33.3 [mm/s] 3) the Poisson's ratio is 0.3 instead of 0.33 and 4) no outlet condition instead of zero pressure condition

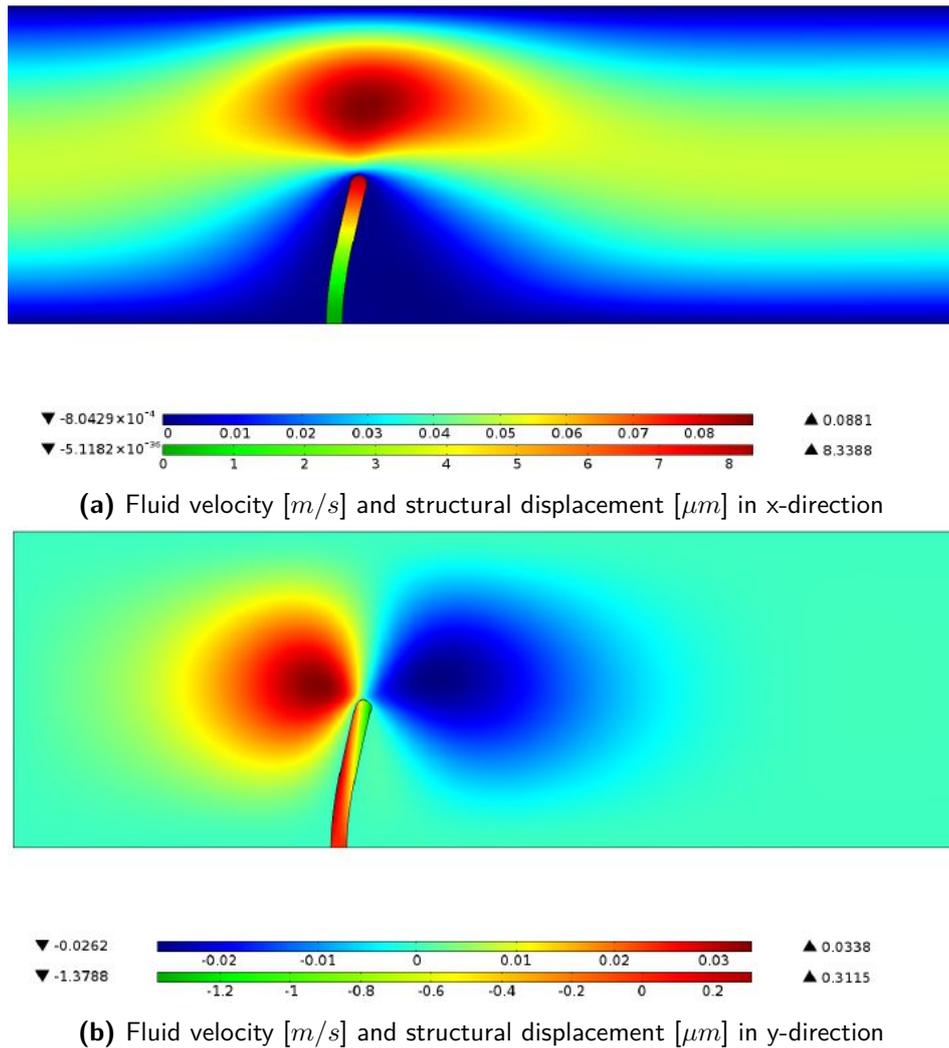


Figure 4-5: Steady state velocity and displacement field in x- and y-direction - COMSOL

the quantitative information that these results provide will also apply to modeling of the actual problem setup.

4-2-1 Solver stages

One of the key features of this staggered setup is that the fluid and structure are solved separately. A couple of intermediate solutions are presented in Figure 4-6. These figures help to understand the residual development presented in Section 4-2-2. The consequences of a staggered setup are clearly visible in Figure 4-6. The figures show the solution after both fluid and structure have gone through an update stage an equal number of times. Since the fluid is updated first, the fluid levelset has not been updated towards the new displacements yet, such that a big discrepancy between the meshes occurs, which is illustrated by the white void areas in Figure 4-6c and Figure 4-6d. These intermediate results have absolutely no physical relevance, only the fully converged solution corresponds to the physics of the system. The

fact that the void area is visible also illustrates the crisp material to void interfaces. There are no intermediate density elements present around the interface. Even though Figure 4-6e and Figure 4-6f show solutions that are not fully converged yet, one can see that this discrepancy reduces when getting closer to the steady state solution. Essentially, as the updates of the structure approach zero, the fluid levelset takes its final form.

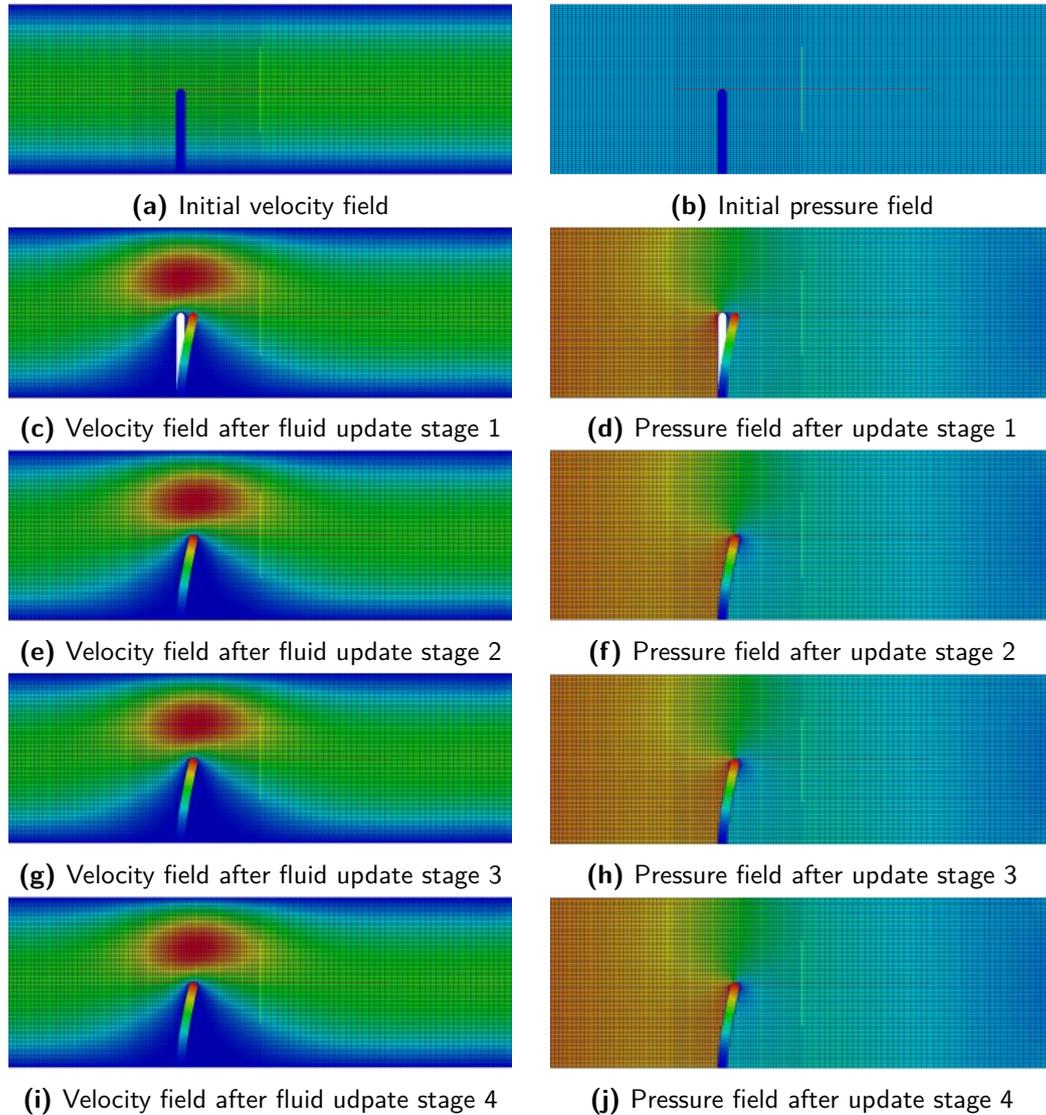
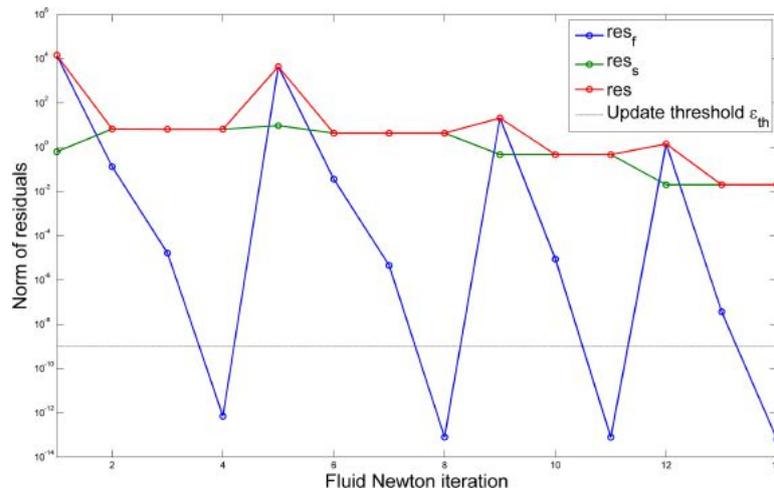


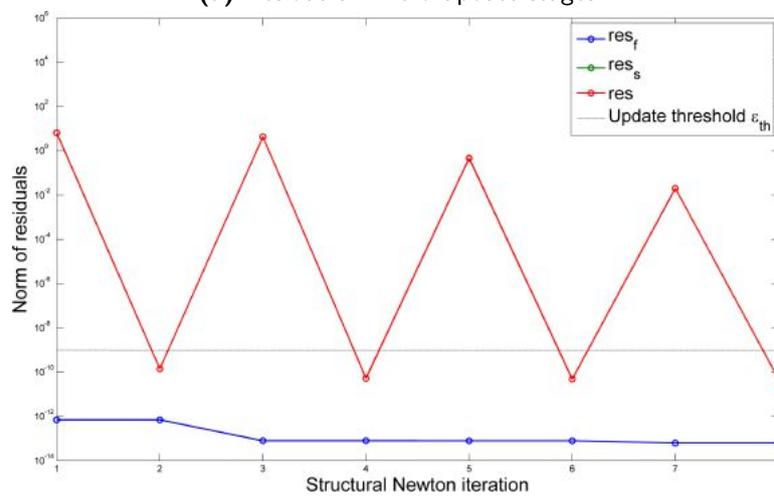
Figure 4-6: A typical staggered solving process

4-2-2 Convergence plots

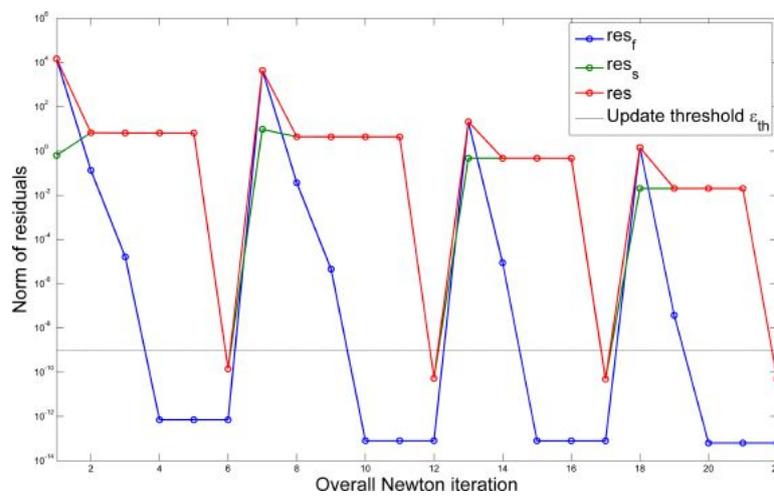
In addition to the visualization of the solving process in Figure 4-6, the residual development has also been plotted in Figure 4-7. These residual plots give insight into the convergence rate of the solver and how the rate is affected by the staggered setup. Below some comments are presented to highlight the most interesting information in the plots.



(a) Residuals in fluid update stages



(b) Residuals in structural update stages



(c) Residuals chronologically

Figure 4-7: Convergence plots of the staggered solve process

Figure 4-7a and Figure 4-7b show the residual development within the update stages per phase. In Figure 4-7c, the same information as in Figure 4-7a and Figure 4-7b is combined to show the process in chronological order until convergence of the complete system is reached according to Alg. 1 in Appendix C.

The update stages for the fluid are characterized by almost straight lines until big jumps in residual occur, see Figure 4-7a. These jumps occur because at the start of each fluid stage the LSF is updated according to the new structural displacement field and hence new elements are intersected making new DOFs part of the solution. These new DOFs may not have been updated yet and still hold an old solution. More details on this phenomenon can be found in Section 4-3-2. The straight lines before the jumps indicate near quadratic convergence, i.e. suggesting a numerically consistent Jacobian, for a single fluid problem with constant LSF/geometry. Due to the linear elastic material model, only one update per structural update stage is needed, see Figure 4-7b. This zigzagging of the structural residual is the result of an updated external force applied by the fluid to the structure.

Both Figure 4-7a and Figure 4-7b show that the total residual is dominated by the phase that is not updated. During the fluid update stages, the traction applied to the structure changes and, hence, one can see a change in structural residual during the fluid update stages. The other way around is different. In the structural update stages, the fluid LSF is not updated, the discrepancy between the meshes stays the same, the structural velocities are always zero and, hence, the interface no-slip condition and elemental contributions to the fluid residual do not change.

The update threshold line plotted in these figures is used to determine whether an individual phase is converged or not. If the residual lies below this line, the update is skipped and the next update stage of the other phase starts. Figure 4-7c shows 4 instances where all three residuals lie below this threshold. However, only the last instance has a physically relevant solution, because at that point the two meshes show the least discrepancy, i.e. the fluid LSF has also reached steady state. This is the consequence of the fact that structural displacement field no longer changes, which is shown in Figure 4-8. This plot is related to the earlier remark on making the two phases ‘communicate’ even though they are solved separately. This figure shows the norm of the update vector for both phases. From this figure it can also be concluded that it took the process 14 Newton-Raphson updates and 4 update stages per phase to reach steady state.

4-2-3 Parameter studies

To improve results and convergence some basic parameter studies have been done. It may be noted that many more studies can and should be performed, and that the results of many of these studies, such as the under-relaxation, *heavily* depend on the coarseness of the mesh. Nonetheless some quantitative conclusions can be drawn from these studies, which may be useful for future research.

Mesh refinement

If a FEM model behaves correctly, a finer and finer mesh should, in theory, reach the analytical solution of the partial differential equations. The other way around the coarser your mesh, the

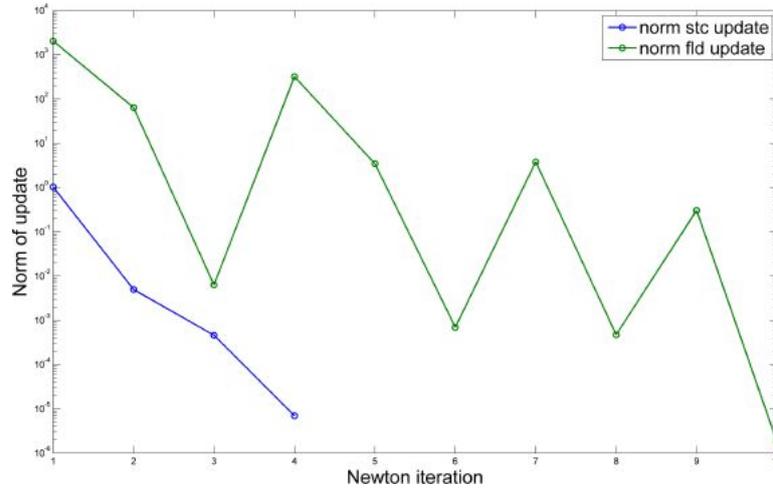


Figure 4-8: Norm of the single phase update vectors - The fluid update vector contains velocity and pressure terms and the structural update vector only displacements

Element area	No. of elements	Max. displacement
1.157e-3	13824	0.1457
8.424e-4	21360	0.1561
5.192e-4	30816	0.1561
3.809e-4	42000	0.1561

Table 4-1: Results of mesh refinement

less accurate your results. There is always a trade-off between accuracy and computational effort, as more elements take more computational effort. Therefore, a small mesh refinement study has been done to verify that the steady state solution converges towards a particular solution with finer and finer meshes. Table 4-1 shows results indicating that the XFEM model works correctly, namely if smaller and smaller elements are used, the maximum displacement in steady state does not change anymore. All results in Section 4-2-2 and Section 4-2-3 are based on the second mesh in Table 4-1².

Under-relaxation

As mentioned earlier in Section 3-4-2, a type of underrelaxation for the staggered solver has been used in this study. After each structural update stage, we choose to update the structural solution with a fixed percentage of that update, denoted by the factor α_s . This smooths the solution towards steady state. Moreover, it influences the convergence of the problem quite strongly. Table 4-2 illustrates this influence. The less update stages are needed, the less time the solver needs and, hence, the results in Section 4-2-2 are based on the underrelaxation parameter α_s of 0.95. This 0.95 is not a coincidence, namely it is approximately the ratio between the maximum displacement at steady state (0.1561) and the maximum displacement of the first non-underrelaxed, overshooting solution for the structure (0.1653). This 0.95 factor

²The mesh used for these results is: left 24×72 , middle 36×72 , right 36×72 elements per sub-area.

α_s	No. of update stages per phase
0.8	7
0.9	5
0.95	4
1	5

Table 4-2: Under-relaxion parameter α_s for the structure

basically forces the structure straight towards the steady state solution after one update. In steady state, the structure is not updated anymore, such that the fluid levelset immediately catches up with the structural displacements and, hence, convergence is reached way quicker. It is important to stress that the 0.95 factor is fully dependent on a converged fluid solution for its initial geometry. If one chooses to reduce the amount of update steps per stage, this parameter α_s will most certainly change. *This analysis suggests that the computational effort of this staggered setup is strongly dominated by the fluid levelset field (LSF) updates*, in the sense that the LSM determines the amount of stages needed. Unfortunately this fixed under-relaxation parameter does not guarantee the best performance of the solver for other geometries. *However, it can be concluded that, if the fluid is converged during the first update stage and the steady state solution of the structure is expected to experience smaller traction forces than the undeformed shape, that this under-relaxation parameter should always be smaller than 1.*

4-3 Analysis - Staggered setup

This section presents analysis of the results presented in the previous sections. Firstly, the results of the COMSOL and XFEM simulation are compared to benchmark the XFEM model for FSI problems. Secondly, a quantitative analysis on convergence of the XFEM model is done.

4-3-1 COMSOL comparison

The coloring of Figure 4-1 to Figure 4-2a and Figure 4-4 suggest that the results are reasonably similar at first sight. The velocity and pressure field show similar patterns and the deformed structures almost have the same shape. In Table 4-3, the extreme values of all the state variables are presented.

