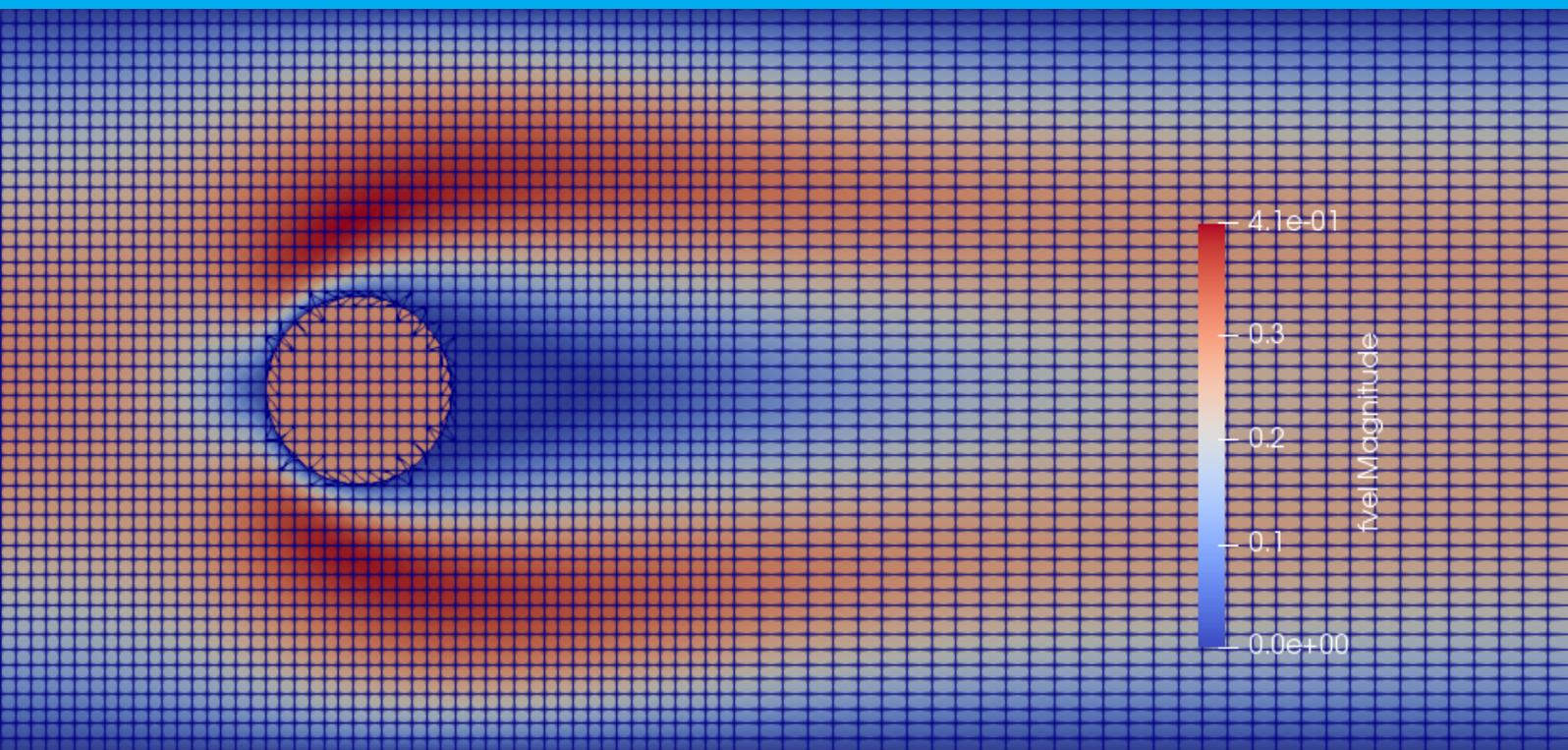


FSI Analysis and Optimization of 3D Beam Structures

A. Mozo Gutierrez

February 9, 2018



FSI Analysis and Optimization of 3D Beam Structures

by

A. Mozo Gutierrez

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday February 9, 2018 at 09:30 AM.

Student number: 4500660
Project duration: March 21, 2017 – February 9, 2018
Thesis committee: Prof. dr. ir. S.R. Turteltaub, TU Delft, Chairman
Dr. ir. M. Pini, TU Delft
Dr. ir. A.H. van Zuijlen, TU Delft

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Abstract

Aiming to enhance the performance of the industrial design process, structural optimization techniques have been proposed as an alternative to traditional design and optimization techniques. They own the potential of achieving an optimal distribution of the material through the design domain, thus reducing waste loss and weight, and increasing the ability to carry the loads and the overall efficiency of the design.

During this thesis work it has been developed a 3D Fluid-Structure Interaction model for beam-like structures, such as wind turbine blades. The cross-section geometry of the beam can be optimized using structural optimization techniques, such as size, shape or topology optimization.

The proposed 3D beam is partially based on the formulation of the classical beam element for slender beams (Euler-Bernoulli), including Saint-Venant torsional effects for isotropic materials, and with the addition of the terms related with the coupling between axial and torsion, and bending and torsion contributions, which may arise when using non-linear materials. The stiffness information of the beam is interpolated from its cross-section geometries and materials, which can vary along the beam length.

The cross-section geometries are defined on a XFEM mesh. The fluid and solid domains are specified using a Level Set Function. This provides a smooth geometry and crisp representation of the solid/fluid interface without the necessity of re-meshing, as in the case of classical FEM. A 2D fluid simulation based on Incompressible Navier Stokes flow at low Reynolds number is carried around each cross-section, in order to obtain the aerodynamic loading over its contour. This aerodynamic loading serves as an input for the beam model, to compute deformation of the beam. This deformation is mapped onto the cross-sections, obtaining the updated displacements and rotations of the geometry. With the updated geometry the fluid field is altered and it needs to be updated as well, forming a non-linear iterative process that loops until a converged structure is obtained.

The 3D FSI model is solved on a monolithic Newton-Raphson solver that treats all the equations involved at once. The Jacobian terms derived for the monolithic solving scheme that has been developed for the forward analysis allow a straightforward computation of the sensitivities using adjoint method. This sensitivity analysis makes possible the optimization of the geometry of the cross-sections based on certain criteria and constraints.

Keywords: **3D BEAM FSI LEVELSET XFEM SENSITIVITIES OPTIMIZATION**

Preface

This Master's Thesis is submitted in fulfillment of the requirements for the Degree of Master of Science in the subject of Aerospace Engineering on the Aerospace Structures and Materials track at Delft University of Technology. I was fortunate to carry this Thesis work at the Center for Aerospace Structures at the University of Colorado Boulder, USA.

I have met many great people over these past months, who I would like to thank for their support and contribution to this work. I am very grateful to Prof.dr.ir. Sergio Turteltaub, professor of the department of Aerospace Structures and Computational Mechanics at the Faculty of Aerospace Engineering, at Delft University of Technology, for supervising and guiding me from the home faculty and making this MSc. Thesis work possible.

I would like to express my sincere appreciation to Prof. Kurt Maute, Palmer Engineering Chair and Director of the Center for Aerospace Structure (CAS) at the Aerospace Engineering Sciences Department at University of Colorado Boulder, for hosting me at his research group and giving me the opportunity to carry this challenging and interesting assignment. I would like to thank for his help to overcome the difficulties that popped up during the project.

I would also like greatly acknowledge the support from MSc. Toshiki Nagai, PhD candidate at the Center for Aerospace Structures at the University of Colorado Boulder, USA, for teaching me the basis of the TransFEM code and his valuable inputs on the Fluid-Structure Interaction analysis. Thank you for the time spent on long debugging sessions.

Furthermore, I would like to thank all my colleagues at both research groups, specially the guys at room ECAE 172 and 188 and the rest of the personnel of the CAS and TU Delft.