Table 4-3 shows that most numbers are of the same order. The biggest differences are found in rows starting with a red colored state variable. The minimum pressure of the XFEM model is about 35% lower than the COMSOL model. This is a rather significant difference and this should be investigated in future research. A possible lead can be the non-smooth pressure contour to the right of the structure in Figure 4-2b. At this pressure contour unexpected behavior is clearly visible. Taking a closer look at Figure 4-2b shows that all contours that cross an edge between the refined mesh and a coarser mesh show an unexpected course.

The displacement field also shows differences. Both x- and y-displacements extremes are off in magnitude. Furthermore, the distribution of y-displacements in Figure 4-3 and Figure 4-5

State variable	XFEM	COMSOL
$v_{f,x}^{max}$ [m/s]	0.0936	0.0881
$v_{f,y}^{max}$ [m/s]	0.0370	0.0338
$v_{f,x}^{min}$ [m/s]	-8.8578e-4	-8.0429e-4
$v_{f,y}^{min}$ [m/s]	-0.0288	-0.0262
p^{max} [Pa]	30.384	30.512
p^{min} [Pa]	-6.786	-10.472
$u_{s,x}^{max}$ [μm]	10.300	8.3388
$u_{s,y}^{max}$ [μm]	0.695	0.3115
$u_{s,x}^{min}$ [μm]	0	0
$u_{s,y}^{min}$ [μm]	-0.705	-1.3788

Table 4-3: The extreme values of the steady state solutions - The XFEM-results are dimensionalized according to App. A

is very different. It may also be noted that the lack of displacement in y-direction could be related to the abundance in x-displacement. Again future research should be performed to find out what the reason for this difference is.

This comparison is not enough to fully validate the XFEM model. It, however, gives enough confidence to continue with this XFEM model. The results show that the residuals built in this model allow to numerically find a steady state solution, which resembles the steady state solution found with COMSOL. Future research should quantitatively point out the reasons, that explain these differences. Below a list with clear differences between the models is presented that will be part of the explanation for the differences:

- Element types - The XFEM model uses quadrilaterals for the non-intersected elements, while COMSOL uses triangular elements throughout the domain;
- Mesh/Discretization - The COMSOL model uses a body fitted mesh, while the XFEM model always contains discrepancy between the two meshes, see Section 3-1-6. Additionally, the nodes at which the extremes occur may not have the same Eulerian coordinates, due to different discretizations;
- FE interpolation - Both fluid problems are solved using linear FE interpolation, but the COMSOL model uses quadratic interpolation and the XFEM model linear interpolation for the displacements. Linear interpolation results in stiffer structures, but oddly enough Table 4-3 shows opposite results as the displacements in x-direction are bigger for the XFEM model;
- Loss of traction information - The current setup is unable to handle structural bars that are completely submersed within a fluid element (no intersections), which means a loss

of physical information, see Section 3-1-6. On these structural bars no external force is applied even though physically there should be.

4-3-2 Convergence in XFEM models with changing DOFs

This section presents an in-depth analysis on convergence of the staggered Newton-Raphson solver in XFEM. It is a general analysis and can be applied to every XFEM-based problem with changing DOFs during the solving process. The changing of the DOFs is a consequence of a changing levelset field on an Eulerian mesh. This is inherent to the methods used in this research and one has to realize the consequences on both the convergence criteria and the convergence rate. Another problem related to changing DOFs was that of the levelset field consistency mentioned in Section 3-1-5. Hence the motivation to implement the flood-fill algorithm has similar grounds as the analysis found below. Both are related to the changing DOFs.

Convergence criteria

In XFEM, a wide variety of convergence criteria can be used. The convergence criteria, in this study, are specifically setup towards the goal of the simulation, namely a converged steady state solution. At this converged steady state solution we know that the residual and the solution update should be close to zero. Furthermore, we expect to see a big drop in residual comparing the final state to one of the first states, see also App. F. A common measure to check whether vectors are close to zero, without inspecting each individual vector entry, is the 2-norm. For two or three entry vectors this norm gives the Euclidian distance. For a vector \mathbf{x} with n entries the 2-norm is defined as follows:

$$\|\mathbf{x}\|_2 = \sqrt{\mathbf{x} \cdot \mathbf{x}} = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2} \quad . \quad (4-1)$$

The norm of the residual vector \mathbf{R} can be used as an indication of how well the FEM-approximation satisfies the solution of the partial differential equations in the weak form. If we compare the residual of a certain updated solution with a residual from an earlier solution, by taking the ratio between the two, one can track how much the solution has improved. These relative residuals are independent of the number of entries in the residual vector and therefore often used in FEM. However, in the XFEM model of this research, the number of entries in the residual varies with the number of intersected fluid elements. Essentially, after each levelset update a different mesh is used, which makes the comparison between residuals more difficult. The dependence on the number of entries has crawled back into the criteria. This can be explained as follows: Suppose we have two residual vectors, \mathbf{R}^{ref} with a entries and \mathbf{R}^n with b entries, and we assume that all entries in both vectors are ε , the relative residual can be written as follows:

$$\frac{\|\mathbf{R}^n\|}{\|\mathbf{R}^{ref}\|} = \frac{\sqrt{b \cdot \varepsilon^2}}{\sqrt{a \cdot \varepsilon^2}} = \sqrt{\frac{b}{a}} \varepsilon \quad . \quad (4-2)$$

Eq. (4-2) shows that, if, for some unlikely reason, the entries of the residual vectors remain equal to ε , the relative residual can still reduce, if the levelset updates reduces the number of entries in b , i.e. $b < a$. In fluid flow analysis, it is common to take the residual after the first

update as a reference residual, since the initial solution usually has no physical relevance to the system. The initial solution in our setup is not physically relevant and, hence, while the solution is updated, the number of DOFS changes. In this XFEM model, fluid DOFS change during the solving process, because fluid levelset field changes. Literature has not presented a standard method to deal with this yet.

The effect of the changing number of entries has not been accounted for in this research. We just say that, if the residual is small enough, it will represent the physics. The reference residuals used in App. F are the residuals built after the first Newton-Raphson update of fluid. This solution gives the first reasonable solution for the fluid and holds a structure that has not yet been deformed.

Convergence rate in XFEM

The previous section explains the consequences of the changing DOFS for the relative residual, but there is more to it. Changing free DOFS also imply a changing solution update related to these new DOFS. This means that the update does not build upon the solution from the previous Newton-Raphson iteration. In a classical Newton-Raphson solving process, one adds the update to the previous solution until convergence, where each update is related to the same DOFS. With changing DOFS a classical Newton-Raphson update scheme is impossible. Essentially, after each levelset update a ‘new’ FEM problem with a different initialization is solved. *Continuing with this line of thought and relating it to the quest for finding the numerically consistent Jacobian, one can philosophize that quadratic convergence, even with using a numerically consistent Jacobian in a system with changing DOFS, is impossible*³.

In order to understand the consequences of this characteristic of the changing DOFS in XFEM, it is useful to visualize this change. In XFEM terminology, the DOFS that are part the solution are called *active DOFS* and the levelset field determines which DOFS are part of the solution, see Figure 4-9. Figure 4-9a shows the active DOFS, related to the intersected fluid elements and depicted by the red squares, in the undeformed configuration. These nodes together with the standard nodes from the rest of the fluid domain (white area) are used to build the residual and Jacobian to produce a solution update. If a solution update contains displacements, which are big compared to the element size, the fluid to void configuration might look like what is shown in Figure 4-9b. Here the pink fluid to void interface has changed significantly, due to the fluid levelset update and, hence, to active DOFS related to both intersected and normal fluid elements have changed. The colored areas relate Figure 4-9a to Figure 4-9b and mean the following:

- Green - Intersected elements in undeformed configuration.
- Light red - ‘Semi-active’ elements, which contain both active and non-active nodes.
- Light blue - Strictly void elements, i.e. all nodes are non-active and or not part of the fluid solution.

³In numerical modeling perfect quadratic convergence is hard to achieve, due to round-off errors and the like. However, these numerical artifacts exist both for FEM and XFEM. The effects of changing DOFS is a whole different problem

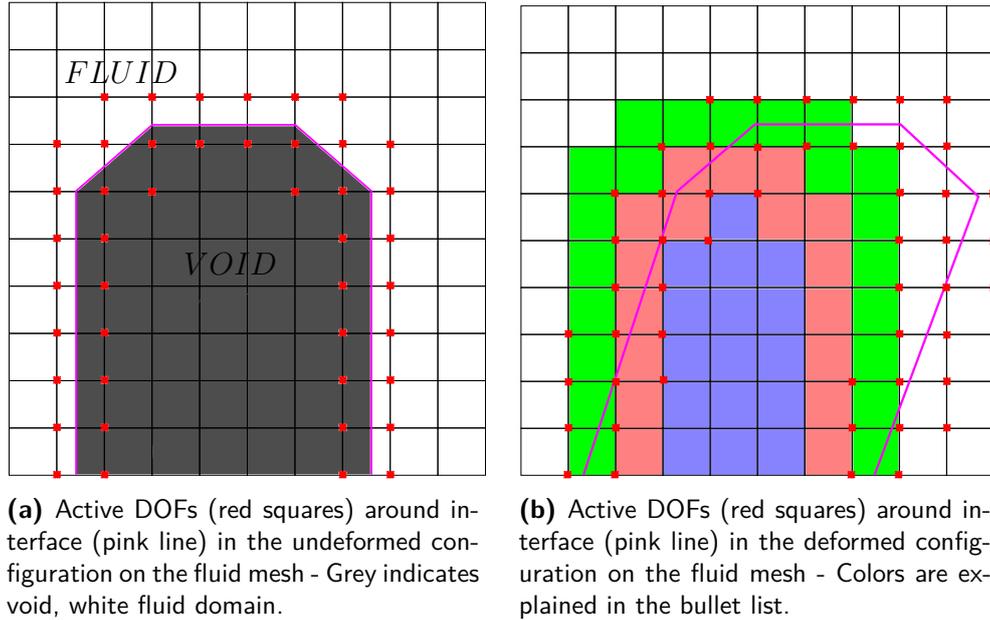


Figure 4-9: Changed fluid DOFs after a fluid LSF update - Both the number and individual nodes changed.

The location of the active nodes with respect to these areas in the deformed configuration lies at the bottom of the argument against quadratic convergence, in models with changing DOFs. Take, for instance, the inactive nodes, which are connected to light blue elements. These light blue elements were not part of the active domain in Figure 4-9a and hence have not experienced an update. Suppose we regard the first Newton-Raphson update, this means that the nodes in the light blue still hold the values of the initialization, when their respective elements are intersected after a fluid LSF update. These initialization values are then used to build the next update upon. Because these elements ‘do not know about the update’, they will most likely have negative effect on the residual in the connected elements⁴. The elements in the light red area are partly updated, so a similar analysis can be done, but the effects on the residual from these elements are expected to be less significant. One can also argue that even the elements that are active in both configurations will negatively influence the convergence. For instance, the elements in the green area were intersected in the undeformed configuration and, hence, the no-slip condition forces the velocities to zero in that element. In the deformed configuration these elements are no longer intersected and the further they are away from the new interface, the higher the fluid velocities will be for the converged solution. However, when the residual is built the element still contains zero velocities. The total negative effect is illustrated nicely in Figure 4-7a by the big jumps in fluid residual after an update stage.

⁴A metaphor to understand the effect of the changing DOFs is that of a play in development. A director (the solver) has more actors (the nodes) available than he can use on stage (the active domain). The rest of the actors are back-stage (the non-active domain) waiting to be called upon the stage. During practice, the director decides continuously to change and improve (the update) the script, based on what he sees (the solution). He calls 10 actors on stage and sends 3 actors backstage (the switching between active and non-active domain). The actors on stage that have been present for a while are aware of the changes in the script, but the director ‘forgets’ to brief the new actors on the changes in the script. The new actors act their part, however, since they are unaware of the changes in the script, they negatively influence the momentum of the scene.

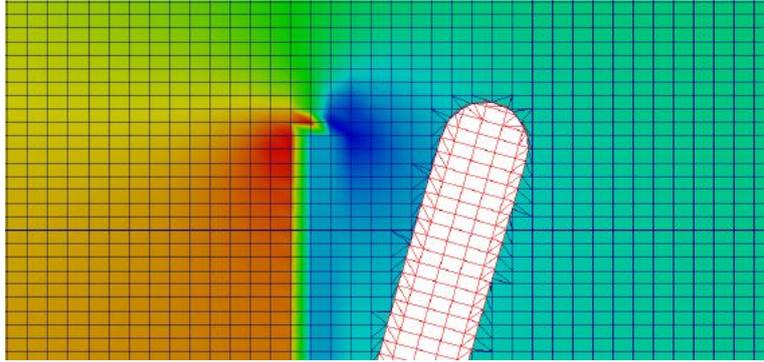


Figure 4-10: The pressure field plotted right after the fluid levelset update in the second fluid update stage - A weird solution is visible in the area where the undeformed structure was located. Furthermore the pressure field has not adapted yet to the new configuration.

Figure 4-10 confirms the previous analysis as a weird solution is visible right after the fluid levelset update. The same areas as in Figure 4-9 are (partly) visible in Figure 4-10; the green rectangle corresponds to semi-active elements in the initial configuration and the light blue corresponds to the inactive elements. The pressure field extremes are still situated around the area of the undeformed structure. This weird solution negatively affects the convergence rate of the solver and causes the big jump in fluid residual. As the solver progresses and the solution gets closer to the final steady state solution, the DOFs do not change anymore and from that point on, quadratic convergence may be expected. Unfortunately the benefit of quadratic convergence is less significant in the final stages of the process. If displacements are small compared to the element size (either coarse mesh or stiff structure), such that the fluid levelset updates remain small and the fluid to void interface stays within the same intersected elements, the negative effect will be minimized.

Based on this qualitative analysis, it can be concluded that the changing of the DOFs influences the behavior of the solving process. The extent of this influence on the monolithic setup is yet to be determined quantitatively. This analysis leads to very interesting research possibilities for the future. Updating the non-active domain might enhance numerical stability and convergence of the solving process. Setting nodal values of the non-active nodes to the average of the active nodal values of neighboring elements, for instance, sends the solution in the non-active domain in similar direction as the update of the active domain. Another option would be to prevent changing of the DOFs with an ALE approach, which was developed in Gerstenberger and Wall [2008b].

4-4 Summary

In this chapter, the results from the XFEM model, using the staggered setup, and the COMSOL model are presented. The results of the XFEM model were produced using optimal settings based on some basic parameter studies. The staggered solving process is characterized by the separated updates, which is clearly visible in both the intermediate solutions in Figure 4-6 and the associated residual development in Figure 4-7. The most interesting observations are the big jumps in the residual plots, which are the consequences of a change in DOFs, due to the fluid LSF update.

Further analysis of the results of both models shows that the results are moderately close to each other. For the continuation of this research the MATLAB results are *sufficient*, i.e. the staggered setup is able to find a steady state solution, based on the implemented residual build-up. At first sight the results look similar, but a more quantitative analysis shows interesting differences, regarding the structural displacements and pressure contours. Future research should be performed to determine what can explain these differences and more elaborate validation of the XFEM model is needed.

The fact that in the XFEM model the fluid DOFs change, due to a changing fluid LSF, has some rather interesting consequences. Firstly, the number of entries in the residual vector changes. If one uses relative residuals as convergence criteria, one has to realize that the norms of the residual vector use a different number of entries. Even more interesting is the effect of the changing DOFs on the convergence rate. In the staggered setup, this manifests itself by big jumps in residuals and a qualitative analysis predicts that also the monolithic setup will be affected. Even with a numerically consistent Jacobian the desired quadratic convergence of the monolithic solver is impossible, since the consecutive updates do not build upon the solution from the previous update iteration. After each LSF update a new FSI problem with a different initialization is solved. Since XFEM turns of particular elements, these elements will not be updated, even though the rest of the domain is updated. As soon these ‘forgotten’ elements become part of the solution, they still hold an old solution and therefore influence the convergence rate negatively. Future research should be performed to verify and quantify the effects on the monolithic convergence rate and to see whether methods need to be developed and implemented to improve the convergence.

Model verification - Monolithic setup

The previous chapter shows us that the XFEM model is capable of finding a steady state solution with the current residual build-up. The next step then is to check whether the complete Jacobian is numerically consistent. In the monolithic setup, all Jacobian entries Eq. (3-87) are used in the Newton-Raphson solver, in contrast to the staggered setup. Since this model was set up because of its flexible geometry description and the sensitivity-analysis requires a complete Jacobian, investigation on behavior and characteristics of the monolithic setup is valuable. Below the results of a number of tests are shown, which provide information on the monolithic setup. In Section 5-1, exploded solutions of the monolithic setup are presented. Section 5-2 present results used to answer the question why the monolithic setup is unable to find the steady state solution in Section 5-3.

5-1 Results - Monolithic setup

The results of the monolithic solving process are presented in Figure 5-1. These figures show the solution just before the solver broke down. The solution exploded and hence the code was not able to continue anymore. The results show that both the velocity and the pressure field are approximately zero in the entire domain. This is not a correct solution for steady state. The question now is ‘*Why did the solution explode?*’. As mentioned in Section 3-4-1 Newton-Raphson solvers sometimes need some tuning to be able to robustly find the correct solution. However, even the tuned solvers require that the solver ingredients are correct to produce a relevant solution, i.e. the residual and Jacobian should be correct. As the staggered solver proved to be able to find the steady state solution similar to COMSOL, the Jacobian should be investigated to check that it is numerically consistent.

5-2 Jacobian - 2 element Finite Differences

As is shown in Section 5-1, the monolithic setup does not work. This suggests that the Jacobian sends the solving process in the wrong direction. The construction of the numerically

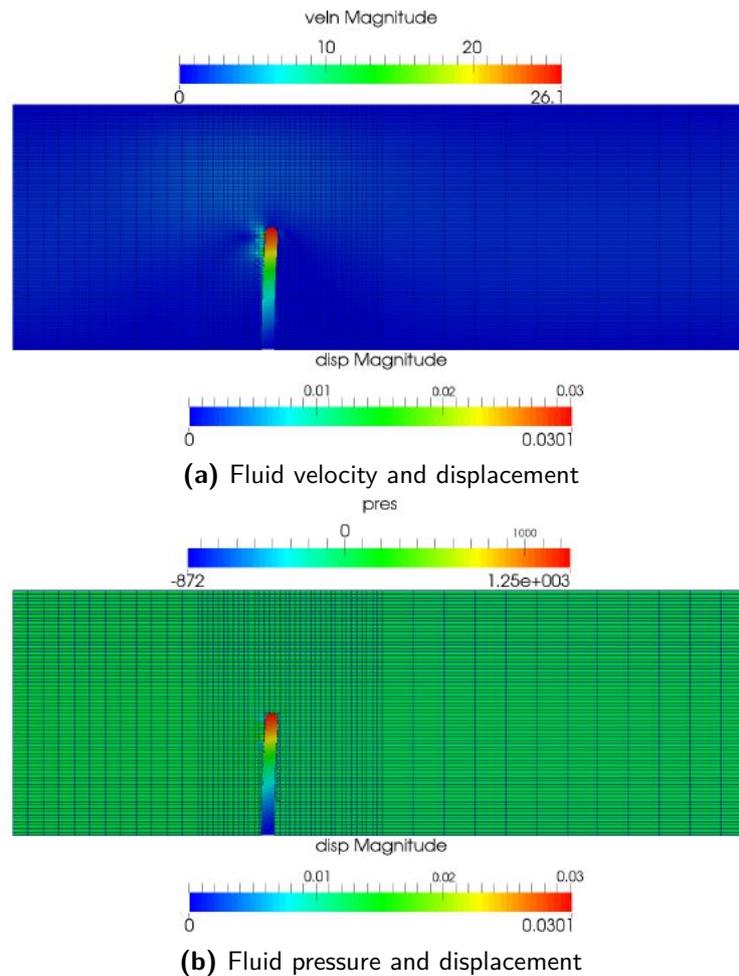
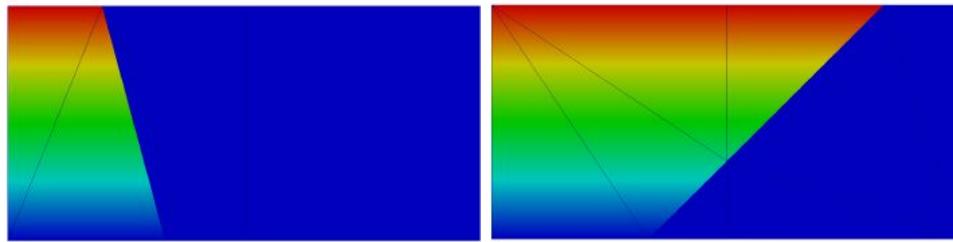


Figure 5-1: The exploded solution of the monolithic solver

consistent Jacobian in this XFEM model is complex, both conceptually and from the point of view of implementation in MATLAB. Code-bugs can occur and sometimes are very difficult to find. Reducing the complexity of the problems and designing simple numerical tests help providing insight into where errors occur. Finite differences, for instance, can be used to check derivatives. It is a brute force method to check whether the Jacobian of the system is numerically consistent with the residual. Within some tolerance range, the analytical Jacobian and FD Jacobian should be equal. This FD approach is only able to give insight into whether the Jacobian is consistent with the residual, and cannot say anything about the residual itself. That is why a staggered setup has also been used. Choosing the step size for the FD is not easy and hence a sweep has been done to determine the influence of the step size. Below the results for two different configurations are presented and discussed as different configurations illustrate and/or highlight different problems. Both problems contain fluid on the left and structure on the right. For the fluid a linearly increasing velocity field and for the structure a uniformly zero displacement field is shown. The pressure field is uniformly non-zero. The big difference between the two configurations is that in the left problem only the left element is intersected, while in the right problem both elements are intersected and the fluid-structure interface crosses the shared element edge.



(a) Two elements with element 1 on the left intersected (b) Two elements with both element 1 and 2 intersected

Figure 5-2: Simplified FSI problems - Finite difference configurations with two elements

5-2-1 The finite differences process

Computing the Jacobian with central finite differences is computationally expensive, but usually provides better results than forward or backward FD, see Section 2-4-1. For each perturbed DOF the complete residual is built twice, and in systems with many DOFs that is computationally expensive. This motivates the choice to work with simple problems with few DOFs. The simple problems in Figure 5-2 have no physical relevance, but do give great insight into the numerical and mathematical process behind the complete model. As mentioned, using the correct step size is crucial. A step size that is too big gives wrong derivative information and a step size that is too small leads to numerical problems related to round-off errors and tolerances used within the model. In this work, a sweep over different step sizes is done (detailed results omitted) to find a suitable step size of $1e-8$. The finite difference algorithm can be found in Alg. 3 in Appendix C.

5-2-2 Methods to compare Jacobians

To compare the analytic and FD Jacobian, one has several options. In this work, a combination of methods has been used to compare the Jacobians on both global and individual entry level. The methods used are the following and the results are presented in Section 5-3:

- Spy on location of entries - The spy function in Matlab gives insight in the locations of non-zero entries of sparse matrices; nz at the bottom of the plots indicates the number of non-zero entries¹.
- Frobenius norm on absolute numbers - The Frobenius norm is defined as the square root of the sum of all entries squared and is a measure to check the absolute values of all entries on a global level.
- Signs on entry pairs - Each entry in a Jacobian has an associated entry in the other Jacobian, corresponding to the same state variables. The signs of these associated entries should be equal. If one multiplies the signs of each entry entry-wise with the entries of the other Jacobian and both entries of the entry pair have the same sign, the result is 1.

¹Note that non-zero entries may include Inf and NaN

Note that one has a lot of freedom in methods to compare Jacobians, so the method should be adjusted to produce the desired information. Different methods may lead to different insights.

5-3 Finite Differences - Results

In this section, the results of the finite differences and the analytic Jacobian for the two setups are given and the most interesting observations are presented. The reader is warned at forehand that only the structural displacements in x-direction have been perturbed (see Section 5-3-4 for an explanation). First the results of the configuration in Figure 5-2a are presented, then the configuration in Figure 5-2b. These results and the consequent analysis should be inspiration for future research.

5-3-1 Two elements with only left element intersected

The most relevant results for the one intersection case are presented in Figure 5-3 and Figure 5-4. Figure 5-3 shows that the location of the entries and the Frobenius norm are good. These

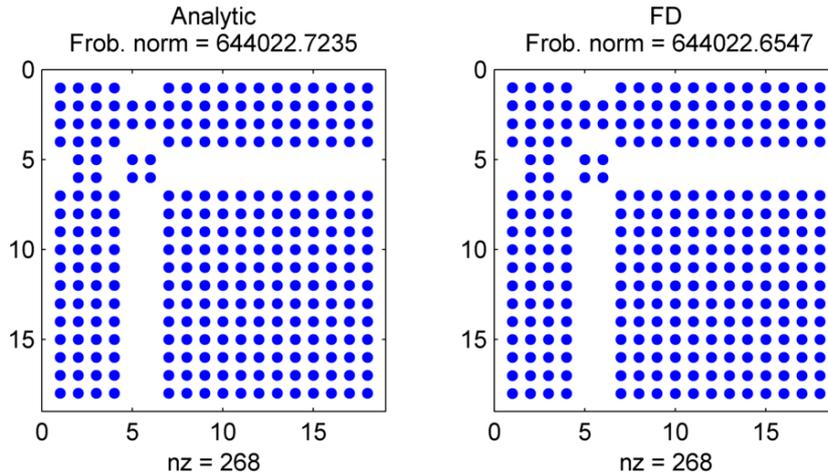


Figure 5-3: Spy plots and Frobenius norm of the complete analytic and FD Jacobian for two elements problem with left element intersected

results suggest that the analytic Jacobian is consistent with the residual. We label the entries in Eq. (3-87) as follows:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathbf{R}_s}{\partial \mathbf{u}_s} & \frac{d\mathbf{R}_s}{d\mathbf{u}_f} \\ \frac{\partial \mathbf{R}_f}{\partial \phi_f} \frac{d\phi_f}{d\mathbf{u}_s} & \frac{\partial \mathbf{R}_f}{\partial \mathbf{u}_s} + \frac{d\mathbf{R}_f}{d\mathbf{u}_f} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{ss} & \mathbf{J}_{sf} \\ \mathbf{J}_{fs} & \mathbf{J}_{ff} \end{bmatrix} \quad (5-1)$$

to simplify the notation of the individual terms. Table 5-1 presents the Frob. norms of the difference between the two Jacobians and the number of entries with different signs of the individual Jacobian terms of Eq. (3-87). The zeros in the third column indicate that the signs of all entries are equal for both full Jacobians. The Frob. norms of the difference between the two Jacobians is in the order of 10^{-8} , which is very small.

Jacobian	Frob. norm	No. entry pairs with unequal signs
\mathbf{J}_{ss}^*	$1.3077e - 08$	0
\mathbf{J}_{sf}	$3.2266e - 08$	0
\mathbf{J}_{fs}	$4.0942e - 05$	0
\mathbf{J}_{ff}	$6.6347e - 06$	0

Table 5-1: Two elements, one intersection - Frobenius norms of the *difference* between the analytic and FD Jacobian of each individual part of the Jacobian (* The missing traction contribution has been accounted for, see Figure 5-4)

Two remarks may be added to the Frob. norms of \mathbf{J}_{fs} and \mathbf{J}_{ff} . The first is that the levelset field Jacobian term (LSFJ) \mathbf{J}_{fs} contains some geometrical mismatch contributing to a larger difference between the two Jacobians, which is discussed as Problem 3 in Section 5-3-3. The second is that fluid Jacobian \mathbf{J}_{ff} appeared to be sensitive to the step size. Explanations for this observation are unknown to the author and future research on this might be interesting. On a global level it may be stated that the spy plots and norms show that for configuration Section 5-2a the analytic and FD Jacobian are pretty much equal.

Missing traction contribution Unfortunately the spy-plots and norms in Figure 5-3 are somewhat misleading, as they do not show separate Jacobian contributions. The dependence

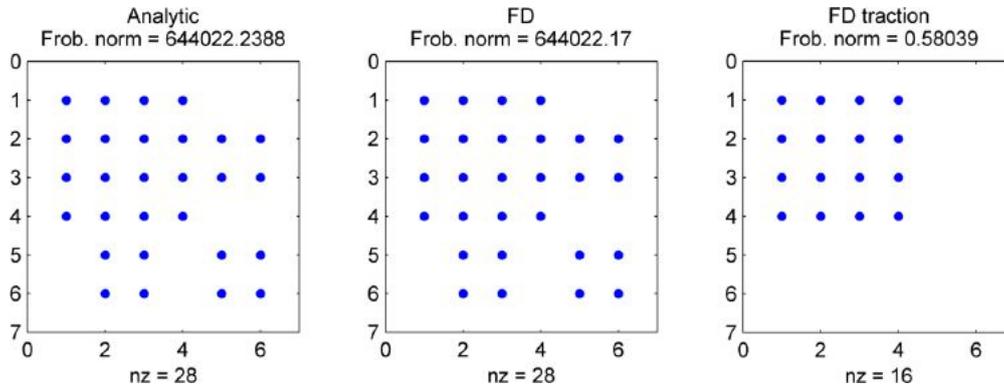


Figure 5-4: Two elements, only left element intersected - Spy plot and Frobenius norm of the analytic structural Jacobian, FD Jacobian and FD traction Jacobian. The Frobenius norm of the FD traction Jacobian is negligible compared to the other Frob. norms.

of the traction on the structural displacements through the fluid levelset is *not* explicitly shown in the complete spy-plots. It is present in the FD Jacobian, but not in the analytic Jacobian and this cannot be found by looking only at the spy-plots. This illustrates the necessity of using multiple methods to compare the Jacobian or even break the analysis down into sub-problems. Figure 5-4 shows that the additional contribution to the structural Jacobian may be neglected as the Frob. norm of the FD traction Jacobian is very small (Frob. norm = 0.5) compared to the actual stiffness matrix entries of the structure (Frob. norm = $6.4e5$). Additionally, the linear structural model used in this work is relatively simple, it converges in one update, and the lack of this contribution will not pose a significant problem. In terms of a numerically consistent Jacobian, it is a minor inconsistency.

5-3-2 Two elements with both elements intersected

The previous section shows promising results as they suggest that the analytic Jacobian is built consistently with the residual for that particular configuration. To fortify this statement, in a more general sense, the other configuration is also tested. In Figure 5-5, the most interesting results for the configuration in Figure 5-2b are presented, as only the \mathbf{J}_{fs} part of the Jacobian shows errors. In App. E, the rest of the results is presented for completeness. Below results are presented that give insight into the problems at hand. To remind the reader: from Eq. (3-85) and Eq. (3-89) can be deduced that the complete term can be written as follows:

$$\mathbf{J}_{fs} = \frac{\partial \mathbf{R}_f}{\partial \phi_f} \frac{\partial \phi_f}{\partial \mathbf{P}^\Gamma} \frac{d\mathbf{P}^\Gamma}{d\mathbf{u}_s} \quad (5-2)$$

This equation mathematically illustrates the following statements:

- The projected intersection points \mathbf{P}^Γ are computed based on nodal structural displacements \mathbf{u}_s ;
- The nodal fluid levelset values ϕ_f are computed based on the projected intersection points \mathbf{P}^Γ ;
- The fluid residual \mathbf{R}_f is computed based on the nodal fluid levelset values ϕ_f .

This information was already presented in Section 3-1-4, but is repeated here, because it is important to be able truly understand the process on both elemental and global level. A quick

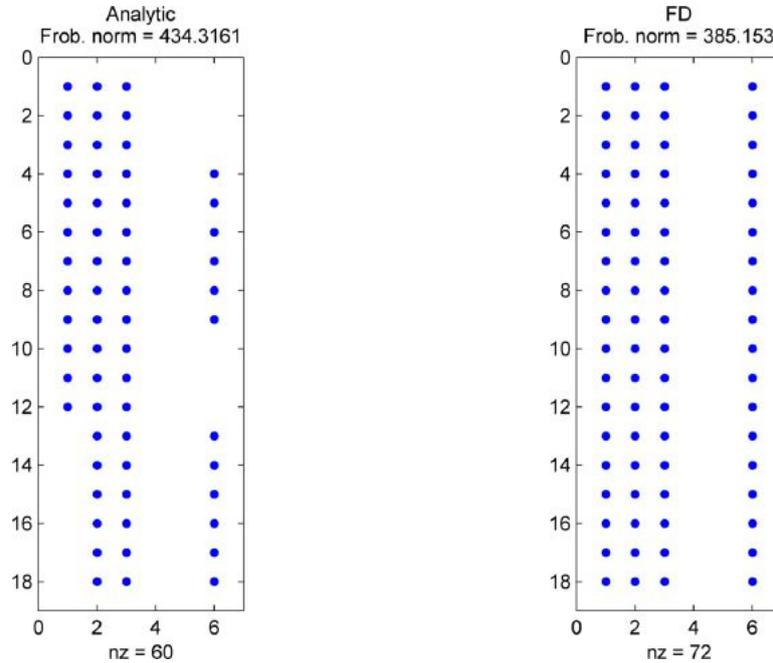


Figure 5-5: Two elements, both intersections - Spy plot and Frobenius norm of the analytic structural \mathbf{J}_{fs} and the FD \mathbf{J}_{fs} . Both entries and Frob. norms differ.

look at the spy-plots in Figure 5-5 shows that the \mathbf{J}_{fs} is off on both the number of entries

($nz = 60$ vs. $nz = 72$) and the absolute values of the entries (Frob. norm = 434 vs Frob. norm = 385). This part of the Jacobian consists of three parts, as presented in Eq. (5-2). The term $\partial \mathbf{R}_f / \partial \phi_f$ is computed with finite differences, while for the term $\partial \phi_f / \partial \mathbf{P}^\Gamma$ and $\partial \mathbf{P}^\Gamma / \partial \mathbf{u}_s$ an analytic approach has been implemented, as presented in Section 3-1-4. The FD term is rather difficult to check and unlikely to be erroneous. The combination of term $\partial \phi_f / \partial \mathbf{P}^\Gamma$ and $\partial \mathbf{P}^\Gamma / \partial \mathbf{u}_s$ equals $\partial \phi / \partial \mathbf{u}_s$ and is easy to check with FD.

Check levelset sensitivities Just like checking the Jacobian of the system by FD, it is possible to check the ‘Jacobian of the levelset field’. This Jacobian gives the sensitivity of the fluid levelset field with respect to the structural displacements, i.e. $\partial \phi / \partial \mathbf{u}_s$ in Eq. (3-90) and Eq. (3-91). The results of the FD and the analytical build-up for the fluid levelset field sensitivities with respect to structural displacements are presented in Table 5-2, Table 5-3 and Table 5-4. In all three tables, each entry contains the first order derivative of ϕ_i (columns) with respect to $u_{x,n}^s$ (rows), where $i = 1..6$ according to the node numbering in Figure 5-6 (see also Eq. (3-90) and Eq. (3-91)). The central question when looking at Table 5-2, Table 5-

$\partial \phi_i$	∂u_{x1}^s	∂u_{x2}^s	∂u_{x3}^s	∂u_{x4}^s	∂u_{x5}^s	∂u_{x6}^s
$\partial \phi_1$	-0.3333	-0.6667	0	0	0	0
$\partial \phi_2$	-0.1179	-0.4714	-0.1179	0	0	0
$\partial \phi_3$	0	-0.2357	-0.2357	0	0	-0.2357
$\partial \phi_4$	-0.1179	-0.4714	-0.1179	0	0	0
$\partial \phi_5$	0	-0.2357	-0.2357	0	0	-0.2357
$\partial \phi_6$	0	0	-0.3333	0	0	-0.6667

Table 5-2: FD on levelset field for both elements - The sensitivities of nodal levelset values ϕ_i with respect to all structural displacements *in x-direction*

$\partial \phi_i$	∂u_{x1}^s	∂u_{x2}^s	∂u_{x3}^s	∂u_{x4}^s	∂u_{x5}^s	∂u_{x6}^s
$\partial \phi_1$	-0.3333	-0.6667	0	0	0	0
$\partial \phi_2$	-0.1179	-0.4714	-0.1179	0	0	0
$\partial \phi_3$	-0.1179	-0.4714	-0.1179	0	0	0
$\partial \phi_4$	-0.1179	-0.4714	-0.1179	0	0	0

Table 5-3: Analytic levelset field for element 1 - The sensitivities of nodal levelset values ϕ_i with respect to all structural displacements *in x-direction*. The red filled cells contain different numbers than the FD results of Table 5-2. The green cells are correct for element 1, but incorrect for element 2, see Table 5-4.

3 and Table 5-4 is *Which nodal fluid levelset values will be affected by a perturbation in the structural x-displacements?* This question is crucial, since it is strongly related to the question whether a residual change will occur when a structural displacement is perturbed. To answer these questions, it is useful to look at Figure 5-6². This schematic will be useful to see to consequences of a structural perturbation, as will be shown Section 5-3-3.

²Note that this configuration was particularly designed such that all nodes except for node 1 and 6 have an orthogonal distance within an element they are connected to.

$\partial\phi_i$	∂u_{x1}^s	∂u_{x2}^s	∂u_{x3}^s	∂u_{x4}^s	∂u_{x5}^s	∂u_{x6}^s
$\partial\phi_2$	0	-0.2357	-0.2357	0	0	-0.2357
$\partial\phi_3$	0	-0.2357	-0.2357	0	0	-0.2357
$\partial\phi_5$	0	-0.2357	-0.2357	0	0	-0.2357
$\partial\phi_6$	0	0.0000	-0.3333	0	0	-0.6667

Table 5-4: Analytic levelset field for element 2 - The sensitivities of nodal levelset values ϕ_i with respect to all structural displacements in x -direction. The red filled cells contain different numbers than the FD results of Table 5-2. The green cells are correct for element 2, but incorrect for element 1, see Table 5-3.