With all of them I have learned a lot of knowledge on numerical modeling and with XFEM and LSM, as well as on the structural optimization topic, which have completed the knowledge gained during the courses at TU Delft.

Finally, my deepest gratitude and love to my family and those closest to me, both in my homeland, the Netherlands and in the USA, for their support and understanding and for being an important source of motivation and inspiration to me.

*A. Mozo Gutierrez
Delft, January 24, 2018*

List of Figures

1.1	Example of LS function, Φ , and its update (Top). Material domain, Ω , void domain, $D \notin \Omega$, and interface, Γ , are shown (Bottom) [45].	2
1.2	Example of enriched shape functions for a 2-node element [15].	3
1.3	Example of enriched finite element mesh, showing fully enriched, partially enriched, and unenriched elements[15].	4
1.4	Global problem diagram	5
2.1	Modeling of a wind turbine blade as a discretized beam finite element model of a wind turbine blade.[9]	6
2.2	Modeling of a wind turbine blade as a discretized beam finite element model of a wind turbine blade.[8]	7
2.3	The space 3D beam element. [18]	8
2.4	Local coordinate system and position of the neutral point O , gravity G , and shear center C , for a non-homogeneous cross-section. [36]	9
2.5	2D FEM model for a NACA 0030 airfoil cross-section (a). Detail of the XFEM mesh at the interface elements (b).	12
2.6	Update procedure of the Level Set function for mapping the displacements dues to beam deformation	14
2.7	Representation of the tapered rectangular beam used for verifying the bean model	15
2.8	Detail of the beam clamped end cross-section (a). Detail of the beam free end cross-section (b).	15
3.1	Verification of the FSI analysis. Geometry of the problem [33].	20
3.2	Verification of the FSI analysis. Fluid velocity (a) and fluid pressure (b) distributions around the disk.	21
4.1	Verification of the Monolithic Newton solver. Initial geometry and deformed configuration of the beam.	24
4.2	Different cross-section geometries used along the beam length. (a) Square cross-section (Located at clamped end), (b) Star cross-section (Located at middle of the beam), and (c) Circle cross-section (Located at tip).	24
4.3	Verification of the Monolithic Newton solver. Convergence of the normalized residual with respect to the number of iterations.	25
5.1	Hollowed circular shape used at the cross-sections in order to verify the Sensitivity Analysis.	28
5.2	Position of the cross-section at the middle of the flow channel.	28
5.3	Influence of the perturbation on the computation of the sensitivities for the Adjoint method and global finite difference schemes. Different mesh refinement values (Circles: $1.37e-2$. Squares: $1.03e-2$) are used.	30
5.4	Influence of the mesh refinement on the objective function.	30
5.5	Influence of the mesh refinement on the Sensitivities. 5.5a) Obtained sensitivities from both methods. 5.5b) Relative difference between the obtained sensitivities from both methods	31
5.6	Influence of the Optimization variable on the objective and constraint.	32
5.7	Convergence rate of the norm of the KKT residual for the minimum tip displacement case. Obtained optimum variable for the inner radius. Initial geometry conditions were satisfying the constraint.	33

5.8	Variation of the objective and constraint through the iterations for the minimum tip displacement case. Initial geometry conditions were satisfying the constraint.	33
5.9	Convergence rate of the norm of the KKT residual for the minimum tip displacement case. Obtained optimum variable for the inner radius. Initial geometry conditions were not satisfying the constraint.	34
5.10	Variation of the objective and constraint through the iterations for the minimum tip displacement case. Initial geometry conditions were not satisfying the constraint.	34
5.11	Convergence rate of the norm of the KKT residual for the minimum strain energy case. Obtained optimum variable for the inner radius. Initial geometry conditions were not satisfying the constraint.	35
5.12	Variation of the objective and constraint through the iterations for the minimum strain energy case. Initial geometry conditions were not satisfying the constraint.	36
5.13	Initial Level Set parametrization of a hollowed rectangular cross-section, using 4 optimization variables S_i	37
6.1	Swiss-cheese hollowed pattern black & white bitmap file. Used to define hollowed geometries necessary as an initial guess for the topology optimization procedure.	39
6.2	Simplified idealization of the performance increment when using Newton relaxation stabilization to help to improve convergence under strong oscillatory behaviors. Without (Left) and with (Right) Newton relaxation. In red the expected converged solution.	41
A.1	Example of intersected element. Position of intersected nodes	54
B.1	Evolution of the horizontal velocity distribution and Level Set representation of the section 1 mesh (Clamped)	58
B.2	Evolution through the Newton iterations of the horizontal velocity distribution and Level Set representation of the section 2 mesh (Mid)	61
B.3	Evolution through the Newton iterations of the horizontal velocity distribution and Level Set representation of the section 3 mesh (Tip)	64
C.1	Evolution of the cross-section geometry through the optimization iterations	66

List of Tables

2.1	Main properties of the tapered rectangular beam used for verifying the beam model	15
2.2	Results for the convergence study carried during the verification process of the beam model.	16
3.1	Main properties of Ω_s and Ω_f used for verifying the FSI analysis tool	20
3.2	Results for the convergence study carried during the verification process of the FSI analysis.	21

Contents

List of Figures	iv
List of Tables	vi
1 Introduction	1
1.1 Background	1
1.1.1 Level Set Method	2
1.1.2 eXtended Finite Element Method (XFEM).	3
1.2 Goal	4
1.3 Methodology	4
1.4 Global problem set	4
1.5 Report Layout.	5
2 3D Beam Model	6
2.1 Beam Element Model	7
2.1.1 Coordinate system	8
2.1.2 Constitutive behavior law.	9
2.1.3 Kinematics	9
2.1.4 Stresses and Resulting Internal Forces	10
2.1.5 Beam Element Stiffness Matrix	10
2.1.6 Beam Assembly	11
2.2 Cross-section Finite Element Model.	12
2.3 Cross-section displacement mapping	12
2.4 Verification of the beam model.	14
3 Fluid-Structure Interaction	17
3.1 Description of the problem	17
3.2 Governing Equations	17
3.3 Resultant aerodynamic loads	18
3.4 Verification of the FSI model	20
4 Global System of Equations. Monolithic solver	22
4.1 Strong form of Governing Equations	22
4.2 Weak form of Governing Equations	22
4.3 Verification of the Full Newton Solver	23
5 Optimization	26
5.1 Sensitivity Analysis	26
5.2 Verification of the sensitivities	27
5.2.1 Influence of Finite Difference perturbation size	29
5.2.2 Influence of Mesh Refinement	30
5.3 Results	31
5.3.1 Minimum Tip Displacement	31
5.3.2 Minimum Strain energy.	35
5.3.3 Other cases.	36
6 Conclusions and Recommendations	38
6.1 Discussion	38
6.1.1 Modeling	38
6.1.2 Level Set field.	39
6.1.3 Numerical Instabilities	40

6.2	Conclusions.	42
6.3	Recommendations for Future Research.	43
A	Derivation of Weak Form of Governing Equations	45
A.1	Derivation of Jacobian matrix terms	45
A.1.1	Beam model terms	45
A.1.2	Fluid model terms	46
A.1.3	Level Set projections terms	51
A.1.4	Aerodynamic forces terms	51
B	Verification of the monolithic Newton solver. Visual results	55
B.1	Clamped section: Square shape	56
B.2	Mid section: Star shape	59
B.3	Tip section: Circular shape.	62
C	Optimization of hollowed cylinder beam. Visual results	65
	Bibliography	67
	Glossary	70

Introduction

1.1. Background

Nowadays one of the main challenges in engineering is the development of a cleaner energy system to reduce fossil fuel consumption and CO₂ pollution. In order to do that, several new renewal sources of energy have appeared. Among all the alternatives, wind energy seems to be as one of the most promising in terms of cost and efficiency [1].