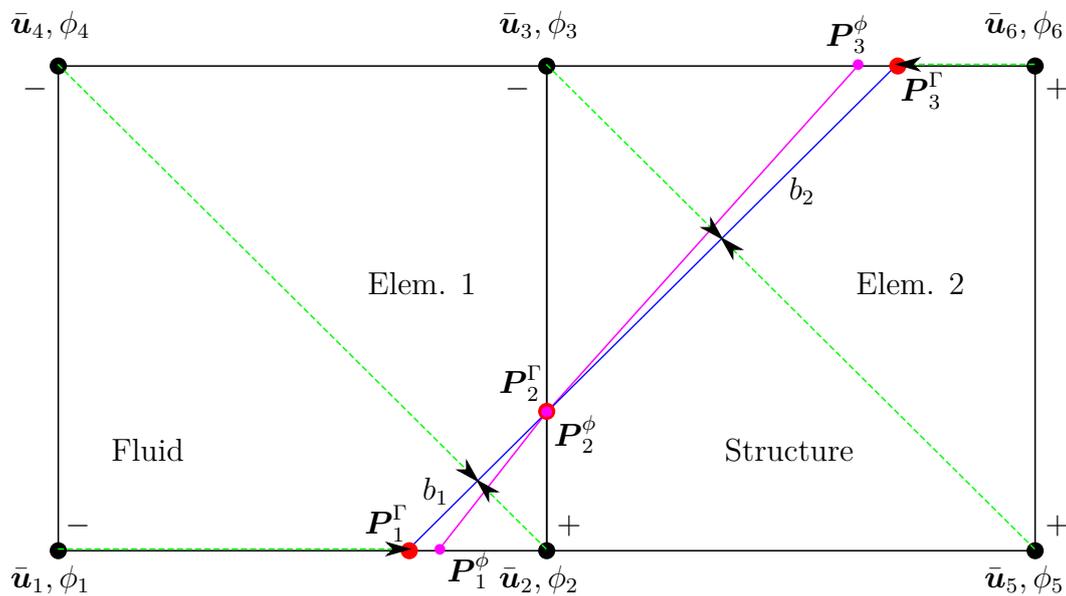


Figure 5-6: A schematic overview of the two element Finite Difference setup of Figure 5-2b - For more information on the content of this figure the reader is referred to items 1 to 10

The following list explains what is shown in Figure 5-6:

1. Every subscripted integer denotes a particular node or point within the global system;
2. The Lagrangian structural mesh is undeformed and hence coincides with the Eulerian fluid mesh, i.e. actually two meshes are present, but it appears to be only one. The same can be stated for the nodes;
3. $\bar{\mathbf{u}}_i$ denote the vectors of nodal degrees of freedom from fluid and structural mesh for the i -th node $[\bar{\mathbf{u}}^s \ \bar{\mathbf{u}}^f]_i = [u_x^s \ u_y^s \ v_x^f \ v_y^f \ p^f]_i$;
4. ϕ_i denote the nodal levelset values on the fluid mesh, which correspond to the norms of the green dashed vectors (-/+ indicates fluid or structural domain, respectively);
5. The straight blue line represents the two structural bars b_1 and b_2 projected onto the fluid mesh. These bars are connected through their end points coinciding with \mathbf{P}_2^Γ ;
6. \mathbf{P}_j^Γ , where $j = 1..3$ indicates the j -th intersection between the structural bars and the element edges, or also projected intersection point;
7. Initial displacements are zero and so the structural bar end points coincide with the projected intersection points \mathbf{P}_j^Γ (see also Figure 3-7 to remember the distinction);
8. \mathbf{P}_k^ϕ , where $k = 1..3$ indicates the k -th intersection between the zero-contour of the fluid levelset based the nodal levelset values ϕ_i and the element edges;
9. The purple line represents the fluid to void interface based on the fluid levelset field (LSF). The reason why the purple and blue line do not coincide is explained in Section 5-3-3 as Problem 3;
10. For node 1 and 6 no orthogonal distance to an interface exists within the domain and so the distance is set equal to the distance to the closest intersection point.

A quick review on the results in Table 5-2, Table 5-3 and Table 5-4, with Figure 5-6 in mind, shows that the nodes connected to only one element (nodes 1,4,5 and 6) have similar results for the analytic and FD setup, denoted by the uncolored cells. The shared nodes (nodes 2 and 3) show good results for the fluid element containing the orthogonal projection for that node, see the green filled cells in Table 5-3 and Table 5-4. For instance, node 2 has an orthogonal distance within element 1 and Table 5-3 indicates that the derivatives are equal to the FD results. The red filled cells indicate erroneous information compared to the equivalent FD values in Table 5-2. These incorrect fluid LSF Jacobian entries will partly explain, why spy Frob. norms in Figure 5-5 are different, if neighboring elements are intersected. Further analysis is presented in the next section.

5-3-3 Detailed analysis of results

The previous section shows that trouble occurs with the levelset derivatives, as soon as multiple *neighboring* elements are intersected. In other words, the analytic Jacobian is not consistent. To understand where the problem originates from, we have go back to the levelset

procedure as described in Section 3-1-4 and Section 5-2. In short, the procedure finds the signed orthogonal distance to the *closest* interface and the sensitivity of this distance with respect to nodal structural displacements. A schematic overview of the two element problem with these orthogonal distances, depicted by the green dashed arrows, is shown in Figure 5-6. It may be noted that finding the nodal levelset value is done on a *neighboring element* level, by looking for the closest interface within the all elements connected to the node in question. This results in dependencies, according Eq. (3-1) to Eq. (3-13), as shown in Table 5-5. An

Nod. LS value	Domain	Proj. int. points	Nod. displacements (x and y)
ϕ_1	- (fluid)	P_1^Γ	$\bar{u}_1^s \bar{u}_2^s \bar{u}_3^s \bar{u}_4^s$
ϕ_2	+ (structure)	$P_1^\Gamma P_2^\Gamma$	$\bar{u}_1^s \bar{u}_2^s \bar{u}_3^s \bar{u}_4^s$
ϕ_3	- (fluid)	$P_3^\Gamma P_2^\Gamma$	$\bar{u}_2^s \bar{u}_5^s \bar{u}_6^s \bar{u}_3^s$
ϕ_4	- (fluid)	$P_1^\Gamma P_2^\Gamma$	$\bar{u}_1^s \bar{u}_2^s \bar{u}_3^s \bar{u}_4^s$
ϕ_5	+ (structure)	$P_3^\Gamma P_2^\Gamma$	$\bar{u}_2^s \bar{u}_5^s \bar{u}_6^s \bar{u}_3^s$
ϕ_6	+ (structure)	P_3^Γ	$\bar{u}_2^s \bar{u}_5^s \bar{u}_6^s \bar{u}_3^s$

Table 5-5: Nodal levelset dependencies on projected intersection points and structural displacements - The rows starting with the blue filled cells contain information the shared nodes of Figure 5-6. The projection intersection points and the nodal displacements have different parental elements for both nodes.

example helps to understand the information in Table 5-5:

1. Take the upper left node left node of element 1, i.e. node 4;
2. The green arrow points at the closest interface and hence its length is the absolute nodal levelset value (- indicates it is a fluid node, see column 2);
3. The closest interface for node 4 is defined between the projected intersection points P_1^Γ and P_2^Γ (see column 3);
4. Points P_1^Γ and P_2^Γ are the intersections of structural bar b_1 and the lower and right element edge of element 1;
5. The location of bar b_1 is determined nodal structural displacements \bar{u}_1 to \bar{u}_4 through the bar ending points (see column 4).

The previous steps follow the reversed process of Section 3-1-4 and hold for each node to fill a row in Table 5-5. This table shows that the shared nodes 2 and 3 couple the two elements through the levelset field formulation; node 2 is connected to element 1, but its nodal levelset value used nodal displacements from element 2 (see column 4).

Eventually, it is the goal to compute the derivative of the fluid residual with respect to the structural displacements, see Eq. (5-2). This equation shows three different pieces of the puzzle that need to be connected. The MATLAB-code that connects the pieces can be found in Appendix D. For readers not familiar with the rest of the code and all the variable names,

a short summary of the code is presented in Alg. 2 in Appendix C. Alg. 2 indicates that each intersected fluid element is regarded individually by looping over all intersected fluid elements. This was already indicated with Eq. (3-89). Hence for each individual element, it is determined which structural bar intersects with one of the 4 element edges. As is shown in Figure 5-6, element 1 intersected by b_1 and element 2 by b_2 . This is where the first problem with the current code structure appears. Additionally, the above gives information to identify in total three problems.

Problem 1 - Nodal dependence on neighboring elements Suppose the loop in Alg. 2 in Appendix C has started and element 1 is being regarded in Figure 5-6. Element 1 contains projected intersection points \mathbf{P}_1^Γ and \mathbf{P}_2^Γ . The derivative of these projection points with respect to the four nodal structural displacements \mathbf{u}_i^s (Eq. (3-32) or the third term in Eq. (5-2) on the right hand side) for element 1 can be written as follows:

$$\frac{d\mathbf{P}_j^\Gamma}{d\mathbf{u}_n^s} = \begin{bmatrix} \frac{d\mathbf{P}_1^\Gamma}{d\mathbf{u}_n^s} \\ \frac{d\mathbf{P}_2^\Gamma}{d\mathbf{u}_n^s} \end{bmatrix} . \quad (5-3)$$

where $j = 1,2$ to indicate projected intersection points 1 and 2. Now we zoom in on node 3. In Table 5-5, we see that the nodal levelset value for node 3 is determined with projected intersection points \mathbf{P}_2^Γ and \mathbf{P}_3^Γ , such that the derivative of the levelset value with respect to the projected intersection points can be written as follows:

$$\frac{\partial\phi_3}{\partial\mathbf{P}_j^\Gamma} = \begin{bmatrix} \frac{\partial\phi_3}{\partial\mathbf{P}_2^\Gamma} & \frac{\partial\phi_3}{\partial\mathbf{P}_3^\Gamma} \end{bmatrix} , \quad (5-4)$$

where $j = 2,3$ to indicate projected intersection points 2 and 3. When equation Eq. (5-3) and Eq. (5-4) are multiplied a mismatch occurs because different projected intersection points are used. The nodal level-set value is connected to the wrong nodal displacements. So even though Eq. (5-3) and Eq. (5-4) may individually be computed correctly, they are non fitting puzzle pieces, on an elemental level. The same holds for node 2, if element 2 is regarded. Essentially, this mismatch will always occur in the current implementation, when a particular node within an element uses information from a neighboring intersected element. The result is an erroneous $\partial\phi/\mathbf{u}_s$ term on an elemental level, which was already shown in Table 5-2 to Table 5-4 with the red filled cells. As the levelset update procedure is developed, such that the nodal levelset values are independent of the element that is regarded, one should find equal values for entries for node 2 and 3 in both elements. This is, however, not the case as can be seen in the red and green filled cells in Table 5-3 and Table 5-4.

Problem 1 points out the fact that the current implementation needs to be improved. Eq. (3-89) is computed on an elemental level, but should be computed on a nodal level and then stored on an elemental level. This problem can be solved by keeping track of which element contains the orthogonal distance to the closest interface for each node individually. Important to note is that the correct information is already computed, if the element containing the orthogonal distance is regarded. This means minor changes to the current implementation.

Problem 2 - Secondary coupling Another problem with the current code setup relates to the missing entries in the analytic Jacobian, see Figure 5-5. This problem can be understood by regarding a perturbed system. The central FD-loop to determine the Finite Differenced Jacobian starts with perturbing the structural DOFs. The first DOF that is perturbed (backwards) is the x-displacement of node 1. This perturbation results in a displacement of structural bar b_1 , according to Eq. (3-3), as is shown in Figure 5-7. In Figure 5-7, an exaggerated

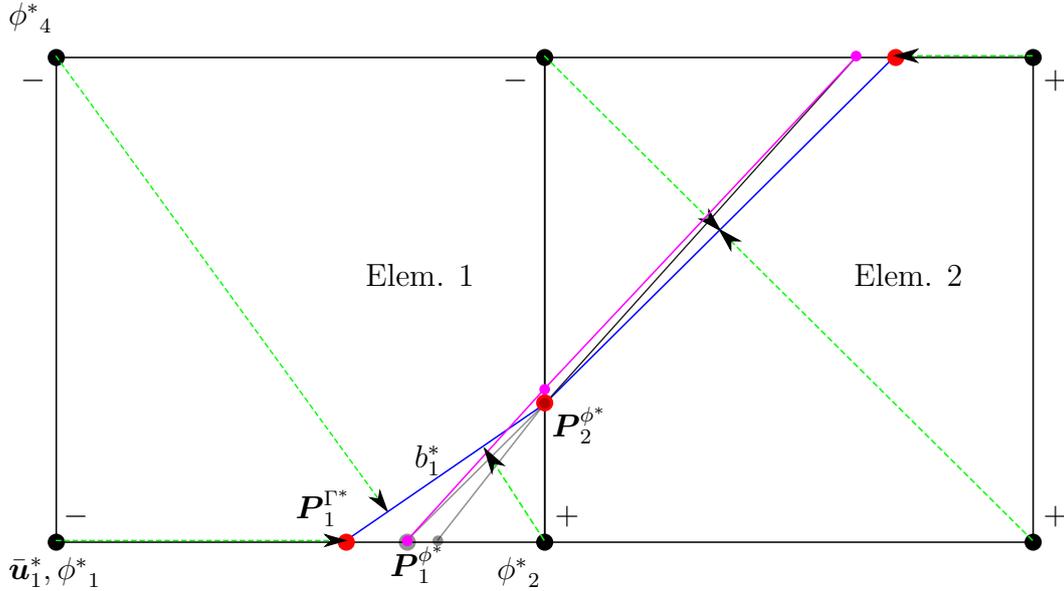


Figure 5-7: Two element setup with x-displacement of node 1 perturbed - Superscript * indicates a changed variable, due to a structural perturbation of node 1. The grey lines and dots in the background depict the Figure 5-6.

perturbation of nodal x-displacement of node 1 and its effects is shown in colors, while grey in the background indicates the original situation of Figure 5-6. From this figure, it can be concluded that a perturbation of displacement in node 1, not only affects the fluid residual of element 1 but also the fluid residual of element 2, meaning coupling on neighboring element level has been introduced. The transition from Figure 5-6 to Figure 5-7 occurs according to Section 3-1-4 and can be summarized as follows:

- The structural perturbation displaces the bar ending points;
- The displacement of the bar results in changed intersection points;
- The changed intersection points give a changed nodal levelset values;
- The changed nodal levelset values give a changed zero contour of the LSF;
- A changed zero contour means changed intersections between the zero contour and the element edges;

Figure 5-6 shows green arrows from node 2 and 4 touching at the interface, while Figure 5-7 shows that these arrows have shifted. The nodal levelset value of node 2 has changed, due to the perturbed displacement in node 1, and that results in shifted zero contour points ϕ_j^*

on the element edges of both element 1 and 2. The zero contour points on the shared element edge, for the configuration in Figure 5-6, can be computed as follows:

$$\mathbf{P}_2^\phi = \mathbf{P}_2^n + \frac{\phi_2}{\phi_2 - \phi_3} \cdot (\mathbf{P}_3^n - \mathbf{P}_2^n) \quad . \quad (5-5)$$

The zero contour points on the shared element edge, for the configuration in Figure 5-7, can be computed as follows:

$$\mathbf{P}_2^{\phi^*} = \mathbf{P}_2^n + \frac{\phi_2^*}{\phi_2^* - \phi_3} \cdot (\mathbf{P}_3^n - \mathbf{P}_2^n) \quad ; \quad (5-6)$$

and because $\phi_2^* > \phi_2$, due to the shift of the structural bar, we can say:

$$\mathbf{P}_2^\phi \neq \mathbf{P}_2^{\phi^*} \quad . \quad (5-7)$$

The purple in Figure 5-7, the fluid to void interface, has moved in element 2 and hence the elemental (domain of integration, see Figure 3-16) and no-slip condition contributions (projection, see Figure 3-18³) to the residual are affected. Concluding; the fluid LSF introduces additional coupling between neighboring elements, which is not present in normal FEM. Similar conclusions hold for perturbing nodes 2,3 and 6. Nodes 4 and 5 are not connected to an intersected element edge and hence will not give any contribution to the location of the bar ending points (shape functions are zero in Eq. (3-32)). This analysis is perfectly reflected by the filled columns 1,2,3 and 6 and the empty entries in columns 4 and 5 on the right spy-plot in Figure 5-5. Nodes 1,2,3, and 6 are coupled to both elements and a change in displacements at these nodes gives a change in the fluid residual. The latter is the definition of an individual Jacobian entry.

In the current code setup, this secondary effect is not incorporated and that explains the missing entries in the analytic Jacobian. Extending this line thought towards the full model of Section 3-5, one needs to incorporate these secondary dependencies in future work in addition to the local dependencies. The local dependencies are already calculated correctly as is illustrated by the single intersection 2 element case.

Every nodal displacement value determining the location of one of the structural bar ending points is coupled to all elements containing element edges intersecting with that structural bar.

It may, however, be possible to ignore these secondary effects as they appear to be small in Figure 5-7, but this can only be checked, if Problem 1 is fixed first. The current implementation is written from the perspective of the fluid elements, so it is recommended to address this problem from the perspective of the structural bars. An important question to investigate in this perspective is ‘How many intersection points does a structural bar have with the fluid mesh?’. All these intersection points depend on the location of the bar ending points and hence depend on the structural displacements determining the location of the bar ending points. More research on this topic is needed.

³In hindsight, we unknowingly had assumed that the projected intersection points and zero contour intersection points coincided. This is only true for long straight geometries, such as the left and right side of the beam in this particular problem setup.

Problem 3 - Zero contour and projection mismatch There is a third problem with the current code structure regarding J_{fs} , namely the mismatch between the projected intersection points \mathbf{P}_j^Γ and zero levelset contour points \mathbf{P}_j^ϕ , where $j = 1..3$. In Figure 5-6, this mismatch is illustrated. The location of the zero levelset contour along the bottom element edge of element 1 can be computed as follows:

$$\mathbf{P}_1^\phi = \mathbf{P}_1^n + \frac{\phi_1}{\phi_1 - \phi_2} (\mathbf{P}_2^n - \mathbf{P}_1^n) \quad . \quad (5-8)$$

The location of the projected intersection point along the bottom element edge of element 1 was already computed in Eq. (3-7) as:

$$\mathbf{P}_1^\Gamma = \mathbf{P}_1^n + r(\mathbf{P}_2^n - \mathbf{P}_1^n) \quad , \quad (5-9)$$

or equivalently:

$$r = \frac{\mathbf{P}_1^\Gamma - \mathbf{P}_1^n}{\mathbf{P}_2^n - \mathbf{P}_1^n} \quad (5-10)$$

where r is the local coordinate of the projected intersection point along the the element edge. Eq. (5-8) and Eq. (5-9) are very similar equations and will produce equal results, i.e. $\mathbf{P}_1^\phi = \mathbf{P}_1^\Gamma$, if the following is true:

$$r = \frac{\mathbf{P}_1^\Gamma - \mathbf{P}_1^n}{\mathbf{P}_2^n - \mathbf{P}_1^n} = \frac{\phi_1}{\phi_1 - \phi_2} \quad (5-11)$$

To investigate Eq. (5-11), we draw a schematic of the bottom edge of element 1 in Figure 5-6 and structural bar b_1 , as shown in Figure 5-8. In Section 3-1-4, it was explained that, if a

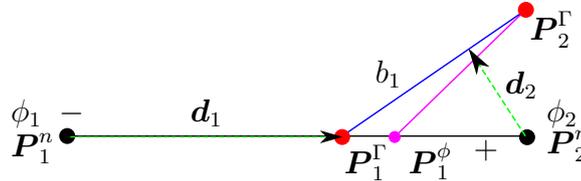


Figure 5-8: Mismatch between the projected intersection points and the zero contour intersection points for 2 element case - The projected intersection point \mathbf{P}_1^Γ does not coincide with the zero contour intersection point \mathbf{P}_1^ϕ , because the nodal absolute nodal levelset value d_2 is not equal to $(\mathbf{P}_2^n - \mathbf{P}_1^\Gamma)$.

node does not have an orthogonal distance within an element, the absolute nodal levelset value will be set equal to the distance from that node to the closest intersection point. Node 1 in Figure 5-6 does not have an orthogonal distance and the closest projected intersection point is \mathbf{P}_1^Γ . This means that ϕ_1 can be defined as:

$$\phi_1 = -\|\mathbf{d}_1\| = -(\mathbf{P}_1^\Gamma - \mathbf{P}_1^n) \quad , \quad (5-12)$$

where the minus sign corresponds to the sign convention of negative fluid levelset values. If we plug Eq. (5-12) into Eq. (5-11), we can deduce a relation for ϕ_2 as follows:

$$\frac{\mathbf{P}_1^\Gamma - \mathbf{P}_1^n}{\mathbf{P}_2^n - \mathbf{P}_1^n} = \frac{-(\mathbf{P}_1^\Gamma - \mathbf{P}_1^n)}{-(\mathbf{P}_1^\Gamma - \mathbf{P}_1^n) - \phi_2} \rightarrow \phi_2 = \mathbf{P}_2^n - \mathbf{P}_1^\Gamma \quad . \quad (5-13)$$

However, Eq. (5-13) contradicts with what we see in Figure 5-8, namely that the orthogonal distance of node 2 is given by \mathbf{d}_2 such that:

$$\phi_2 = \|\mathbf{d}_2\| < \mathbf{P}_2^n - \mathbf{P}_1^\Gamma . \quad (5-14)$$

From this, we can conclude that Eq. (5-11) is not true and that $\mathbf{P}_1^\phi \neq \mathbf{P}_1^\Gamma$. This is visually supported by Figure 5-8. Since $\|\mathbf{d}_2\| < \mathbf{P}_2^n - \mathbf{P}_1^\Gamma$ the zero contour point \mathbf{P}_1^ϕ will lie to the right of \mathbf{P}_1^Γ .

So far, this analysis has focused on the 2 element case of Figure 5-2b and Figure 5-6, where the lower left node was used to illustrate the mismatch between the zero contour intersection points and the projected intersection points. This node does not have an orthogonal distance within an element and that is the cause of the mismatch. However, the same analysis can be done for the situation shown in Figure 5-9. In this figure a lower element contains a vertical bar, such that the nodal levelset value is correctly set to $-(\mathbf{P}_2^n - \mathbf{P}_1^\Gamma)$. It is the explanation for the mesh discrepancy mentioned in Section 3-1-6. Figure 5-9 shows that, if an element

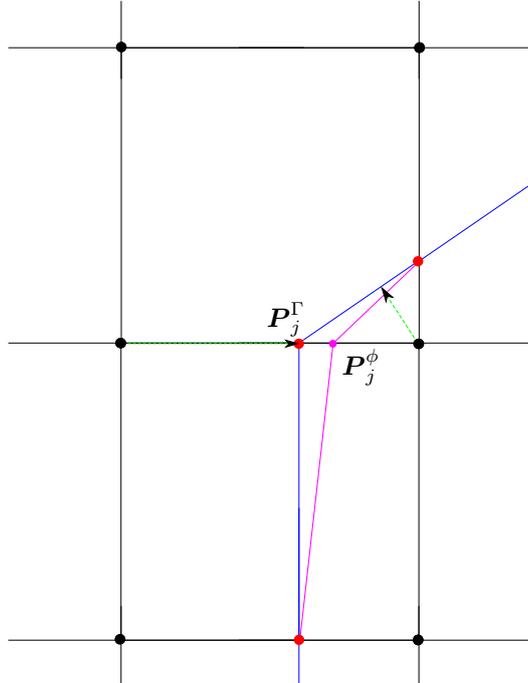


Figure 5-9: Zero contour and projection points mismatch at the top of the structural beam - The fluid residual relates to the purple interface, while the levelset field relates to the blue interface.

sharing the edge in Figure 5-8 contains a vertical bar, the distance is correctly assigned as $-(\mathbf{P}_1^\Gamma - \mathbf{P}_1^n)$, resulting in the mesh discrepancy.

Essentially, if two neighboring elements contain structural bars connected under an angle, this geometric mismatch shows up. The bigger the angle the bigger the mismatch and hence with heavily curved interfaces, such as to top of the beam, the mismatch/discrepancy becomes bigger and bigger. The mesh discrepancy is handled with projection methods, but there is another problem. The nodal levelset values are computed based on the projected intersection points \mathbf{P}_j^Γ and zero levelset contour is calculated using the nodal levelset values ϕ_i . The other

potential problem arising from this geometric mismatch is related to Eq. (5-2). The fluid residual derivative is computed based on the zero levelset contour of the fluid levelset \mathbf{P}_i^ϕ . The levelset derivative, however, is computed based on the projected intersection points \mathbf{P}_i^Γ . This means that both terms do not regard the variation around the same interface, as the projected intersection points and the zero contour intersection points do not always coincide. This may be written down as follows:

$$\frac{d\mathbf{R}_f}{d\mathbf{u}_s} = \frac{\partial\mathbf{R}_f}{\partial\phi_f[\mathbf{P}^\phi]} \frac{d\phi_f[\mathbf{P}^\Gamma]}{d\mathbf{u}_s}, \quad (5-15)$$

where the square brackets indicate ‘*variation around*’ and not dependence. This problem only affects the Jacobian of the fluid and it is expected that finer meshes will show less mismatch. The mismatch is inherent to the current implementation of the used levelset method and the extent of the consequences should be investigated in future work, noting that the first two problems should be addressed first. A possible fix is to hack into the fluid levelset routine of Section 3-1-4 and compute the zero levelset contour points based on the fluid levelset and then use those points to calculate the levelset related derivatives instead.

5-3-4 Choosing a correct 2 element problem

As mentioned, FD gives a lot of good information on potential problems, but designing the correct 2 element problem is not trivial. Below some remarks are presented, that could help the design process in future work. This list presents some lessons learned during this work, but it cannot guarantee an ever successful FD process.

- Initialization of fluid - The fluid should be initialized with a non-zero velocity field and non-uniform pressure field. Otherwise the \mathbf{J}_{f_s} Jacobian term shows big differences with the FD Jacobian. Future work can be done to find the reasons for this problem.
- Geometric symmetry - Symmetric configurations often do not show errors or even introduce additional problems. It can be helpful to try symmetric configurations and to see the difference between non-symmetric configurations, however, solely looking at symmetric problems most likely will not provide enough conclusive information. To illustrate: Problem 1 will always occur for nodes with equal orthogonal distance within multiple elements, see Figure 5-10. The min-operation⁴ in MATLAB then just chooses the first entry. The consequent problem then comes from the fact the sensitivity of the distances with respect to \mathbf{P}_1^Γ and \mathbf{P}_3^Γ show opposite signs and hence derivative information from one element is incorrect for the other element. One should take care of preventing similar configurations.
- No orthogonal distance within the connected element - In some configurations, node 4 and 5 do not have an orthogonal distance within the one element they are connected to and hence the distance is set to the closest intersection point, see Figure 5-11. Similar configurations can occur in the complete model and this has the potential to introduce errors. This problem is also related to the use of the min-operation.

⁴<http://www.mathworks.nl/help/matlab/ref/min.html>

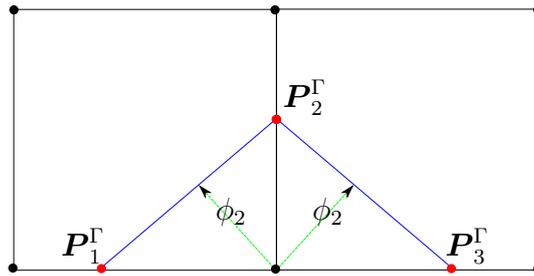


Figure 5-10: Symmetric two element setup with middle bottom node having equal orthogonal distance within both elements

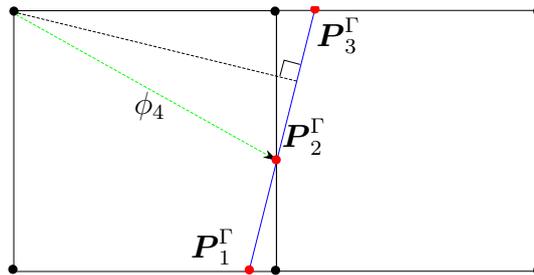


Figure 5-11: The orthogonal distance to the closest interface from upper left node is depicted by the black dashed line, however, the min-operation sets it to the length of the green dashed arrow

- No perturbation of y -displacements - In the two element cases of this chapter, structural perturbations in y -direction are skipped, because they result in bar ending points located outside the domain. It is possible to design 2 element configurations, that shift the problem from the y -direction to the x -direction (top three nodes have equal nodal fluid levelset sign, but opposite to sign of bottom three). This configuration has not been investigated and hence it is impossible to state anything about the consequences of the structural perturbations in y -direction. It is, however, unlikely that major new problems occur when perturbing in y -direction.

5-4 Summary

The current implementation of the XFEM model is unable to find the steady state solution with the monolithic setup. Since it was shown that a staggered solver is able to solve the system, the Jacobian was the next target to investigate. The Jacobian was investigated using finite differences and two different simplified problems. The first problem was a two element problem with one intersected element. This setup did not show any major problems and it was used to motivate to ignore the dependence of the traction on the fluid LSF.

The other problem was also a two, element problem but with both elements intersected. This problem showed significant differences between the analytic and FD Jacobian, regarding the LSFJ \mathbf{J}_{fs} term and more specifically the $\partial\phi_f/\partial\mathbf{u}_s$ term. This indicates that the Jacobian is inconsistent for this problem. Three problems related to this term have been identified and explain why the system can not be solved monolithically:

- Nodes regarded from a particular element give erroneous multiplications for levelset field derivative terms when information from a neighboring element is needed;
- The fluid LSF introduces secondary coupling between elements and this coupling is not incorporated in the current setup;
- Fluid residuals and fluid LSF calculations are performed on different domains, due to a small geometric mismatch.

The first problem needs to be fixed first, by keeping track of what fluid contains the interface associated with the nodal levelset value. Regarding the second problem it should be investigated, whether the secondary coupling contributes significantly. If so, it should be fixed, knowing that each nodal displacement, associated to the a structural bar ending point, is coupled to all elements containing element edges intersecting with that structural bar. The third problem decreases with a finer mesh, but its consequences are yet to be determined. Designing a suitable simplified problem is not easy and one should be aware of the difficulties. The list presented in Section 5-3-4 provides information on encountered problems in the design process.

Chapter 6

Discussion

The XFEM model, developed and analyzed in this research, is conceptually complex and difficult to implement correctly. The current implementation needs to be fixed and improved, in order to robustly solve the system with the monolithic setup and to use it within an optimization framework. The problems regard the construction of a numerically consistent Jacobian. In addition, the code is not yet capable of handling all potential intersection configurations. This leads to less flexibility, regarding modeled geometries, because the mesh has to be adapted to prevent certain configurations. Additionally, a loss of traction information in certain configurations reduces the accuracy of the steady state solution. Using a fine mesh is advised, because this reduces the chances of unmanageable intersections occurring and it reduces the mesh discrepancies. An optimization procedure should have as much freedom as possible to update the geometry without running into problems with the model itself and that is why more work on the implementation is needed. It may be noted that the suggested fixes cannot guarantee a problem free model. However, fixing the mentioned problems is the first step for future research. Without these fixes the model can never be tested to its full potential in an optimization framework. An interesting question to ask at this point is ‘*Is it worth the effort?*’. The next section shows that it is definitely worth the effort.

Multiple beams

The big advantage of this model is that it is easy to change the initial geometry of the structure. Since the LSF is defined with a signed distance function, the LSF zero contour can be changed with minimal effort. For instance, adding an extra structural beam on the same Lagrangian mesh gives the steady state shown in Figure 6-1. The second beam in Figure 6-1 is added by the using the same MATLAB routine to generate the first bar, but with a spatial shift. In both routines, there is an if-statement to check whether the y-coordinate of a node lies below the height of the beam using a \leq sign, see also Section 3-1-2. This if-statement defines structural beams at the bottom wall. If we change that sign in one routine to \geq , reduce the mean inlet velocity with a factor 10, and also assign the clamped boundary condition to the top wall, the beam flips to the top wall and the result is shown in Figure 6-2. Both

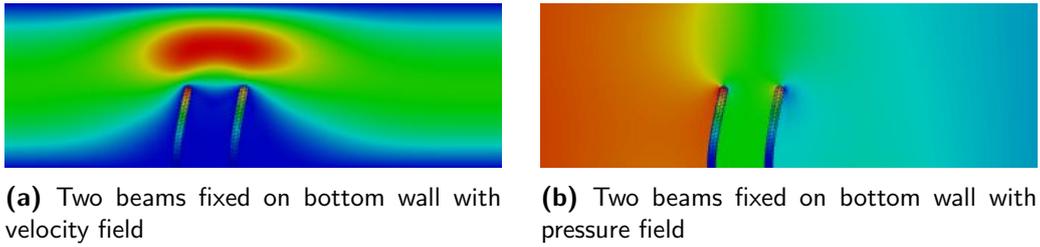


Figure 6-1: Two beams fixed at the bottom wall

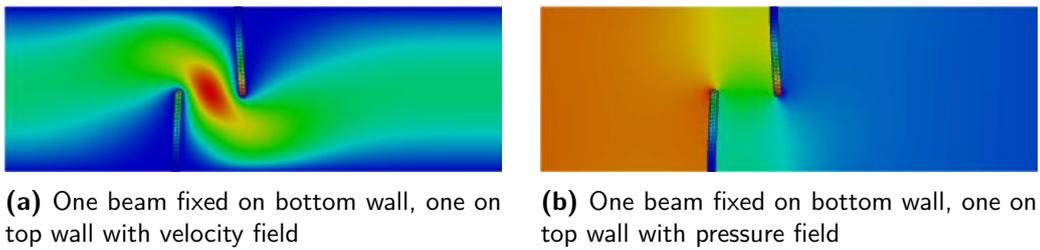


Figure 6-2: One beam fixed on bottom wall, one on top wall

Figure 6-1 and Figure 6-2 already show consequences of changing the topology of the system. Suppose one requires minimal displacements of the left beam, the design in Figure 6-2 is more suitable. If we drop the routine used for Figure 6-1 and Figure 6-2, where a signed distance function is used, and just define an initial LSF with 1 and -1, the freedom to choose an initial configuration is endless. With only 1 for-loop and two if-statements a capital letter T can be modeled. This is shown in Figure 6-3. The figures, shown in this section, give some taste of

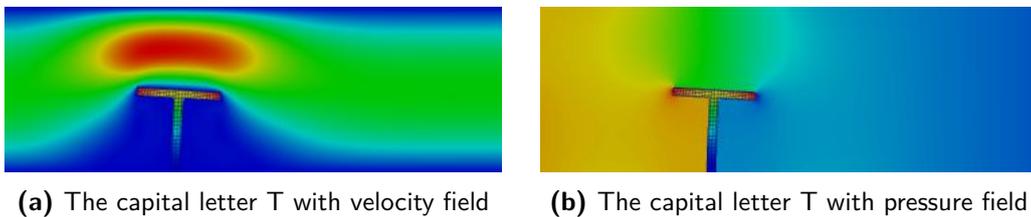


Figure 6-3: FSI simulation of the capital letter T

the flexible geometry description referred to so often in this report. *For all the problems the same mesh has been used.* The figures visualize that it is worth the effort to fix the monolithic solver and use it in an optimization procedure. In these examples, the initial configurations are defined manually, but it is clear that an optimization algorithm perform this task quite easily in order to improve the system's performance to the desired objective. Using this model for optimization purposes would be a very interesting research topic, since the alternatives, such as the density-based approach, struggle with the physics at the interface. It has the potential to be applied successfully to optimization of FSI problems that are more complex than, for instance, given in Yoon [2010]. In that study, they mention that levelset based modeling for FSI problems has greater potential than their method, regarding optimization of transient FSI problems. Since the current implementation is not yet ready, the next section presents a list of research topics for the future.

Conclusions and Recommendations

The ultimate goal of this research was to solve the system monolithically using a numerically consistent Jacobian, however the current setup needs to be improved to be able to do so. The XFEM model has not been tested to its full potential, because the system can not be solved monolithically yet. The method will show its benefits, in particular, when it is used within an optimization framework. The crisp FSI boundary definition gives good physical behavior in the vicinity of the interfaces, which is a major advantage compared to density based methods for optimization.

The fluid-structure interface is described with a LSM. The LSM of this research uses a signed distance function to determine the nodal levelset values. This setup allows the computation of analytical derivatives, that are needed for a numerically consistent Jacobian. Some drawbacks of the current implementation are the ever present discrepancy between the two meshes and the inability to robustly handle every geometric intersection configuration between the two superposed meshes.

Simulation with a staggered setup showed that the model was capable of finding a steady state solution close to the solution of a COMSOL simulation. The staggered setup ignored parts of the Jacobian such that potential errors in those parts did not affect the solving process. Based on quantitative analysis of the solving process, it was philosophized that even with a numerically consistent Jacobian the current model will not show quadratic convergence rates in the monolithic setup, due to the changing DOFs on the fluid mesh.

A finite differences check on the Jacobian showed that it was not numerically consistent, which explains the observation that the monolithic setup could not find the steady state solution. Three different problems are identified that need to be fixed and/or investigated further. The first problem occurs when nodes, connected to an intersected element, use interface information from an neighboring intersected element to compute the nodal levelset value and its derivatives. The second problem is related to introduction of secondary coupling between elements through the LSF, which is not present in the current Jacobian build-up. The third problem regards the geometric mismatch between the zero levelset contour of the fluid and the actual structural projection onto the fluid mesh. The mismatch in this last problem becomes

smaller and smaller with a finer mesh and hence future research should determine whether this has any consequences for a stable monolithic solving process.

Overall it can be stated that this research contributes to the identification of characteristics and limitations of the model and the current implementation. Many different aspects of the model have been investigated, but the main focus was to find a numerically consistent Jacobian. Future research should focus on fixing the monolithic setup and more quantitative validation of the results. The overall concept is still very promising and it opens a world of new and exciting research possibilities. The next section provides suggestions on future research.

Future research

Below, a list of possible research directions is presented. They have been organized into 5 categories, each having their own relevance for future research. The necessary steps have to be taken to continue with this model, the others should be regarded as inspiration. The improvements will increase the robustness over the code. The extensions will produce additional potentially relevant results that are not presented in this report. More validation studies need to be performed in order to confirm that the results of the model are useful. The last category is that of optimization, which was the primary motivation for developing this model.

1. Necessary steps

- Fix monolithic setup - The monolithic setup can be fixed by tackling the three problems mentioned in Section 5-3-3 regarding \mathbf{J}_{fs} . The numerically consistent Jacobian is needed for optimization and is expected to improve computation time significantly. This is particularly desirable for future research.