In the recent years, this industry has evolved a lot, moving from onshore to offshore farms and turning into larger and more numerous installations. To deal with this evolution and market growth, wind turbine blade designs need to be adapted to the new trends. Major structural changes have consisted on an enlargement of the blade, turning slenderer, while keeping the weight and cost to the minimum [46]. To ensure this, the choice of the cross-section design as well as the employed materials is a topic of major interest. In addition, manufacturing techniques need to provide a cost-effective mean of production for wind turbine blades at a large scale [12].

Current designs are constrained by the actual manufacturing techniques, e.g. vacuum infusion, and by the availability of fiber-reinforced composites. The emerge of new manufacturing techniques, such as 3D printing [3] and additive manufacturing [35], has released the possibility of achieving new innovative wind turbine blade designs, which own the potential of achieving an optimal distribution of the material through the cross-section, thus reducing waste loss and weight, and increasing the ability to carry the loads and the overall efficiency of the design.

Aiming to enhance the performance of these designs, structural optimization techniques have been proposed as an alternative to traditional design and optimization processes. Topology optimization methodology is considered, among all other structural optimization techniques, the one that allows a larger degree of freedom when distributing the material along the design space, i.e. requiring less input information for the initial design guess [7].

Some authors have already applied this methodology to find the optimal material distribution for the cross-section of beam structures (e.g. wind turbine blades), obtaining a novel cross-section topology with a total mass lower than the one from the reference, while fulfilling the stiffness criteria and keeping the structural performance [25, 28, 29].

Classical topology optimization approaches present several issues (E.g.: Numerical instabilities, local optima convergence, non-crispy geometry definition) that prevent the direct step-less application of this methodology from a step-less design to manufacturing [40]. One of the major issues is the apparition of the so called “gray areas”, i.e. the apparition of areas with intermediate density of the material (With no physical meaning) at the boundaries of the design, as a result of the numerical modeling of the problem, thus preventing a clear and crispy definition of the geometry of the design. This well-defined geometry is necessary for the direct application of the obtained design without further time-consuming post-treatment [45].

In the recent years, several approaches have emerged that potentially allow a clear “black

and white” geometry description of the design, i.e. without gray areas, such as the Level Set method joint with additional techniques, such as the eXtended Finite Element Method, XFEM [5, 6], that prevent computationally expensive re-meshing steps during the optimization process, among others [45].

In this research, the behavior of the 3D beam structure is modeled as a Fluid-Structure Interaction (FSI) problem. Fluid structure interaction research field studies the interaction between a movable structures within fluid flows. It has a long range of practical applications in engineering. In this case the FSI modeling of the 3D beam will combine the eXtended Finite Element Method (XFEM), and the Level Set Method (LSM), in order to define the cross-section geometry. This approach has a strong potential on optimization procedures, obtaining of crisp interface between the solid and fluid phases, at a lower computational cost [11, 39, 47, 48].

1.1.1. Level Set Method

On the Level Set approach an implicit description of boundaries is used to parametrize the geometry [2, 38, 49]. The interface between the material and the void phase is defined by the isocontour ($\Phi = \text{Constant}$) of a Level Set function, $\Phi(\mathbf{X})$, which allows a crisp representation of the design boundaries, hence improving the accuracy of the mechanical response captured on the boundary regions of the design and avoiding ambiguities of intermediate density material phases when using classical density-based approaches [45].

The LS function, Φ , defines, inside the design domain, D , the material and void regions, as well as the material interfaces, as stated in [40].

$$\begin{cases} \Phi(\mathbf{X}) > c & \mathbf{X} \in \Omega \text{ (Material)} \\ \Phi(\mathbf{X}) = c & \mathbf{X} \in \Gamma \text{ (Interface)} \\ \Phi(\mathbf{X}) < c & \mathbf{X} \in D \notin \Omega \text{ (Void)} \end{cases}$$

Where \mathbf{X} is a point within D , and c is a constant, generally $c = 0$. Altering the LS function may change the shape and the topology of the material/void domain (Figure 1.1) [45].

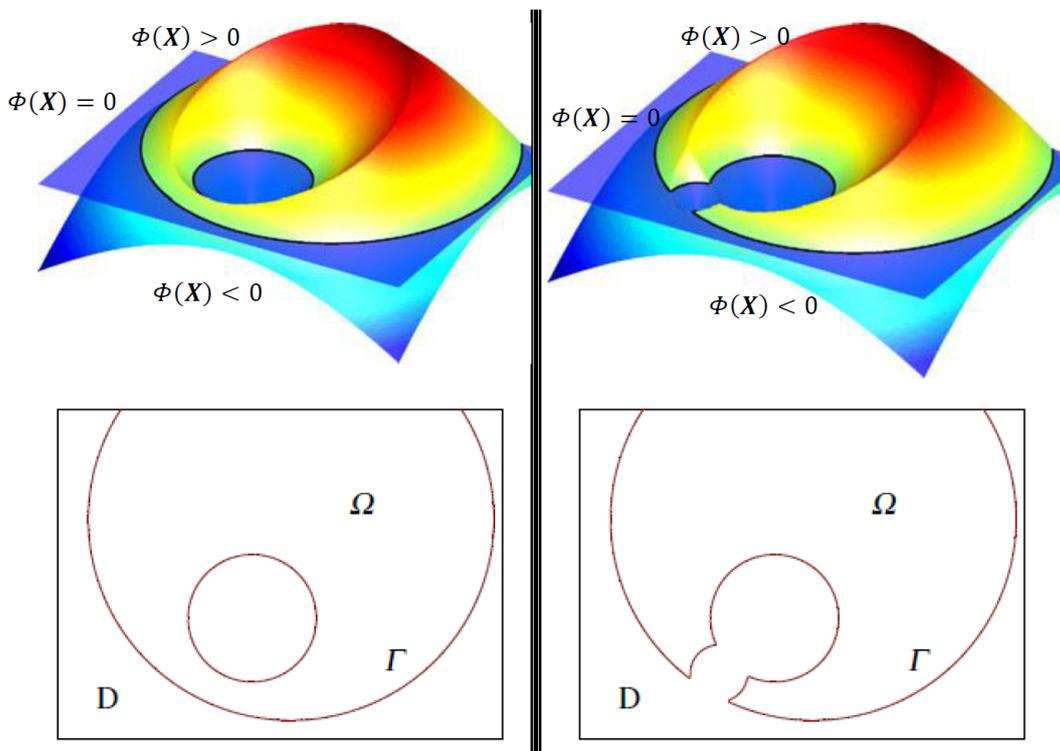


Figure 1.1: Example of LS function, Φ , and its update (Top). Material domain, Ω , void domain, $D \notin \Omega$, and interface, Γ , are shown (Bottom) [45].

1.1.2. eXtended Finite Element Method (XFEM)

X-FEM was first proposed by [5] as a method for minimizing re-meshing steps in crack growth problems, following an evolution of the Partition of Unity Method (PUM) [31]. The classical finite element model was enriched with discontinuous functions that were able to treat the crack growth phenomena regardless the alignment of the crack within the mesh.

The basis of X-FEM is to locally increase the number of degrees of freedom of the FEM shape function with an enrichment discontinuous function, only at the elements placed along the boundary and the discontinuities, in order to model arbitrary displacements and shape changes at the geometry interface. On the rest of the elements no further manipulation is needed, hence the re-meshing process is not necessary any more.

In the classical approach of FEM, the nodal displacement field is given by continuous shape functions (Equation 1.1) [52].

$$\mathbf{u}(\mathbf{x}) = \sum_{i \in I} N_i(\mathbf{x}) u_i \quad (1.1)$$

Where i is the index of the nodes, I is the total number of nodes, u_i corresponds to the displacements at the node and $N_i(\mathbf{x})$ are the shape functions.

In the case of X-FEM, the displacement field is enriched with additional discontinuous shape functions (Equation 1.2)[5, 19, 50].

$$\mathbf{u}(\mathbf{x}) = \sum_{i \in I} N_i(\mathbf{x}) u_i + \sum_{j \in J} \Psi(\mathbf{x}) N_j(\mathbf{x}) a_j \quad (1.2)$$

Where j is the index of the nodes at the interface, J is the total number of nodes at the interface, a_j corresponds to the enriched degrees of freedom at the nodes and $\Psi(\mathbf{x})$ are the enrichment functions. Several enrichment functions have been presented in the literature, such as [15, 17, 41].

An example of enriched shape functions for a 2-node element can be seen on Figure 1.2.

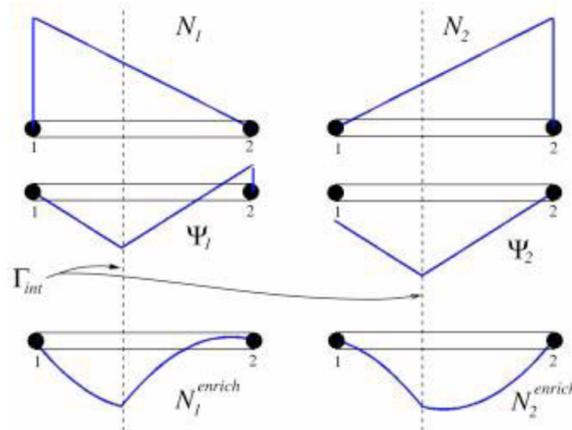


Figure 1.2: Example of enriched shape functions for a 2-node element [15].

The enrichment fully affects the elements cut by the interface, where all nodes are enriched following the partition of unity rule [31]; thus the enrichment function, $\Psi(\mathbf{x})$, is reproduced exactly. Elements that are adjacent to the elements cut by the interface (i.e. partially enriched elements) do not present all the nodes enriched. Hence, in this case the enrichment provides a blending so that the enrichment vanishes at the edge of the support of the bisected nodes (Figure 1.3) [5, 15, 19, 32].

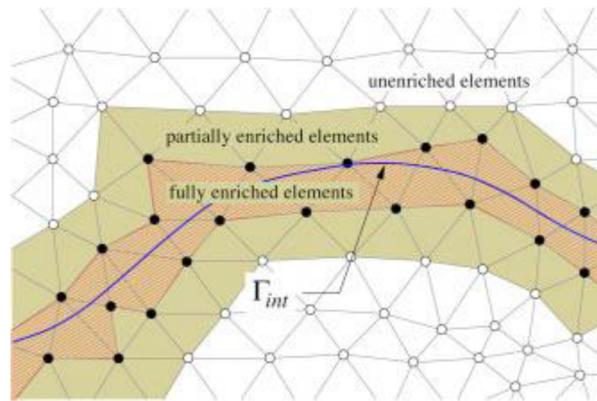


Figure 1.3: Example of enriched finite element mesh, showing fully enriched, partially enriched, and unenriched elements[15].

The enrichment is a local process which results in a sparse matrix system. The displacement field is only affected in the elements crossed by the interface and in the adjacent elements i.e. the ones sharing the enriched nodes (Figure 1.3).

1.2. Goal

The objective of this Master Thesis project is to design a 3D FSI system that combines a 3D beam model with a 2D fluid analysis per section, to reduce computational cost of full 3D FSI problem. In addition, the geometry of the cross-section is to be optimized according to certain design criteria, obtaining a crispy geometry and avoiding re-meshing by the use of Level Set + XFEM methods in a full Eulerian mesh (Immersed boundary approach).

This model could be applicable to beam like structures, such as wind turbines blades, potentially making more efficient their design and analysis.

1.3. Methodology

The basis of this project is provided by the TransFEM code, a FEM meshing and analysis tool, including LSM and XFEM, developed on Matlab by the Aerospace Structures department at CU Boulder, led by prof. K. Maute. An adaption of this code has been used to set up the 3D beam model, computing the cross-section mass and inertia properties, and performing the FSI analysis.

In addition, the research of MSc. T. Nagai [34] has been used for setting up the fluid analysis and adapting the Level Set projection on TransFEM.

The Globally Convergent Method of Moving Asymptotes (GCMMA) Matlab code, based on [42], has been added to TransFEM in order to perform structural optimization.

A Newton-Raphson method has been developed for solving the monolithic non-linear system of equations, in order to find the solution of the forward analysis in an iterative manner.

Furthermore, the adjoint method [26] has been implemented in order to perform the sensitivity analysis necessary for the optimization process. The analytical Jacobians derived for the forward analysis were used, in combination with finite differences procedures [39].

1.4. Global problem set

The turbine blade is modeled as 3D beam. On the one hand, the inertia properties are taken from the cross-sectional analysis. On the other hand, the load input for the beam model is obtained from solving a 2D Fluid-Structure Interaction problem at each cross-section. This done, the nodal displacements of the beam can be computed. This information is then used to calculate the deformation of the beam cross-sections and to perform a new iteration of the Inertia and FSI models, until a certain level of convergence is achieved. The overall procedure is summarized on the diagram below (Figure 1.4).

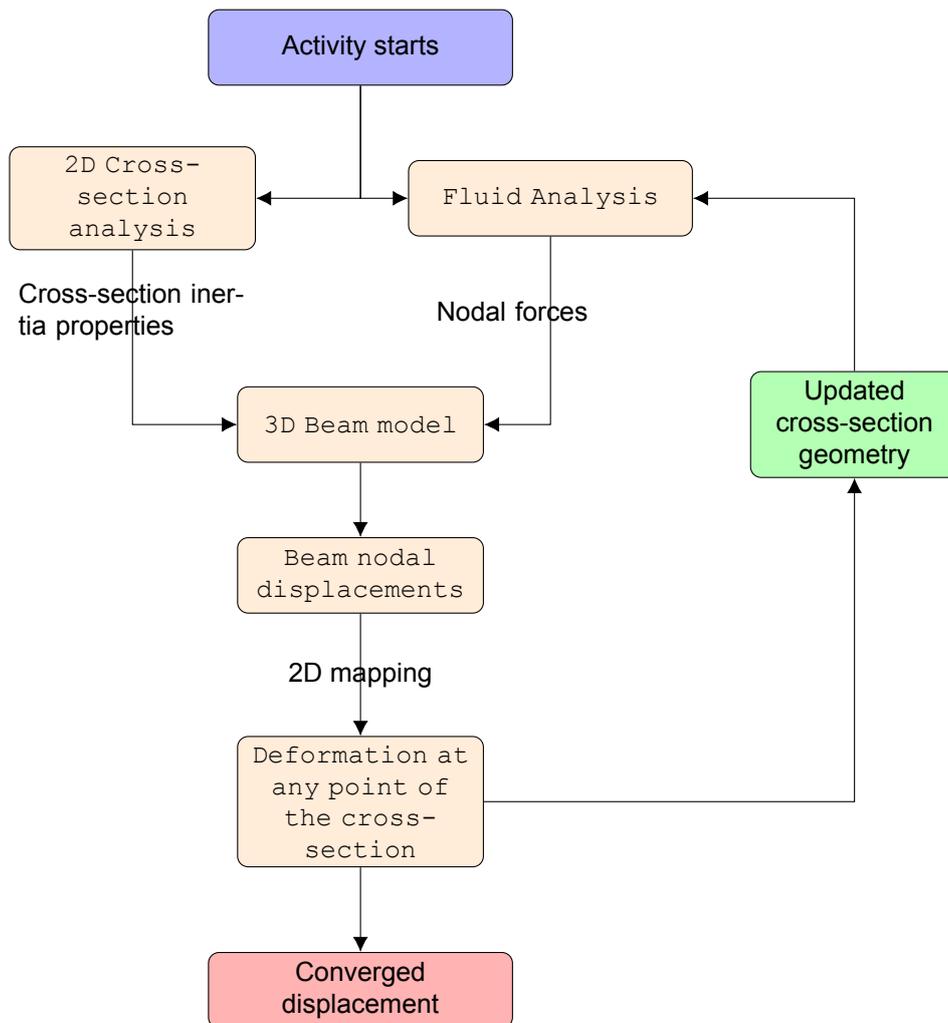


Figure 1.4: Global problem diagram

Once a converged displacement has been obtained, the original geometry of the cross-section can be optimized based on certain criteria, e.g. Minimizing the displacement at the tip of the beam.