2. Improvements

- Include pre-conditioner - In XFEM, it is common practice to use a pre-conditioner to improve numerical performance. The current code already has a LSF dependent pre-conditioner implemented, however this to work the fluid LSF Jacobian needs to be fixed first.
- Local mesh refinement - To handle the multiple intersections along one element edge and submerged structural beams, local mesh refinement would be an excellent choice. Smartly dividing the associated elements into smaller elements, if one of the problematic situations occurs, would greatly increase the freedom to choose different meshes and success of an optimization algorithm.
- Tuning the monolithic solver - Newton-Raphson solver can be tuned to improve numerical performance and stability. Two interesting research options, related to the problems mention in Section 2-3, are
 - Relaxation - The size of the update can be regulated by a scalar, as is shown in Eq. (7-1):

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \alpha \cdot \frac{\mathbf{f}(\mathbf{x}_n)}{\mathbf{f}'(\mathbf{x}_n)} \quad . \quad (7-1)$$

The scalar α can be different for both phases and even vary when the solver progresses.

- Quadratic initialization of the structural solution - The levelset initialization defines the fluid-structure interface perpendicular to the horizontal element edges. The $\partial\phi_f/\partial\mathbf{P}^\Gamma$ term, in particular, contains a lot of zeros in this configuration, while it is highly non-linear.

3. Extensions

- Time integration - The current code already has the ability to do time integration, but first the steady state should be found with the monolithic solver.
- Non-linear structural model - The current structure uses a linear elastic material model, which is, actually, no longer valid for the large displacements shown in the results. Implementation of a non-linear structural model will increase the physical relevance of the results.

4. Validation

- Benchmarking - The results of the XFEM model showed differences with the results from the COMSOL simulation. Displacements and pressure differences were the most noticeable. Further research should determine what the reasons for these differences are.
- \mathbf{J}_{ff} sensitivity to step size - The fluid Jacobian and no-slip condition Jacobian appeared to be sensitive to the step size in of the FD. It is unclear which one and why this part of the Jacobian is so sensitive to different step sizes. The initial solution also appeared to have a significant influence on the success of the FD comparison.
- Mesh mismatch - The mismatch between the fluid and structural mesh reduce the accuracy of the results. This discrepancy increases with more complex geometries, which is unfortunate from the perspective of optimization. Some measures to investigate how big this discrepancy is, are the lengths of both material to void interfaces, the areas filled with material and the length of the projection vectors.
- Zero contour of the LSF and projection mismatch - Since this problem reduces with a finer mesh and it should be checked whether this mismatch disturbs the monolithic solving process.
- Finite differences accuracy - Although the current FD setup already provides a lot of interesting information, it can be improved as follows:
 - Make the step size DOF type dependent - The absolute numbers for fluid pressure and velocity and structural displacement differ quite a lot and so the relative impact of the step size differs per type of DOF.
 - Relate the step size of the solution perturbation to the step size of the $\partial\mathbf{R}_f/\partial\phi$ perturbation - In Section 3-3, it was explained that $\partial\mathbf{R}_f/\partial\phi$ is computed with finite differences. For term $\partial\mathbf{R}_f/\partial\phi$ a perturbation of the levelset field gives a geometrical change of the fluid domain. The aim is to make the geometrical change in the projected intersection points $\mathbf{P}_{1,2}^\Gamma$ and fluid levelset zero contour points $\mathbf{P}_{1,2}^\phi$ from the analytic setup for $\partial\mathbf{R}_f/\partial\phi_f$ approximately equal (see Figure 5-6).

5. Optimization

- Sensitivity check - Optimization might be possible with a inconsistent Jacobian. After the fluid LSF Jacobian is fixed, a sensitivity check might show that these terms are negligible for a correct adjoint formulation.
- Gradient based Topology Optimization - As soon as a correct sensitivity information can be computed, the fun begins and the model can show its full potential. It is expected that the model will produce much better physical solutions during optimization of more complex FSI problems.

Appendix A

Derivation of weak form of non dimensional Navier-Stokes equations

This appendix explains how the NV equations are non-dimensionalized and rewritten into the weak formulation, see Section 3-2:

The reference parameters to non-dimensionalize all parameters in this research have the following physical interpretation:

- Reference length L_r - Height of the beam $5e - 5[m]$;
- Reference density ρ_r - Density of the fluid $1000[kg/m^3]$;
- Reference viscosity μ_r - Viscosity of the fluid $1e - 3[Pa \cdot s]$;
- Reference velocity v_r - Mean velocity of the parabolic inlet condition $0.0333[m/s]$;
- Reference pressure p_r - Dynamic pressure defined as $\rho_r \cdot v_r^2 = 1.10889[Pa]$;
- Reference time t_r - The ratio between the reference length and reference velocity $L_r/v_r = 1.5e - 3[s]$

These reference parameters are used for both fluid and structure. For the fluid the dimensional incompressible Navier-Stokes (NS) equations look as follows:

$$\rho_f \frac{\partial v_i}{\partial t} + \rho_f v_j \frac{\partial v_i}{\partial x_j} = \frac{\partial}{\partial x_j} \left(-p \delta_{ij} + \mu \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \right) \quad (A-1)$$

$$\frac{\partial v_i}{\partial x_i} = 0 \quad . \quad (A-2)$$

The non dimensional parameters are defined as follows:

$$\hat{\rho}_f = \frac{\rho_f}{\rho_r}, \quad \hat{v}_i = \frac{v_i}{v_r}, \quad \hat{p} = \frac{p}{\rho_r v_r^2}, \quad \hat{x}_i = \frac{x_i}{L_r},$$
$$\hat{\mu} = \frac{\mu}{L_r \rho_r v_r} = \frac{1}{Re}, \quad \hat{t} = \frac{t}{t_r}, \quad t_r = \frac{L_r}{v_r} \quad , \quad (A-3)$$

where $\hat{\cdot}$ indicates a non dimensional variable, no hat a dimensional variable and subscript r reference parameter.

Plugging the non dimensional parameters in term by term gives:

$$\rho_f \frac{\partial v_i}{\partial t} = \rho_r \hat{\rho} \frac{\partial v_r \hat{v}_i}{\partial t_r \hat{t}} = \frac{\rho_r v_r}{t_r} \left(\hat{\rho} \frac{\partial \hat{v}_i}{\partial \hat{t}} \right) = \frac{\rho_r v_r^2}{L_r} \left(\frac{L_r}{v_r t_r} \right) \left(\hat{\rho} \frac{\partial \hat{v}_i}{\partial \hat{t}} \right) \quad (\text{A-4})$$

$$\rho_f v_j \frac{\partial v_i}{\partial x_j} = \rho_r \hat{\rho} v_r \hat{v}_j \frac{\partial v_r \hat{v}_i}{\partial L_r \hat{x}_j} = \frac{\rho_r v_r^2}{L_r} \left(\hat{\rho} \hat{v}_j \frac{\partial \hat{v}_i}{\partial \hat{x}_j} \right) , \quad (\text{A-5})$$

$$\frac{\partial p}{\partial x_j} \delta_{ij} = \frac{\partial \rho_r v_r^2 \hat{p}}{\partial L_r \hat{x}_j} \delta_{ij} = \frac{\rho_r v_r^2}{L_r} \left(\frac{\partial \hat{p}}{\partial \hat{x}_j} \delta_{ij} \right) , \quad (\text{A-6})$$

$$\begin{aligned} \frac{\partial}{\partial x_j} \left(\mu_r \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \right) \\ = \frac{\partial}{\partial L_r \hat{x}_j} \left(\mu_r \left(\frac{\partial v_r \hat{v}_i}{\partial L_r \hat{x}_j} + \frac{\partial v_r \hat{v}_j}{\partial L_r \hat{x}_i} \right) \right) \\ = \frac{v_r}{L_r^2} \left(\mu_r \left(\frac{\partial^2 \hat{v}_i}{\partial \hat{x}_j \partial \hat{x}_j} + \frac{\partial^2 \hat{v}_j}{\partial \hat{x}_i \partial \hat{x}_j} \right) \right) . \end{aligned} \quad (\text{A-7})$$

Putting all the terms back together again and dividing by $\frac{\rho_r v_r^2}{L_r}$ gives:

$$\frac{L_r}{v_r t_r} \left(\hat{\rho} \frac{\partial \hat{v}_i}{\partial \hat{t}} \right) + \hat{\rho} \hat{v}_j \frac{\partial \hat{v}_i}{\partial \hat{x}_j} = - \frac{\partial \hat{p}}{\partial \hat{x}_j} \delta_{ij} + \frac{\mu_r}{\rho_r v_r L_r} \left(\frac{\partial^2 \hat{v}_i}{\partial \hat{x}_j \partial \hat{x}_j} + \frac{\partial^2 \hat{v}_j}{\partial \hat{x}_i \partial \hat{x}_j} \right) , \quad (\text{A-8})$$

where $\frac{L_r}{v_r t_r} = \frac{1}{St} = 1$ and $\frac{\mu_r}{\rho_r v_r L_r} = \frac{1}{Re} = \hat{\mu}$. St is the non dimensionless Strouhal number is an indication oscillations in the system and Re is the Reynolds number as an indication of the ratio between inertia and viscous forces. The final form we use to derive the weak formulation:

$$\hat{\rho} \frac{\partial \hat{v}_i}{\partial \hat{t}} + \hat{\rho} \hat{v}_j \frac{\partial \hat{v}_i}{\partial \hat{x}_j} = - \frac{\partial \hat{p}}{\partial \hat{x}_j} \delta_{ij} + \hat{\mu} \left(\frac{\partial^2 \hat{v}_i}{\partial \hat{x}_j \partial \hat{x}_j} + \frac{\partial^2 \hat{v}_j}{\partial \hat{x}_i \partial \hat{x}_j} \right) = \frac{\partial \hat{\sigma}_{ij}}{\partial \hat{x}_j} . \quad (\text{A-9})$$

Casting the final last equation into the weak form looks as follows:

$$R_f = \int_{\Omega} \delta v_i \left[\hat{\rho} \frac{\partial \hat{v}_i}{\partial \hat{t}} + \hat{\rho} \hat{v}_j \frac{\partial \hat{v}_i}{\partial \hat{x}_j} + \frac{\partial \hat{p}}{\partial \hat{x}_j} \delta_{ij} - \hat{\mu} \left(\frac{\partial^2 \hat{v}_i}{\partial \hat{x}_j \partial \hat{x}_j} + \frac{\partial^2 \hat{v}_j}{\partial \hat{x}_i \partial \hat{x}_j} \right) \right] d\Omega = 0 . \quad (\text{A-10})$$

Note do not confuse the delta from δv_i with δ_{ij} . The delta with indices is the Kronecker delta. The other delta indicates a variation in velocity. To reduce the order, convert the second derivatives to boundary terms and to be able to enforce boundary traction conditions we will use the Gauss theorem and integration by parts. Doing this term by term gives the following:

$$\begin{aligned} \int_{\Omega} \delta v_i \frac{\partial \hat{p}}{\partial \hat{x}_j} \delta_{ij} d\Omega &= \int_{\Omega} \frac{\partial}{\partial \hat{x}_j} (\delta v_i \hat{p} \delta_{ij}) d\Omega - \int_{\Omega} \frac{\partial \delta v_i}{\partial \hat{x}_j} (\hat{p} \delta_{ij}) d\Omega \\ &= \int_{\Gamma} (\delta v_i \hat{p} \delta_{ij} n_j) d\Gamma - \int_{\Omega} \frac{\partial \delta v_i}{\partial \hat{x}_j} (\hat{p} \delta_{ij}) d\Omega , \end{aligned} \quad (\text{A-11})$$

$$\begin{aligned}
\int_{\Omega} \delta v_i \left(-\hat{\mu} \left(\frac{\partial^2 \hat{v}_i}{\partial \hat{x}_j \partial \hat{x}_j} \right) \right) d\Omega &= - \left[\int_{\Omega} \frac{\partial}{\partial \hat{x}_j} \left(\delta v_i \hat{\mu} \left(\frac{\partial \hat{v}_i}{\partial \hat{x}_j} \right) \right) d\Omega - \int_{\Omega} \frac{\partial \delta v_i}{\partial \hat{x}_j} \hat{\mu} \left(\frac{\partial \hat{v}_i}{\partial \hat{x}_j} \right) d\Omega \right] \\
&= - \left[\int_{\Gamma} \delta v_i \hat{\mu} \left(\frac{\partial \hat{v}_i}{\partial \hat{x}_j} \right) n_j d\Gamma - \int_{\Omega} \frac{\partial \delta v_i}{\partial \hat{x}_j} \hat{\mu} \left(\frac{\partial \hat{v}_i}{\partial \hat{x}_j} \right) d\Omega \right] \quad , \quad (\text{A-12})
\end{aligned}$$

$$\begin{aligned}
\int_{\Omega} \delta v_i \left(-\hat{\mu} \left(\frac{\partial^2 \hat{v}_j}{\partial \hat{x}_i \partial \hat{x}_j} \right) \right) d\Omega &= - \left[\int_{\Omega} \frac{\partial}{\partial \hat{x}_j} \left(\delta v_i \hat{\mu} \left(\frac{\partial \hat{v}_j}{\partial \hat{x}_i} \right) \right) d\Omega - \int_{\Omega} \frac{\partial \delta v_i}{\partial \hat{x}_j} \hat{\mu} \left(\frac{\partial \hat{v}_j}{\partial \hat{x}_i} \right) d\Omega \right] \\
&= - \left[\int_{\Gamma} \delta v_i \hat{\mu} \left(\frac{\partial \hat{v}_j}{\partial \hat{x}_i} \right) n_j d\Gamma - \int_{\Omega} \frac{\partial \delta v_i}{\partial \hat{x}_j} \hat{\mu} \left(\frac{\partial \hat{v}_j}{\partial \hat{x}_i} \right) d\Omega \right] \quad , \quad (\text{A-13})
\end{aligned}$$

where n_j denotes the outward unit vector on interface Γ . Combining all these terms now gives:

$$\begin{aligned}
R_{f,static} &= \int_{\Omega} \delta v_i \hat{\rho} \left(\frac{\partial \hat{v}_i}{\partial \hat{t}} + \hat{v}_j \frac{\partial \hat{v}_i}{\partial \hat{x}_j} \right) + \frac{\partial \delta v_i}{\partial \hat{x}_j} \left(-\hat{p} \delta_{ij} + \hat{\mu} \left(\frac{\partial \hat{v}_i}{\partial \hat{x}_j} + \frac{\partial \hat{v}_j}{\partial \hat{x}_i} \right) \right) d\Omega \\
&\quad - \int_{\Gamma} \delta v_i n_j \left(-\hat{p} \delta_{ij} + \hat{\mu} \left(\frac{\partial \hat{v}_i}{\partial \hat{x}_j} + \frac{\partial \hat{v}_j}{\partial \hat{x}_i} \right) \right) d\Gamma \quad . \quad (\text{A-14})
\end{aligned}$$

The same methods holds for the derivation of the non dimensional incompressibility condition and is straightforward. Hence, the details on that are omitted.

Appendix B

No-slip condition - Condensing out the additional stress field in intersected elements

This appendix shows how to condense out the additional stress field, when applying the no-slip condition at the embedded interface Γ^+ , see Section 3-2-3:

The system of equations in Eq. (3-62) can be written as follows:

$$\boldsymbol{\sigma}^\sigma = [\mathbf{K}_{\sigma\sigma}]^{-1} \{ \mathbf{F}_{\sigma v} - (\mathbf{K}_{\sigma v} + \mathbf{G}_{\sigma v}) \mathbf{v} - \mathbf{K}_{\sigma p} \mathbf{p} \} . \quad (\text{B-1})$$

The contribution to the momentum equations can be written as:

$$- \mathbf{G}_{\sigma v}^T [\mathbf{K}_{\sigma\sigma}]^{-1} \{ \mathbf{F}_{\sigma v} - (\mathbf{K}_{\sigma v} + \mathbf{G}_{\sigma v}) \mathbf{v} + \mathbf{K}_{\sigma p} \mathbf{p} \} . \quad (\text{B-2})$$

Continuing in 2D all contributions are defined below. The contribution from the compatibility condition can be written as:

$$\int_{\Omega} \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{22} \end{bmatrix} \begin{bmatrix} \epsilon_{11}^v - \epsilon_{11}^\sigma \\ \epsilon_{12}^v - \epsilon_{12}^\sigma \\ \epsilon_{22}^v - \epsilon_{22}^\sigma \end{bmatrix} d\Omega \quad (\text{B-3})$$

and if Eq. (3-57), Eq. (3-58) and Eq. (3-59) are plugged in this gives:

$$\int_{\Omega} \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{22} \end{bmatrix} \begin{bmatrix} \epsilon_{11}^v - \frac{1}{2\mu}(\sigma_{11}^\sigma + p) \\ \epsilon_{12}^v - \frac{1}{2\mu}\sigma_{12}^\sigma \\ \epsilon_{22}^v - \frac{1}{2\mu}(\sigma_{22}^\sigma + p) \end{bmatrix} d\Omega . \quad (\text{B-4})$$

The compatibility matrices are the given by:

$$\mathbf{K}_{\sigma\sigma} = -\frac{k}{2\mu} \begin{bmatrix} \int_{\Omega} \gamma_{11}\gamma_{11}d\Omega & 0 & 0 \\ 0 & \int_{\Omega} \gamma_{12}\gamma_{12}d\Omega & 0 \\ 0 & 0 & \int_{\Omega} \gamma_{22}\gamma_{22}d\Omega \end{bmatrix} , \quad (\text{B-5})$$

$$\mathbf{K}_{\sigma p} = -\frac{k}{2\mu} \begin{bmatrix} \int_{\Omega} \gamma_{11} p \\ 0 \\ \int_{\Omega} \gamma_{22} p d\Omega \end{bmatrix}, \quad (\text{B-6})$$

$$\mathbf{K}_{\sigma v} = k \begin{bmatrix} \int_{\Omega} \gamma_{11} \frac{\partial v_1}{\partial x_1} d\Omega & 0 \\ \int_{\Omega} \gamma_{12} \frac{\partial v_1}{\partial x_2} d\Omega & \int_{\Omega} \gamma_{12} \frac{\partial v_2}{\partial x_1} d\Omega \\ 0 & \int_{\Omega} \gamma_{22} \frac{\partial v_2}{\partial x_2} d\Omega \end{bmatrix}. \quad (\text{B-7})$$

The XFEM weak form can be written as:

$$\begin{aligned} \int_{\Gamma^+} \delta v_i n_j \sigma_{ij}^{\sigma} d\Gamma^+ &= \int_{\Gamma^+} (\delta v_1 (n_1 \sigma_{11}^{\sigma} + n_2 \sigma_{12}^{\sigma}) + \delta v_2 (n_1 \sigma_{21}^{\sigma} + n_2 \sigma_{22}^{\sigma})) d\Gamma^+ \\ &= \mathbf{G}_{v\sigma} \boldsymbol{\sigma}_{\sigma} = \mathbf{G}_{v\sigma}^T \boldsymbol{\sigma}_{\sigma} \end{aligned} \quad (\text{B-8})$$

The third and final term in Eq. (3-61) can be written as:

$$\begin{aligned} &\int_{\Gamma^+} \gamma_{ij} n_j (v_i - \dot{u}_i) d\Gamma^+ \\ &= \int_{\Gamma^+} (\gamma_{11} n_1 (v_1 - \dot{u}_1) + \gamma_{12} n_2 (v_1 - \dot{u}_1) + \gamma_{12} n_1 (v_2 - \dot{u}_2) + \gamma_{22} n_2 (v_2 - \dot{u}_2)) d\Gamma^+ \\ &= \int_{\Gamma^+} (\gamma_{11} n_1 v_1 + \gamma_{12} (n_1 v_2 + n_2 v_1) + \gamma_{22} n_2 v_2) d\Gamma^+ \\ &\quad - \int_{\Gamma^+} (\gamma_{11} n_1 \dot{u}_1 + \gamma_{12} (n_1 \dot{u}_2 + n_2 \dot{u}_1) + \gamma_{22} n_2 \dot{u}_2) d\Gamma^+ \\ &= \mathbf{G}_{\sigma v} \mathbf{v} - \mathbf{F}_{\sigma v} \end{aligned} \quad (\text{B-9})$$

with:

$$\mathbf{G}_{\sigma v} \begin{bmatrix} \int_{\Gamma^+} \gamma_{11} v_1 n_1 d\Gamma^+ & 0 \\ \int_{\Gamma^+} \gamma_{12} v_1 n_2 d\Gamma^+ & \int_{\Gamma^+} \gamma_{12} v_2 n_1 d\Gamma^+ \\ 0 & \int_{\Gamma^+} \gamma_{22} v_2 n_2 d\Gamma^+ \end{bmatrix}, \quad (\text{B-10})$$

$$\mathbf{F}_{\sigma v} \begin{bmatrix} \int_{\Gamma^+} \gamma_{11} \dot{u}_1 n_1 d\Gamma^+ \\ \int_{\Gamma^+} \gamma_{12} (\dot{u}_1 n_2 + \dot{u}_2 n_1) d\Gamma^+ \\ \int_{\Gamma^+} \gamma_{22} \dot{u}_2 n_2 d\Gamma^+ \end{bmatrix}. \quad (\text{B-11})$$

Appendix C

Algorithms

Data: Staggered Newton-Raphson solver

Result: Converged solution per phase for current fluid levelset field initialization;

// Staggered loop

```
while  $\|\mathbf{R}\| > \epsilon$  do
  for  $i = 1 : 2$  do
    if  $i = 1$  then
      // Newton-Raphson update stage: Fluid
      for  $ie = 1 : 4$  do
        if  $ie = 1$  then
          | update fluid levelset field  $\phi_f$ 
        end
        build  $\mathbf{R}$  and  $\mathbf{J}$ ;
        store  $\|\mathbf{R}\|$ ;
        if  $\|\mathbf{R}\| < \epsilon_{th}$  then
          | break
        end
        solve and update fluid;
      end
    else
      // Newton-Raphson update stage: Structure
      for  $ie = 1 : 2$  do
        build  $\mathbf{R}$  and  $\mathbf{J}$ ;
        store  $\|\mathbf{R}\|$ ;
        if  $\|\mathbf{R}\| < \epsilon_{th}$  then
          | break
        end
        solve and partially update structure;
      end
    end
  end
end
end
```

Algorithm 1: Steps in the staggered Newton-Raphson solver

Result: The Jacobian contribution $\partial \mathbf{R}_f / \partial \mathbf{u}_s$

```
// Loop over all fluid elements
for i = 1 : no. fld elems do
  if Elem is intersected then
    //  $\partial \mathbf{P}_i^F / \partial \mathbf{u}_s^n$ 
    Calc. Eq. (3-23) to Eq. (3-32)
    Reorganize  $\partial \phi_f / \partial \mathbf{P}_i^F$ 
    //  $\partial \phi_f / \partial \mathbf{u}_s$ 
    Calc. Eq. (3-89)
    // Loop over all nodal LS values
    for in = 1 : 4 do
      | Central FD to calc.  $\partial \mathbf{R}_f / \partial \phi_{in}$ 
    end
    //  $\partial \mathbf{R}_f / \partial \mathbf{u}_s$ 
    Calc. Eq. (3-85)
  end
end
```

end

Algorithm 2: Loop to determine $\partial \mathbf{R}_f / \partial \mathbf{u}_s$ in the current implementation

Result: The FD Jacobian

initialization zero Jacobian matrix;

// Backward ($i = 1$) and Forward loop ($i = 2$)

```
for i = 1 : 2 do
  for ie = 1 : no. of state variables do
    if i = 1 then
      | Perturb solution backwards for state variable ie ;
    else
      | Perturb solution forwards for state variable ie
    end
    Build residual ;
    if i = 1 then
      | Subtract residual from Jacobian-matrix at column ie ;
    else
      | Add residual to Jacobian-matrix at column ie ;
    end
  end
end
```

end

Divide Jacobian by twice the step size;

Algorithm 3: The finite differencing process

Appendix D

MATLAB code for fluid residual to structural displacements Jacobian

```
1 function [drf_dus,dphidus_all] = alexfem_drfdus(numelems_stc,numelems_fld
    ,bardata,ux,uy,fetopo,ex,ey,iscfld,sctfld,sbcfld,...
2     nsbfld,levsign,ep,edof,fsol,fn,fdot,edginfo_glb,uxdot,uydot,ndftot,
    uxdof,uydof,ddistvec_glb)
3
4 % ALEXFEM_DRFDUS
5 % This function computes the derivatives of the non-smoothed fluid
    residual
6 % with respect to the structural displacement dofs. It is needed for the
7 % computation of the derivatives of the smoothed fluid residual with
8 % respect to the structural displacement dofs.
9
10 % get solution vector of any element
11 edf=edof(1,2:end);
12 edt=reshape(fsol(edf,1),1,[]);
13
14 % get total number of dofs per element
15 [ntdof]=alexfem_adddofs(ep(1,:),edt);
16
17 % get total number of elements
18 numelems=numelems_stc+numelems_fld;
19
20 % initialize cell array for storing elemental dphi_dus matrices
21 dphidus_all=cell(numelems,1);
22
23 % initialize final drf/dus Jacobian
24 drf_dus=sparse(ndftot,ndftot);
25
26 % loop over fluid elements
27 for ie=1:numelems_fld
28     iglb=ie+numelems_stc;
```

```

29
30 % check if fluid element is intersected
31 if iscfld(iglb)==0
32     continue;
33 end
34
35 % initialize dphi/dus matrix of element
36 dphidus_elem=sparse(4,ndftot);
37
38 % initialize drf/dphi of element
39 drdphi_elem=sparse(ndftot,4);
40
41 % get number of intersections
42 nsct=iscfld(iglb);
43
44 % edge intersection information
45 edginfo=cell2mat(sctfld(iglb,1:nsct)');
46 edginfo=unique(edginfo,'rows','stable');
47
48
49 % get fluid node coordinates
50 pin=zeros(4,2);
51
52 for in=1:4
53     pin(in,:)=[ex(iglb,in) ey(iglb,in)];
54 end
55
56 % number of intersection points
57 npts=size(edginfo,1);
58
59 % vector of s-values along fluid element edges
60 svalvec=edginfo(:,2);
61
62 % vector of intersected edge IDs
63 edgevec=edginfo(:,3);
64
65 % initialize intersection points coordinates
66 x_gamma=zeros(npts,1);
67 y_gamma=zeros(npts,1);
68
69 % build intersection points coordinates
70 for ip=1:npts
71     edge=edgevec(ip);
72
73     switch edge
74
75         case 1
76             x_gamma(ip)=ex(iglb,1)+svalvec(ip)*abs(ex(iglb,2)-ex(iglb,1));
77             y_gamma(ip)=ey(iglb,1);
78
79         case 2
80             x_gamma(ip)=ex(iglb,2);

```

```

81         y_gamma(ip)=ey(iglb,2)+svalvec(ip)*abs(ey(iglb,3)-ey(iglb
           ,2));
82
83         case 3
84             x_gamma(ip)=ex(iglb,3)-svalvec(ip)*abs(ex(iglb,4)-ex(iglb
           ,3));
85             y_gamma(ip)=ey(iglb,3);
86
87         case 4
88             x_gamma(ip)=ex(iglb,4);
89             y_gamma(ip)=ey(iglb,4)-svalvec(ip)*abs(ey(iglb,4)-ey(iglb
           ,1));
90     end
91 end
92
93 % store intersection point coordinates
94 pi_gamma=[x_gamma y_gamma];
95
96 % associate node IDs with closest respective intersection point IDs
97 gamma_ids=zeros(4,1);
98
99 for in=1:4
100     dist=zeros(npts,1);
101
102     for ip=1:npts
103         dist(ip)=norm([x_gamma(ip) y_gamma(ip)]-[ex(iglb,in) ey(iglb,
           in)]);
104     end
105
106     [~,gamma_ids(in)]=min(dist);
107 end
108
109
110 % Reorganize nodal level-set value derivatives
111 for in=1:4
112     if isempty(ddistvec_glb{iglb,in})
113         ddistvec_glb{iglb,in}=[0 0;0 0];
114     end
115 end
116
117 dphi_dpig{1,1}=[ddistvec_glb{iglb,1}(1,:); ddistvec_glb{iglb,2}(1,:);
118               ddistvec_glb{iglb,3}(1,:); ddistvec_glb{iglb,4}(1,:)];
119 dphi_dpig{2,1}=[ddistvec_glb{iglb,1}(2,:); ddistvec_glb{iglb,2}(2,:);
120               ddistvec_glb{iglb,3}(2,:); ddistvec_glb{iglb,4}(2,:)];
121
122 % get ID of intersecting bar for each intersection point
123 bar_ids=edgeinfo(:,1);
124
125 % get ID of parent structural element for each intersection point
126 stc_ids=zeros(length(bar_ids),1);
127 for ip=1:npts
128     stc_ids(ip)=bardata{bar_ids(ip),1};

```

```

128     end
129
130     % initialize (s,r) for every intersection point
131     s_r=zeros(npts,2);
132
133     % edge map
134     map=[1 2; 2 3; 3 4; 4 1];
135
136     % initialize dr/dub derivatives
137     dr_dub=zeros(npts,4);
138
139     % loop over intersection points: compute (r,s)
140     for ip=1:npts
141         bar_id=bar_ids(ip);
142         edg_id=edginfo(ip,3);
143
144         A=[(bardata{bar_id,3}'-bardata{bar_id,2}') (pin(map(edg_id,1),:))
145             '-pin(map(edg_id,2),:)' )];
146         b=pin(map(edg_id,1),:)'-bardata{bar_id,2}';
147
148         % get inverse of matrix A
149         B=inv(A);
150
151         s_r(ip,:)=B*b; % s_r(:,2) should be equal to edginfo(:,2)
152
153         % compute dr/dub derivatives
154         dr_dub(ip,1)=B(2,1)*(-1+s_r(ip,1)); % dr/dux1b
155         dr_dub(ip,2)=B(2,1)*(-s_r(ip,1)); % dr/dux2b
156         dr_dub(ip,3)=B(2,2)*(-1+s_r(ip,2)); % dr/duy1b
157         dr_dub(ip,4)=B(2,2)*(-s_r(ip,2)); % dr/duy2b
158     end
159
160     % initialize dpi_gamma/dub derivatives
161     dpig_dub=cell(npts,4);
162
163     % loop over intersection points: compute dpi_gamma/dub derivatives
164     for ip=1:npts
165         edg_id=edginfo(ip,3);
166
167         dpig_dub{ip,1}=(pin(map(edg_id,2),:)'-pin(map(edg_id,1),:))'*
168             dr_dub(ip,1); % dpig/dux1b
169         dpig_dub{ip,2}=(pin(map(edg_id,2),:)'-pin(map(edg_id,1),:))'*
170             dr_dub(ip,2); % dpig/dux2b
171         dpig_dub{ip,3}=(pin(map(edg_id,2),:)'-pin(map(edg_id,1),:))'*
172             dr_dub(ip,3); % dpig/duy1b
173         dpig_dub{ip,4}=(pin(map(edg_id,2),:)'-pin(map(edg_id,1),:))'*
174             dr_dub(ip,4); % dpig/duy2b
175     end
176
177     % initialize shape functions cell array for post-multiplication
178     shapemat=cell(npts,2);
179
180     % loop over intersection points of fluid element

```

```

176
177     for ip=1:npts
178         bar_id=bar_ids(ip);
179         stc_id=bardata{bar_ids(ip),1};
180
181         % loop over end points of intersecting bar
182         for ibp=1:2
183
184             % compute local coordinates of bar ending point
185             xlgp=[bardata{bar_id,(1+ibp)}(1) bardata{bar_id,(3+ibp)}(1)];
186             ylgp=[bardata{bar_id,(1+ibp)}(2) bardata{bar_id,(3+ibp)}(2)];
187
188             xloc=[xlgp(1,2) ylgp(1,2)];
189
190             % compute shape functions at bar ending point in solid phase
191             [Ne]=xfem4getsol(ex(stc_id,:),ey(stc_id,:),ep(stc_id,:),xloc,
192                 '1',1,0);
193
194             % store shape functions at bar ending point
195             shapemat{ip,ibp}=Ne;
196         end
197     end
198
199     % initialize dpi_gamma/dun derivatives
200     dpig_dun=cell(npts,4);
201
202     % loop over intersection points: compute dpi_gamma/dun derivatives
203     for ip=1:npts
204
205         % dpig/duxn (structural element of bar ending point 1)
206         dpig_dun{ip,1}=dpig_dub{ip,1}*shapemat{ip,1};
207
208         % dpig/duxn (structural element of bar ending point 2)
209         dpig_dun{ip,2}=dpig_dub{ip,2}*shapemat{ip,2};
210
211         % dpig/duyn (structural element of bar ending point 1)
212         dpig_dun{ip,3}=dpig_dub{ip,3}*shapemat{ip,1};
213
214         % dpig/duyn (structural element of bar ending point 2)
215         dpig_dun{ip,4}=dpig_dub{ip,4}*shapemat{ip,2};
216     end
217
218     % loop over intersection points: compute dphi/duxn and dphi/duyn
219     for ip=1:npts
220
221         % separate dphii/dpig derivatives
222         dphi1_dp=dphi_dpig{ip,1}(1,:)';
223         dphi2_dp=dphi_dpig{ip,1}(2,:)';
224         dphi3_dp=dphi_dpig{ip,1}(3,:)';
225         dphi4_dp=dphi_dpig{ip,1}(4,:)';
226
227         % get ID of parent structural element of intersecting bar
228         stc_id=stc_ids(ip);

```

```

228
229     % get dof sorting info of element
230     [ntdof, numdof, aldofs]=alexfem_adddofs(ep(stc_id,:),edt);
231
232     % initialize dphi/dus for (fld. elem./stc. elem) w/o Lagr. mult.
233     dphidus=zeros(4,numdof);
234
235     % extrct structural dofs IDs
236     dx dof =4:7:numdof;
237     dy dof =5:7:numdof;
238
239     % compute dphi/dus for (fld. elem./stc. elem) w/o Lagr. mult.
240     dphidus(1,dx dof)=dphi1_dp *( dpig_dun{ip,1}+dpig_dun{ip,2});
241     dphidus(1,dy dof)=dphi1_dp *( dpig_dun{ip,3}+dpig_dun{ip,4});
242
243     dphidus(2,dx dof)=dphi2_dp *( dpig_dun{ip,1}+dpig_dun{ip,2});
244     dphidus(2,dy dof)=dphi2_dp *( dpig_dun{ip,3}+dpig_dun{ip,4});
245
246     dphidus(3,dx dof)=dphi3_dp *( dpig_dun{ip,1}+dpig_dun{ip,2});
247     dphidus(3,dy dof)=dphi3_dp *( dpig_dun{ip,3}+dpig_dun{ip,4});
248
249     dphidus(4,dx dof)=dphi4_dp *( dpig_dun{ip,1}+dpig_dun{ip,2});
250     dphidus(4,dy dof)=dphi4_dp *( dpig_dun{ip,3}+dpig_dun{ip,4});
251
252
253     % initialize dphi/dus for (fld. elem./stc. elem) with Lagr. mult.
254     dphidus_temp=zeros(4,ntdof);
255
256     % compute dphi/dus for (fld. elem./stc. elem) with Lagr. mult.
257     dphidus_temp(:,(aldofs))=dphidus;
258
259     % get global addresses of the structural element's dofs
260     glbaddr_stc=edof(stc_id,2:end);
261
262     % store dphi/dus for (fld. elem./stc. elem) in the global system
263     dphidus_elem(:,glbaddr_stc)=dphidus_elem(:,glbaddr_stc)+
        dphidus_temp;
264 end
265
266 % store elemental dphidus in global cell array
267 dphidus_all{iglb,1}=dphidus_elem;
268
269 % set value of epsilon for forward and backward perturbation
270 depsilon=1e-8;
271
272 % loop over fluid element nodes: compute dr/dphi of element
273 for in=1:4
274
275     % initialize cell array for storing the perturbed responses
276     res_prtbd=cell(2,1);
277
278     res_prtbd{1,1}=zeros(ndftot,1);
279     res_prtbd{2,1}=zeros(ndftot,1);

```

```

280
281 % initialize flag for forw., backw. and central differencing
282 pert=zeros(2,1);
283
284 % loop over the perturbed states: forward and backward
285 for ipt=1:2
286
287 % get solution vector of element
288 edf=edof(iglb,2:end);
289
290 edt=reshape(fsol(edf,1),1,[]);
291 edn=extract(edof(ie,:),fn);
292 edot=extract(edof(ie,:),fdot);
293
294 % nodal level set values of element
295 levs=ep{iglb,4};
296
297 % perturbate level set value of node
298 switch ipt
299
300 % ipt==1: forward perturbation
301 case 1
302
303 % check if perturbation changes nodal level set sign
304 if sign(levs(in)+depsilon)==sign(levs(in))
305     levs(in)=levs(in)+depsilon;
306     pert(ipt,1)=1;
307
308 else
309     pert(ipt,1)=0;
310 end
311
312 % ipt==2: backward perturbation
313 case 2
314
315 % check if pert. changes nodal level set sign
316 if sign(levs(in)-depsilon)==sign(levs(in))
317     levs(in)=levs(in)-depsilon;
318     pert(ipt,1)=-1;
319
320 else
321     pert(ipt,1)=0;
322 end
323 end
324
325 % copy element properties to apply perturbation
326 ep_prtb=ep;
327
328 % store the perturbation in copy of element properties
329 ep_prtb{iglb,4}=levs;
330
331 % compute nodal volume contribution of element