1.5. Report Layout

The layout of this MSc Thesis report follows a bottom-up strategy. First, the 3D beam model is presented in section 2. In section 3 the Fluid-Structure Interaction part is explained. In section 4 the Global System of Equations and the Monolithic solver are shown. Section 5 presents the optimization technique. In section 6 it can be seen the conclusions and recommendations. Finally, Appendixes and References are shown at the end of the report.

2

3D Beam Model

To perform a structural analysis, wind turbine blades can be modeled as beams. A beam is a solid structural body that presents slenderness, this means the length of the blade is considerably larger than the cross-sectional dimensions. Due to this reason, the Euler-Bernoulli model can be used; hence neglecting the effect of traverse shear stress on the beam stiffness (Figure 2.1).

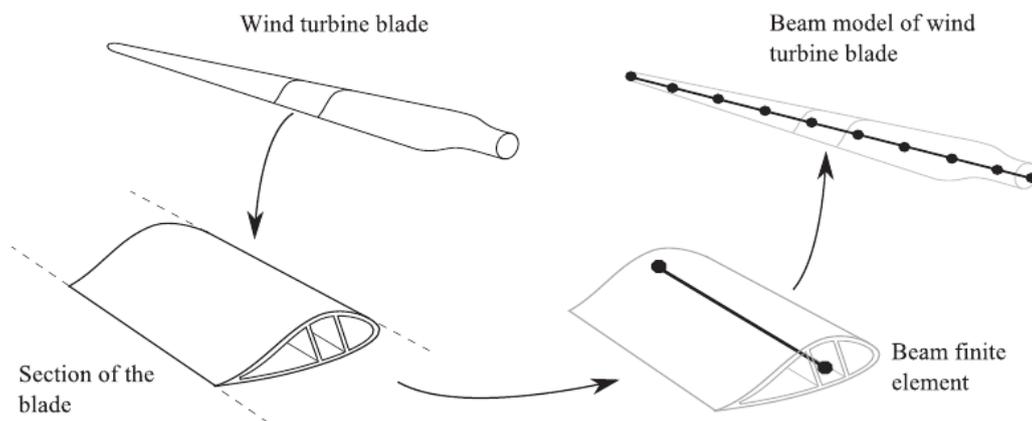


Figure 2.1: Modeling of a wind turbine blade as a discretized beam finite element model of a wind turbine blade.[9]

In order to simplify the problem and reduce the computational cost required for carrying a structural analysis on a full 3D solid model, the beam is split into two parts (Figure 2.2). First a 2D analysis is performed on each of the cross-sections, which carry information about the material, geometry and topology. This done, the information is assembled into a 1D beam element model, which considers the global state of loading and response of the blade. It is assumed that the cross-section properties change smoothly throughout the length of the beam, hence the resulting strain-stress state will also present a moderate variation.

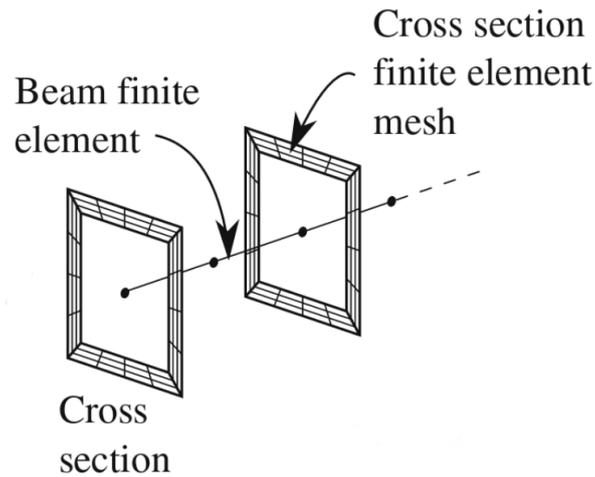


Figure 2.2: Modeling of a wind turbine blade as a discretized beam finite element model of a wind turbine blade.[8]

2.1. Beam Element Model

For simplicity, the blade is considered as a straight beam, where all the beam elements are assembled one followed by each other, with no relative rotations between them. Hence the global coordinate system (x, y, z) is defined. For convenience, y and z global axes are related with the flap-wise and chord-wise directions of the blade, respectively.

The space beam element considered is a prismatic 1D element that 2 nodes placed at each end. The nodes are centered on the position of the centroid of the cross section located at the end of the element (Figure 2.3).

The line between the two nodes defines the local x' axis which superimposes the global x axis of the beam, whereas the local y' and z' axes are parallel to global y and z axes. Inertia properties are defined with respect to these axes.

6 degrees of freedom are allowed at each node i : 3 translations $(u_{x_i}, u_{y_i}, u_{z_i})$, and 3 rotations $(\theta_{x_i}, \theta_{y_i}, \theta_{z_i})$.

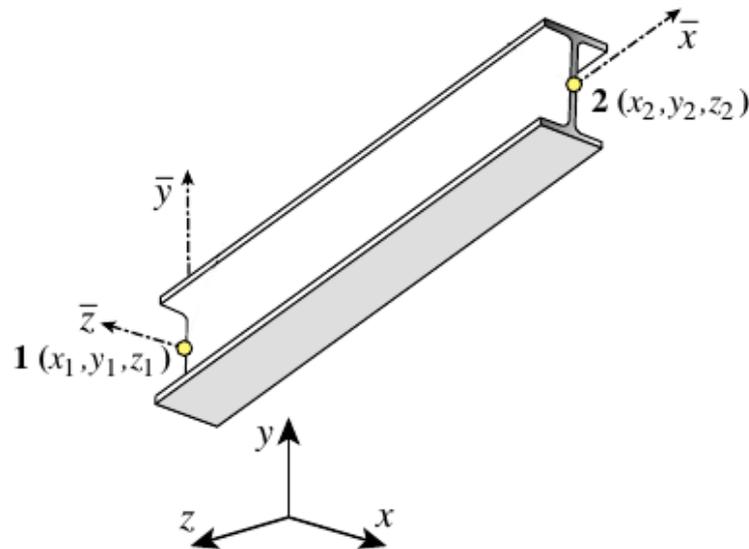


Figure 2.3: The space 3D beam element. [18]

The forces acting on the beam are axial forces f_{x_i} , shearing forces (f_{y_i}, f_{z_i}), bending moments (m_{y_i}, m_{z_i}), and twisting moments (torques) m_{x_i} . Their orientation and positive direction correspond to the positive direction of the axes. The corresponding displacements u_{x_i} will be taken to be positive in the positive directions of the forces.

Beam stiffness properties are determined by the material, geometry, and topology of the cross-sections. The material is characterized by the elastic modulus E and the shear modulus G , and the geometry defines the inertia properties. The length of the beam element is denoted by L .

2.1.1. Coordinate system

In order to determine the beam stiffness properties, it will be used the local coordinate system of the cross-sections (O, x', y', z') (Figure 2.4). For simplicity, it will be assumed that the local x' axis corresponds to the neutral axis of the beam, which allows the decoupling of bending and axial effects. Point O defines the origin of the local system at the cross-section, corresponding to the neutral point. Bending and torsional effects decoupling occurs when the external forces are applied at the shear center C of the section. The line connecting the C point of all section corresponds to the elastic axis. Generally the neutral point O and the shear center C may not coincide with the gravity center of the section G for non-homogeneous sections [36].

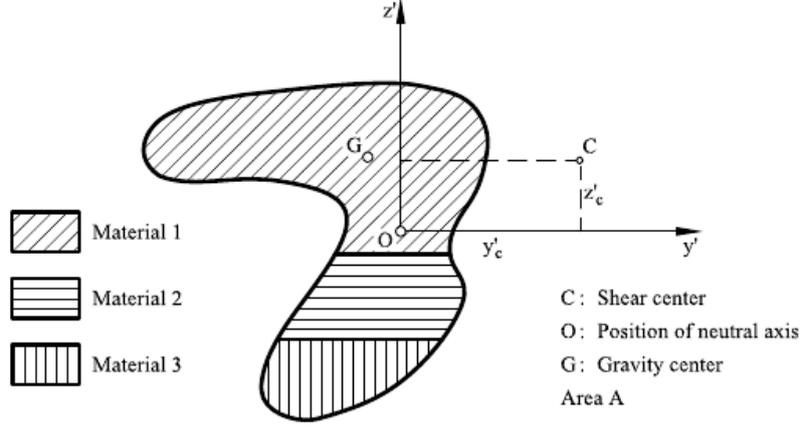


Figure 2.4: Local coordinate system and position of the neutral point O , gravity G , and shear center C , for a non-homogeneous cross-section. [36]

2.1.2. Constitutive behavior law

For linear isotropic materials and transversely isotropic composite materials the constitutive equations relate the local stresses at the cross-section ($\sigma_{x'}$, $\tau_{x'y'}$, $\tau_{x'z'}$) with the conjugate strains ($\epsilon_{x'}$, $\gamma_{x'y'}$, $\gamma_{x'z'}$) (Equation 2.1) [36].

$$\begin{Bmatrix} \sigma_{x'} \\ \tau_{x'y'} \\ \tau_{x'z'} \end{Bmatrix} = \begin{bmatrix} E & 0 & 0 \\ 0 & G_{x'y'} & 0 \\ 0 & 0 & G_{x'z'} \end{bmatrix} \begin{Bmatrix} \epsilon_{x'} \\ \gamma_{x'y'} \\ \gamma_{x'z'} \end{Bmatrix} \quad (2.1)$$

Where E is the elastic modulus and G the transverse shear modulus ($G_{x'y'} = G_{x'z'} = G$ for isotropic material)

The position of the neutral axis needs to be defined on each section. Equation 2.2 gives the coordinates of the neutral point, with respect to an auxiliary coordinate system $(\bar{x}, \bar{y}, \bar{z})$ oriented parallel to the global coordinates system will be used and a centered on a certain point of the global cross-section design domain.

$$\bar{z}_O = \frac{\int \int_{\Omega} E \bar{z} dA}{\int \int_{\Omega} E dA}, \quad \bar{y}_O = \frac{\int \int_{\Omega} E \bar{y} dA}{\int \int_{\Omega} E dA} \quad (2.2)$$

2.1.3. Kinematics

Based on the Timoshenko beam theory and the Saint-Venant principle, the displacement field for a point P at the the cross-section can be expressed as Equation 2.3 [4].

$$\begin{aligned} u_p(x', y', z') &= u_O(x) + z' \theta_y(x) - y' \theta_z(x) + \omega(y', z') \frac{\partial \theta_{x'}}{\partial x'} \\ v_p(x', y', z') &= v'_C - (z' - z'_C) \theta_{x'} \\ w_p(x', y', z') &= w'_C + (y' - y'_C) \theta_{x'} \end{aligned} \quad (2.3)$$

Where $u_O(x)$ is the axial displacement of O , v'_C and w'_C are the coordinates of C , $\omega(y', z')$ is the wrapping function, θ_x is the twist rotation, and θ_y and θ_z are the bending rotations. As local system axes are parallel to global ones, displacements and their derivatives have the same values in both systems.

Considering the slenderness of wind turbine blades, Euler-Bernoulli theory for beams can be applied. This means $\frac{\partial v'_C}{\partial x'} = \theta'_z$ and $\frac{\partial w'_C}{\partial x'} = -\theta'_y$.

The local strain field can be derived from Equation 2.3 as follows (Equation 2.4) [36].

$$\begin{Bmatrix} \epsilon_{x'} \\ \gamma_{x'y'} \\ \gamma_{x'z'} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u'_O}{\partial x'} + z' \frac{\partial \theta_{y'}}{\partial x'} - y' \frac{\partial \theta_{z'}}{\partial x'} \\ \left(\frac{\partial \omega}{\partial y'} - (z' - z'_C) \right) \frac{\partial \theta_{x'}}{\partial x'} \\ \left(\frac{\partial \omega}{\partial z'} + (y' - y'_C) \right) \frac{\partial \theta_{x'}}{\partial x'} \end{Bmatrix} \quad (2.4)$$

2.1.4. Stresses and Resulting Internal Forces

The resultant internal forces vector is defined as follows (Equation 2.5). Based on the Bernoulli-Euler hypothesis, the effect of transverse shear on the beam stiffness has been neglected.

$$\begin{Bmatrix} N \\ M_y \\ M_z \\ M_x \end{Bmatrix} = \int \int_{\Omega} \begin{Bmatrix} \sigma_x \\ z' \sigma_x \\ -y' \sigma_x \\ (y' - y'_c) \tau_{x'z'} - (z' - z'_c) \tau_{x'y'} \end{Bmatrix} dA \quad (2.5)$$

By the application of the constitutive relation (Equation 2.1) and the strain-displacement relation (Equation 2.4) to Equation 2.5, the generalized constitutive matrix is obtained (Equation 2.6) [4].

$$\begin{Bmatrix} N \\ M_z \\ M_y \\ M_x \end{Bmatrix} = \int \int_{\Omega} \begin{bmatrix} E & 0 & 0 & 0 \\ 0 & Ez'^2 & -Ey'z' & 0 \\ 0 & -Ey'z' & Ez'^2 & 0 \\ 0 & 0 & 0 & G(y'^2 + z'^2) \end{bmatrix} dA \begin{Bmatrix} \frac{\partial u'_0}{\partial x'} \\ \frac{\partial \theta'_{z'}}{\partial x'} \\ \frac{\partial \theta'_{y'}}{\partial x'} \\ \frac{\partial \theta'_{x'}}{\partial x'} \end{Bmatrix} \quad (2.6)$$

$$\hat{\mathbf{n}} = \mathbf{D} \hat{\boldsymbol{\epsilon}}$$

2.1.5. Beam Element Stiffness Matrix

In Equation 2.12 the axial, bending and torsion effects are uncoupled between them. The only coupling behavior appears between both bending directions, as non principal axes are used.