```

```

332     [res_vol]=xfem4ir(ex(iglb,:),ey(iglb,:),edt,edn,edot,ep_prtb(
           iglb,:));
333
334     % get addresses of residual contributions in global system
335     glbaddr=edof(iglb,2:end);
336
337     % insert residual contributions into global residual
338     revol_glb=zeros(ndftot,1);
339     revol_glb(glbaddr)=revol_glb(glbaddr)+res_vol;
340     res_prtbd{ipt,1}=res_prtbd{ipt,1}+revol_glb;
341
342     % compute necessary interface information for integration
343     [ne,dnex,dney,nestd,wgp,epc,ifcinfo]=alexfem_intinfo(ex(iglb
           ,:),ey(iglb,:),ep_prtb(iglb,:),edt,levs);
344
345     % get edge intersection points
346     edginfo=edginfo_glb{iglb,1};
347     [esct,edpts,edgpba]=alexfem_edgisctFD(ex(iglb,:),ey(iglb,:),
           edginfo,ep_prtb(iglb,:));
348
349     nedc=0;
350
351     % loop over interfaces
352     for ib=1:esct
353
354         % extract edge intersection points
355         edgpt1 = edpts(2*(ib-1)+1,:);
356         edgpt2 = edpts(2*(ib-1)+2,:);
357
358         % compute intersection normal
359         dedg=(edgpt2-edgpt1)';
360         norml=[dedg(2); -dedg(1)];
361
362         % initialize vector of velocity along interface
363         cxvec=1234*ones(10,1);
364         cyvec=1234*ones(10,1);
365
366         % initialize vector of s-values along interface
367         svec =1234*ones(10,1);
368
369         nprc=0;
370
371         % loop over sub-bars
372         for is=1:nsbfld(iglb)
373
374             % get ID of parent bar
375             ids=cell2mat(sbcfld{iglb,is}(1));
376             pbid=ids(1);
377
378             % skip sub-bar if parent bar ID is not in edge
              intersection pba
379             if ~ismember(pbid,edgpba); continue; end
380

```

```

381         % loop over sub-bar Gauss points
382         for ig=1:2
383
384             % get ID of parent structural element of bar
385             stcid=cell2mat(sbcfld{iglb,is}(6));
386
387             % get global coordinates of Gauss point
388             gpts=cell2mat(sbcfld{iglb,is}(1+ig));
389
390             % compute s-value of projected Gauss point along
391             interface
392             A = [dedg norml];
393             r = gpts - edgpt1;
394             s = A\r';
395
396             % check needed on s(1)
397             sval=min([1,max([s(1),0])]);
398
399             % compute local coordinates of Gauss point
400             [xlocfld]=xfem4loc(gpts',ex(iglb,:),ey(iglb,:));
401             xlocstc =xlocfld;
402
403             % compute structural velocity at Gauss point
404             stcnoids=fetopo(stcid,2:5);
405             cx=getSol4i(uxdot(stcnoids),xlocstc');
406             cy=getSol4i(uydot(stcnoids),xlocstc');
407
408             % store projected values
409             nprc=nprc+1;
410             svec(nprc)=sval;
411             cxvec(nprc)=cx;
412             cyvec(nprc)=cy;
413         end
414
415         % extrapolate structural velocities at edge
416         intersection points
417         cxedg=1234*ones(2,1);
418         cyedg=1234*ones(2,1);
419
420         if (nprc==2 && length(unique(svec(1:nprc)))==2) || (
421             nprc==4 && length(unique(svec(1:nprc)))==4)
422
423             nedc=nedc+1;
424             cxedg(nedc)=interp1(svec(1:nprc),cxvec(1:nprc),0,
425                 'linear','extrap');
426             cyedg(nedc)=interp1(svec(1:nprc),cyvec(1:nprc),0,
427                 'linear','extrap');
428
429             nedc=nedc+1;
430             cxedg(nedc)=interp1(svec(1:nprc),cxvec(1:nprc),1,
431                 'linear','extrap');
432             cyedg(nedc)=interp1(svec(1:nprc),cyvec(1:nprc),1,
433                 'linear','extrap');

```

```

427
428         else
429             cxedg=zeros(2,1);
430             cyedg=zeros(2,1);
431         end
432
433     end
434
435     % get Gauss points and weights along fluid element
436     interface
437         xlgploc=ifcinfo{ib,1}{1}{:,2};
438         ylgploc=ifcinfo{ib,1}{2}{:,2};
439
440         wlgp=ifcinfo{ib,1}{3};
441
442     % reorganize Gauss point information
443     gpts=zeros(3,2);
444     gpts(:,1)=xlgploc;
445     gpts(:,2)=ylgploc;
446
447     edgptsloc=cell(2,1);
448
449     % get local coordinates of interface edge points
450     [edgptsloc{1,1}]=xfem4loc(edgpt1',ex(iglb,:),ey(iglb,:));
451     [edgptsloc{2,1}]=xfem4loc(edgpt2',ex(iglb,:),ey(iglb,:));
452
453     % delete empty entries in interface edge velocities
454     vector
455     cxedg(cxedg==1234)=[];
456     cyedg(cyedg==1234)=[];
457
458     % insert interface edge velocities into interface
459     velocity vector
460     cxvec((nprc+1):(nprc+2))=cxedg;
461     cyvec((nprc+1):(nprc+2))=cyedg;
462
463     % delete empty entries in interface velocity vector
464     cxvec(cxvec==1234)=[];
465     cyvec(cyvec==1234)=[];
466
467     % insert s-values of interface edge points into s-values
468     vector
469     svec(nprc+1)=0;
470     svec(nprc+2)=1;
471     svec(svec==1234)=[];
472
473     % interpolate structural velocities at fluid interface
474     Gauss points
475     if norm(cxedg)==0 && norm(cyedg)==0;
476         ifcvelx=zeros(3,1);
477         ifcvely=zeros(3,1);
478     else

```

```

474         [ifcvelx]=alexfem_edgintpol(edgptsloc,gpts,wlgp,svec,
475         cxvec);
476         [ifcvely]=alexfem_edgintpol(edgptsloc,gpts,wlgp,svec,
477         cyvec);
478     end
479     % extract necessary information for the interface
480     % integration
481     epe=ep_prtb{iglb,5};
482     nel=ifcinfo{ib,2}{1};      % line integration shape
483     % functions
484     dnelx=ifcinfo{ib,2}{2};    % x-derivatives of shape
485     % functions
486     dnely=ifcinfo{ib,2}{3};    % y-derivatives of shape
487     % functions
488     nelstd=ifcinfo{ib,4};      % standard shape functions
489     nrml=ifcinfo{ib,5}{:,1};   % interface normal
490     % get addresses and values of necessary dofs
491     [ntdof,numdof,aldofs,fldof,soltfld,~,~,~,~]=
492     alexfem_addofs(ep_prtb(iglb,:),edt);
493     % compute fluid boundary Jacobian and residual
494     % contributions
495     [recal]=alexfem_inscon(soltfld,nrml, ...
496     ne,dnex,dney,wgp,nestd, ...
497     nel,dnelx,dnely,wlgp,nelstd, ...
498     epe(2:end),epc,ifcvelx,ifcvely,1);
499     % store residual contributions on an elemental basis
500     reale=zeros(numdof,1);
501     res_ifc=zeros(ntdof,1);
502     reale(fldof)=recal;      % without Lagrange multipliers
503     res_ifc(aldofs)=reale;  % with Lagrange multipliers
504     reifc_glb=zeros(ndftot,1);
505     % get addresses of residual contributions in global
506     % system
507     glbaddr=edof(iglb,2:end);
508     % insert residual contributions into global residual
509     reifc_glb(glbaddr)=reifc_glb(glbaddr)+res_ifc;
510     res_prtbd{ipt,1}=res_prtbd{ipt,1}+reifc_glb;
511 end
512 end
513 % finite differencing: get delta_X
514 delta_x=(pert(1,1)-pert(2,1))*depsilon;
515

```

```
518     % finite differencing: compute dr/dphi for given node
519     drdphi_elem(:,in)=(res_prtbd{1,1}-res_prtbd{2,1})/delta_x;
520
521     end
522
523     dphidus_elem=-abs(dphidus_elem);
524
525     % post-multiply dr/dphi with dphi/dus of element to get dr/dus
526     drdus_elem=drdphi_elem*dphidus_elem;
527
528     % insert contributions into global system
529     drf_dus=drf_dus+drdus_elem;
530 end
```

Appendix E

Spy-plots for the two elements, two intersections case

The rest of the results of the 2 element FD with both elements intersected:

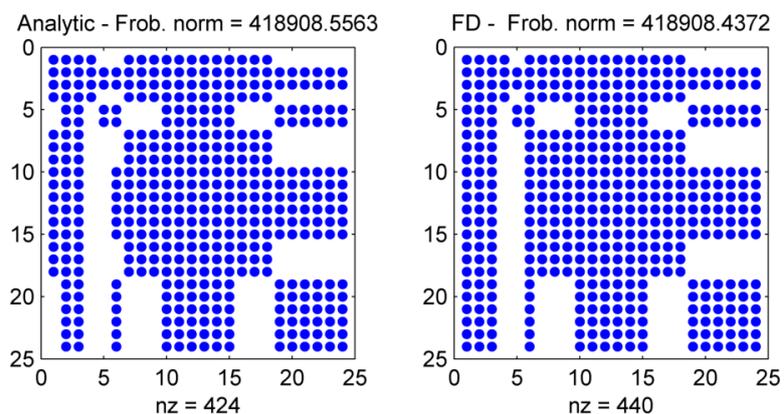


Figure E-1: Two elements, two intersections - Spy plot and Frobenius norm of the complete analytic Jacobian, FD Jacobian

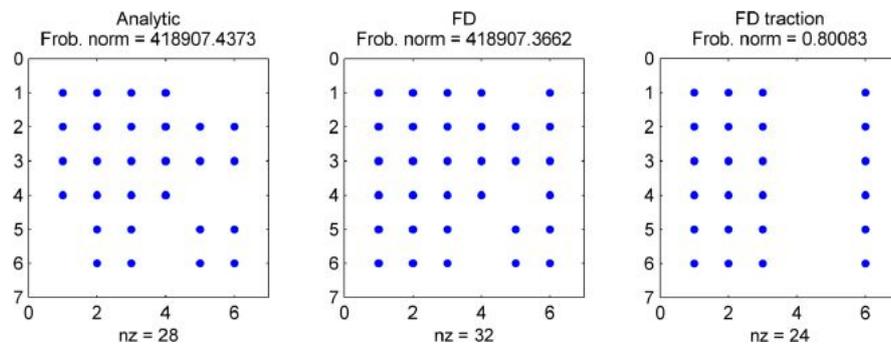


Figure E-2: Two elements, two intersections - Spy plot and Frobenius norm of the analytic structural Jacobian, FD structural Jacobian and FD traction Jacobian

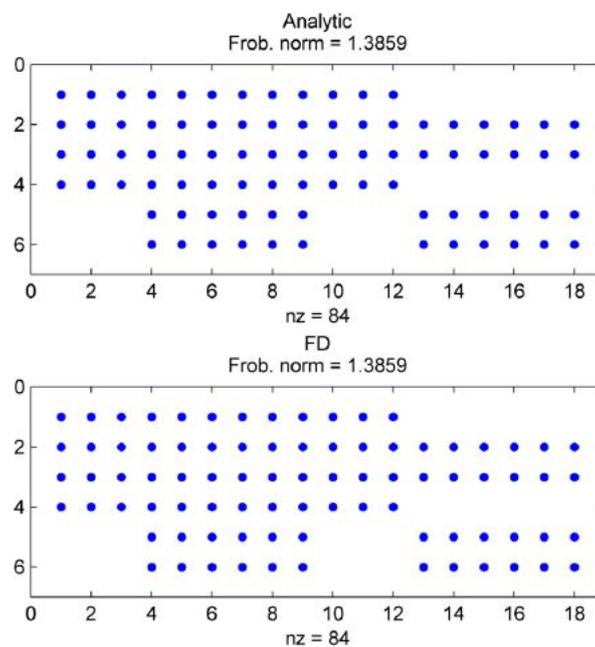


Figure E-3: Two elements, two intersections J_{sf} - Spy plot and Frobenius norm of the analytic structural Jacobian and FD Jacobian

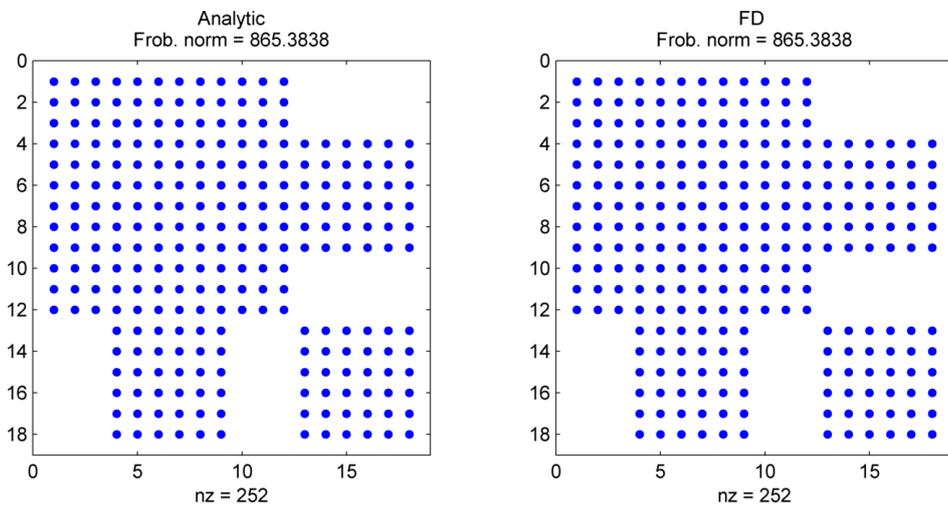
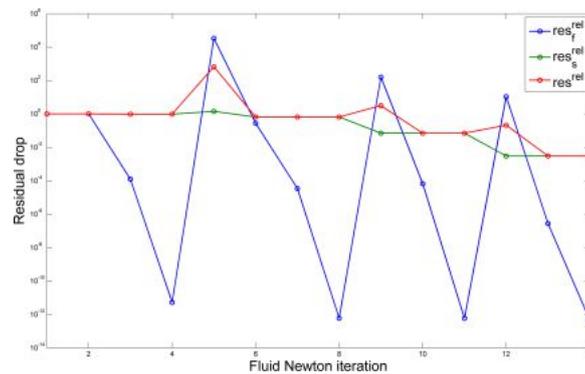


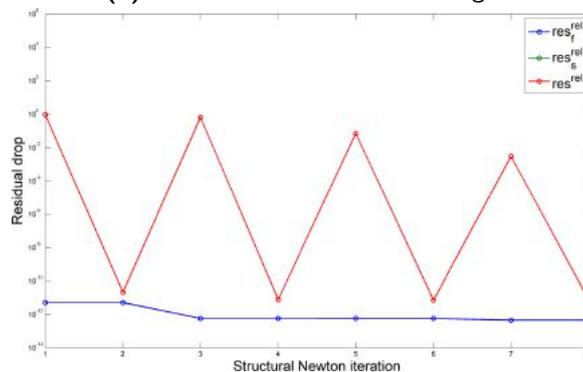
Figure E-4: Two elements, two intersections \mathbf{J}_{ff} - Spy plot and Frobenius norm of the analytic structural Jacobian and FD Jacobian

Relative residual plots - Staggered setup

This appendix contains relative residual plots associated with Figure 4-7:



(a) Relative residuals in fluid stages



(b) Relative residuals in structural stages

Figure F-1: Convergence plots - Relative to the residuals of the solution after the first fluid Newton-Raphson update

Bibliography

- Aloun, D. (2012). Fluid-Structure Interaction with the eXtended Finite Element Method and the Level Set Method. Master's thesis, University of Colorado.
- Belytschko, T. and Black, T. (1999). Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering*, 45(5):601–620.
- Belytschko, T., Gracie, R., and Ventura, G. (2009). A review of extended/generalized finite element methods for material modeling. *Modelling and Simulation in Materials Science and Engineering*, 17(4):043001.
- Donea, J. and Huerta, A. (2004). *Finite Element Methods for Flow Problems*. John Wiley & Sons, Ltd.
- Felippa, C. (2013). Introduction to Finite Elements Methods (ASEN 5007). Lecture Notes. University of Colorado, Boulder, CO, USA.
- Fish, J. and Belytschko, T. (2007). *A first course in Finite Elements*. John Wiley & Sons, Ltd.
- Fries, T.-P. and Belytschko, T. (2010). The extended/generalized finite element method: An overview of the method and its applications. *International Journal for Numerical Methods in Engineering*, 84(3):253–304.
- Fries, T.-P. and Matthies, H. (2005). Meshfree Petrov-Galerkin Methods for the Incompressible Navier-Stokes Equations. In Griebel, M. and Schweitzer, M., editors, *Meshfree Methods for Partial Differential Equations II*, volume 43 of *Lecture Notes in Computational Science and Engineering*, pages 39–54. Springer Berlin Heidelberg.
- Gerstenberger, A. and Wall, W. A. (2008a). An eXtended Finite Element Method/Lagrange multiplier based approach for fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 197(19-20):1699–1714.
- Gerstenberger, A. and Wall, W. A. (2008b). Enhancement of fixed-grid methods towards complex fluid-structure interaction applications. *International Journal for Numerical Methods in Fluids*, 57:1227–1248.

- Gerstenberger, A. and Wall, W. A. (2010). An embedded Dirichlet formulation for 3D continua. *International Journal for Numerical Methods in Engineering*, 82(5):537–563.
- James, K. A. and Martins, J. R. (2012). An isoparametric approach to level set topology optimization using a body-fitted finite-element mesh. *Computers & Structures*, 90-91(0):97–106.
- Kreissl, S. and Maute, K. (2012). Levelset based fluid topology optimization using the extended finite element method. *Structural and Multidisciplinary Optimization*, 46:311–326. 10.1007/s00158-012-0782-8.
- Maute, K., Nikbay, M., and Farhat, C. (2003). Sensitivity analysis and design optimization of three-dimensional non-linear aeroelastic systems by the adjoint method. *International Journal for Numerical Methods in Engineering*, 56(6):911–933.
- Moës, N. and Belytschko, T. (2002). Extended finite element method for cohesive crack growth. *Engineering Fracture Mechanics*, 69(7):813 – 833.
- Osher, S. and Sethian, J. (1988). Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12 – 49.
- Simone, A. (2012). Partition of Unity-based enriched Finite Element Methods: GFEM, PUFEM, XFEM... Lecture notes. Delft University of Technology, Delft, The Netherlands.
- Van Dijk, N., Maute, K., Langelaar, M., and Keulen, F. (2013). Level-set methods for structural topology optimization: a review. *Structural and Multidisciplinary Optimization*, 48(3):437–472.
- Van Keulen, F., Haftka, R., and Kim, N. (2005). Review of options for structural design sensitivity analysis. Part 1: Linear systems. *Computer Methods in Applied Mechanics and Engineering*, 194(30 - 33):3213 – 3243.
- Wall, W., Gerstenberger, A., Gammizter, P., Förster, C., and Ramm, E. (2006). Large Deformation Fluid-Structure Interaction - Advances in ALE Methods and New Fixed Grid Approaches. In Bungartz, H.-J. and Schäfer, M., editors, *Fluid-Structure Interaction*, volume 53 of *Lecture Notes in Computational Science and Engineering*, pages 195–232. Springer Berlin Heidelberg.
- Yoon, G. (2010). Topology optimization for stationary fluid-structure interaction problems using a new monolithic formulation. *International Journal for Numerical Methods in Engineering*, 82(5):591–616.

Glossary

List of Acronyms

FSI	fluid-structure interaction
XFEM	eXtended Finite Element Method
FEM	Finite Element Method
FE	Finite Element
LSM	levelset method
LSF	levelset field
LSFJ	levelset field Jacobian term
NS	Navier-Stokes
PUM	Partition of Unity
GFEM	Generalized Finite Element Method
DOFs	Degrees Of Freedom
FD	Finite Differences
ALE	Arbitrary Lagrangian Eulerian
SUPG	Streamline-Upwind/Petrov Galerkin
PSPG	Pressure-Stabilized/Petrov Galerkin