The neutral line of the beam is discretized into a certain number of 1D straight beam elements. The global nodal displacement and load vectors, can be defined as follows (Equation 2.7).

$$\begin{aligned} \mathbf{u}^i &= [u_{x'_i}, u_{y'_i}, u_{z'_i}, \theta_{x'_i}, \theta_{y'_i}, \theta_{z'_i}]^T \\ \mathbf{f}^i &= [f_{x'_i}, f_{y'_i}, f_{z'_i}, m_{x'_i}, m_{y'_i}, m_{z'_i}]^T \end{aligned} \quad (2.7)$$

The displacements of any point located at the reference line of the element can be expressed as a function of the nodal displacements by the use of a C^0 continuous isoparametric linear interpolation for the axial displacement u'_0 and the the twist rotation θ'_x , and a C^1 continuous cubic Hermite interpolation for v'_c and w'_c (Equation 2.8).

$$\mathbf{u}' = \begin{Bmatrix} u'_0 \\ v'_c \\ w'_c \\ \theta'_x \end{Bmatrix} = \begin{bmatrix} N_i & 0 & 0 & 0 & 0 & 0 \\ 0 & N_i^H & 0 & 0 & 0 & \bar{N}_i^H \\ 0 & 0 & N_i^H & 0 & \bar{N}_i^H & 0 \\ 0 & 0 & 0 & N_i & 0 & 0 \end{bmatrix} \begin{Bmatrix} u'_{0i} \\ v'_{ci} \\ w'_{ci} \\ \theta'_{xi} \\ \theta'_{yi} \\ \theta'_{zi} \end{Bmatrix} = \sum_{i=1}^{n_n} \mathbf{N}_i \mathbf{u}_i \quad (2.8)$$

Where the beam element shape functions associated to each node can be seen on Equation 2.9 for $\xi = \frac{2}{L}x' - 1$ [36].

$$\begin{aligned} N_1(\xi) &= \frac{1}{2}(1 - \xi) \\ N_2(\xi) &= \frac{1}{2}(1 + \xi) \\ N_1^H &= \frac{1}{4}(2 - 3\xi + \xi^3) \\ \bar{N}_1^H &= \frac{1}{4}(1 - \xi - \xi^2 + \xi^3) \end{aligned} \quad (2.9)$$

$$N_2^H = \frac{1}{4}(2 + 3\xi - \xi^3)$$

$$\overline{N}_2^H = \frac{1}{4}(-1 - \xi + \xi^2 + \xi^3)$$

Hence the generalized strain vector $\hat{\boldsymbol{\epsilon}}$ can be expressed as Equation 2.10.

$$\hat{\boldsymbol{\epsilon}} = \sum_{i=1}^{n_e} \mathbf{B}_i \mathbf{u}^i \quad (2.10)$$

Where \mathbf{B}_i is defined as Equation 2.11.

$$\mathbf{B}_i = \begin{bmatrix} \frac{\partial N_i}{\partial x'} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial^2 N_i^H}{\partial x'^2} & 0 & -\frac{\partial^2 \overline{N}_i^H}{\partial x'^2} \frac{L}{2} & 0 \\ 0 & \frac{\partial^2 N_i^H}{\partial x'^2} & 0 & 0 & 0 & \frac{\partial^2 \overline{N}_i^H}{\partial x'^2} \frac{L}{2} \\ 0 & 0 & 0 & \frac{\partial N_i}{\partial x'} & 0 & 0 \end{bmatrix} \quad (2.11)$$

The principle of the virtual work can be expressed as Equation 2.12.

$$\delta W = \int_0^L \delta \hat{\boldsymbol{\epsilon}}^T \hat{\mathbf{n}} dx' = \int_0^L \delta \hat{\boldsymbol{\epsilon}}^T \mathbf{D} \hat{\boldsymbol{\epsilon}} dx' = \int_0^L \delta \mathbf{u}^{iT} \mathbf{B}_i^T \mathbf{D} \mathbf{B}_j \mathbf{u}^j dx' \quad (2.12)$$

Where $\mathbf{K}_{ij}^e = \int_0^L \mathbf{B}_i^T \mathbf{D} \mathbf{B}_j dx'$ is the stiffness matrix of the beam element. In order to deal with changes in the cross-section properties along the beam, a C^0 continuous isoparametric linear interpolation of the \mathbf{D} matrix within each beam element will be performed (Equation 2.13).

$$\mathbf{D}(\xi) = [N_1(\xi) \quad N_2(\xi)] \begin{bmatrix} \mathbf{D}^1 \\ \mathbf{D}^2 \end{bmatrix} \quad (2.13)$$

Where \mathbf{D}^1 and \mathbf{D}^2 are the generalized constitutive matrices of the cross-sections located at nodes 1 and 2 respectively.

The integration is performed using numerical integration with $n_{GP} = 2$ Gauss points ($\xi_g = \pm 0.5773502692$ and $W_g = 1$) (Equation 2.14).

$$\mathbf{K}_{ij}^e = \int_0^L \mathbf{B}_i^T \mathbf{D} \mathbf{B}_j dx' = \int_{-1}^1 \mathbf{B}_i^T(\xi) \mathbf{D}(\xi) \mathbf{B}_j(\xi) J d\xi = \sum_{g=1}^{n_{GP}} \mathbf{B}_i^T(\xi_g) \mathbf{D}(\xi_g) \mathbf{B}_j(\xi_g) J W_g \quad (2.14)$$

2.1.6. Beam Assembly

As the blade is modeled as a long straight beam, the assembly of the stiffness matrices can be easily done without rotating the element matrices, as follows (Equation 2.15).

$$\mathbf{K}^{glob} = \sum_i^{n_e} \mathbf{K}_{ij}^e = \begin{bmatrix} \mathbf{K}_{11}^1 & \mathbf{K}_{12}^1 & 0 & \vdots & 0 & 0 \\ \mathbf{K}_{12}^1 & \mathbf{K}_{22}^1 + \mathbf{K}_{11}^2 & \mathbf{K}_{12}^2 & \vdots & 0 & 0 \\ 0 & \mathbf{K}_{12}^2 & \mathbf{K}_{22}^2 + \mathbf{K}_{11}^3 & \vdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \vdots & \mathbf{K}_{22}^{n_e-1} + \mathbf{K}_{11}^{n_e} & \mathbf{K}_{12}^{n_e} \\ 0 & 0 & 0 & \vdots & \mathbf{K}_{12}^{n_e} & \mathbf{K}_{22}^{n_e} \end{bmatrix} \quad (2.15)$$

This global stiffness matrix relates the global nodal loads and displacements of the beam (Equation 2.16).

$$\begin{bmatrix} \mathbf{f}^1 \\ \vdots \\ \mathbf{f}^{n_n} \end{bmatrix} = \mathbf{K}^{glob} \begin{bmatrix} \mathbf{u}^1 \\ \vdots \\ \mathbf{u}^{n_n} \end{bmatrix} \quad (2.16)$$

2.2. Cross-section Finite Element Model

In order to obtain the inertia properties of each cross-section, and to compute the \mathbf{D} matrix a 2D finite element model has been employed (Figure 2.5a). A crispy description of the geometry is obtained by the use of X-FEM techniques, avoiding the need of re-meshing to increase the convergence of the results.

The mesh is made of regular squared elements made of 4 nodes, owning 2 degrees of freedom per node (In plane displacements). In addition, those elements located at the interface of the geometry present 2 extended degrees of freedom per enrichment level, with a total of 5 enrichment levels. 1 interface is allowed per element (Figure 2.5b).

The geometry of the cross-section is defined on the mesh by the use of a Level Set function Φ . The region where $\Phi > 0$ is considered as the geometric shape and topology of the cross-section.

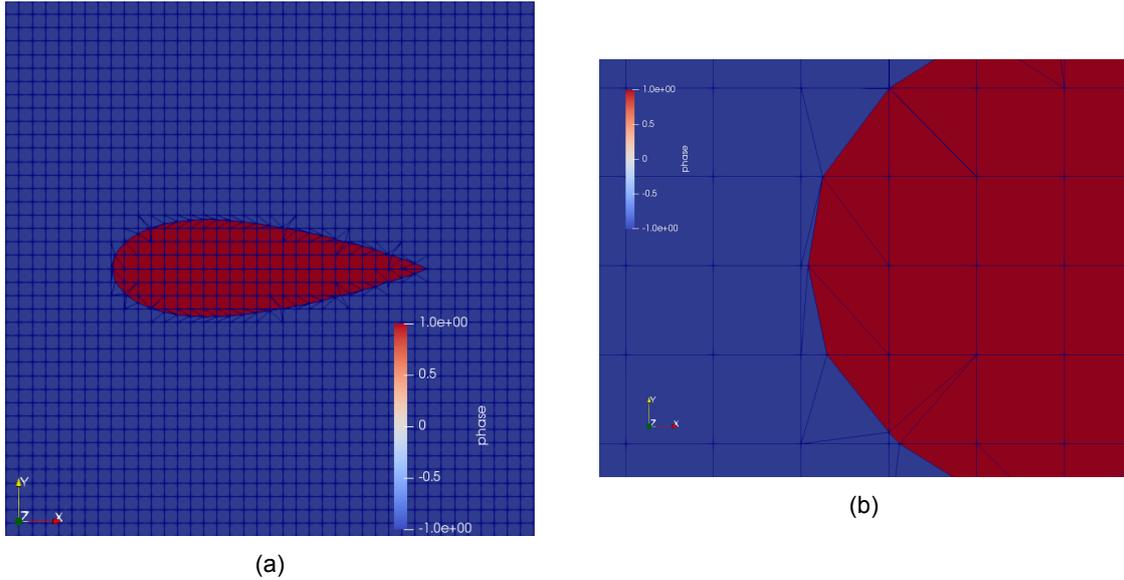


Figure 2.5: 2D FEM model for a NACA 0030 airfoil cross-section (a). Detail of the XFEM mesh at the interface elements (b).

By the use of the position and weight of the Gauss Points of each element, the computation of \mathbf{D} matrix is done (Equation 2.17).

$$\mathbf{D} = \int \int_{\Omega} \begin{bmatrix} E & 0 & 0 & 0 \\ 0 & Ez'^2 & -Ey'z' & 0 \\ 0 & -Ey'z' & Ez'^2 & 0 \\ 0 & 0 & 0 & G(y'^2 + z'^2) \end{bmatrix} dA = \sum_{g=1}^{n_{GP}} \begin{bmatrix} E & 0 & 0 & 0 \\ 0 & Ez_g'^2 & -Ey_g'z_g' & 0 \\ 0 & -Ey_g'z_g' & Ez_g'^2 & 0 \\ 0 & 0 & 0 & G(y_g'^2 + z_g'^2) \end{bmatrix} W_g \quad (2.17)$$

2.3. Cross-section displacement mapping

By solving the equation system (2.16), the nodal displacements of the beam are obtained. This nodal displacements correspond with the solid rigid displacements (And rotations) of the centroid of each cross-section. By mapping these displacements onto the cross-sections, the displacement of any point of the cross-section are derived (equation 2.18) [10]. This information is used for updating the geometry of the cross-section for the upcoming iteration steps.

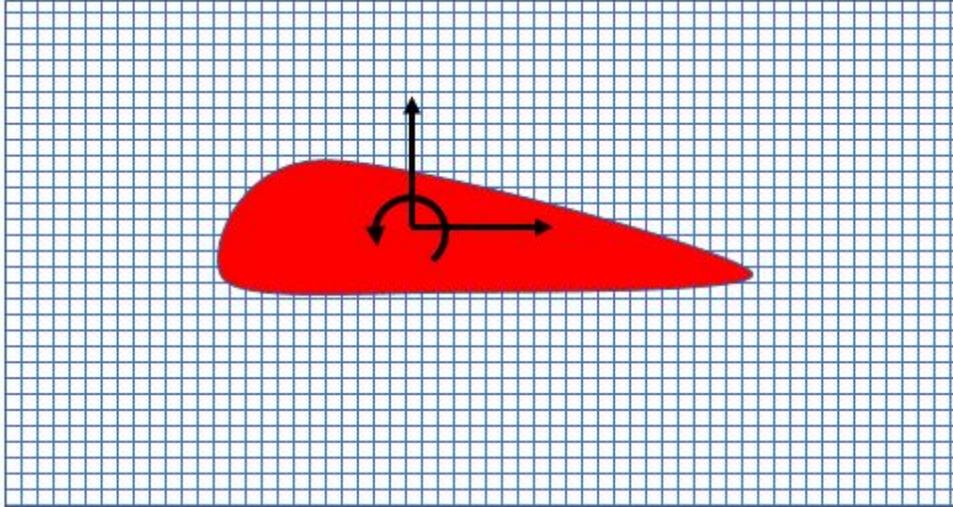
$$\begin{cases} \Delta y^e \\ \Delta z^e \end{cases} = \begin{bmatrix} 0 & 1 & 0 & -z'^e & 0 & 0 \\ 0 & 0 & 1 & y'^e & 0 & 0 \end{bmatrix} \begin{cases} u_{x'_i} \\ u_{y'_i} \\ u_{z'_i} \\ \theta_{x'_i} \\ \theta_{y'_i} \\ \theta_{z'_i} \end{cases} \quad (2.18)$$

$$\Delta \mathbf{x} = \mathbf{Z}(\mathbf{x}) \mathbf{u}_{sec}^l$$

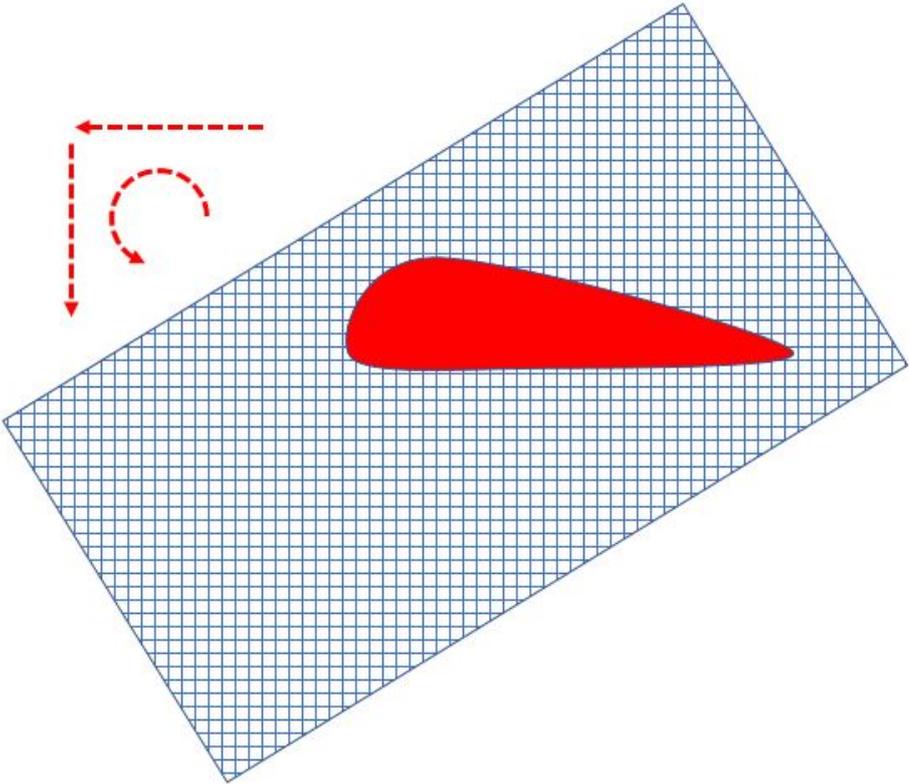
Where \mathbf{Z} is the mapping matrix for translation and rotation of the cross-section. As the cross-section deforms only as a rigid body (Translation+rotation), the Level Set function that represents the geometry can be updated as follows (Equation 2.19). First, the mesh is displaced and rotated mirror-wise to the displacements calculated on Equation 2.18. Then, the Level Set function is mapped on this deformed mesh. Finally, the nodes of the original mesh take the values of Φ from their equivalent nodes of the deformed mesh. The process is graphically explained on Figure 2.6.

$$\Phi(\mathbf{x}) = \Phi_0(\mathbf{x}') \quad (2.19)$$

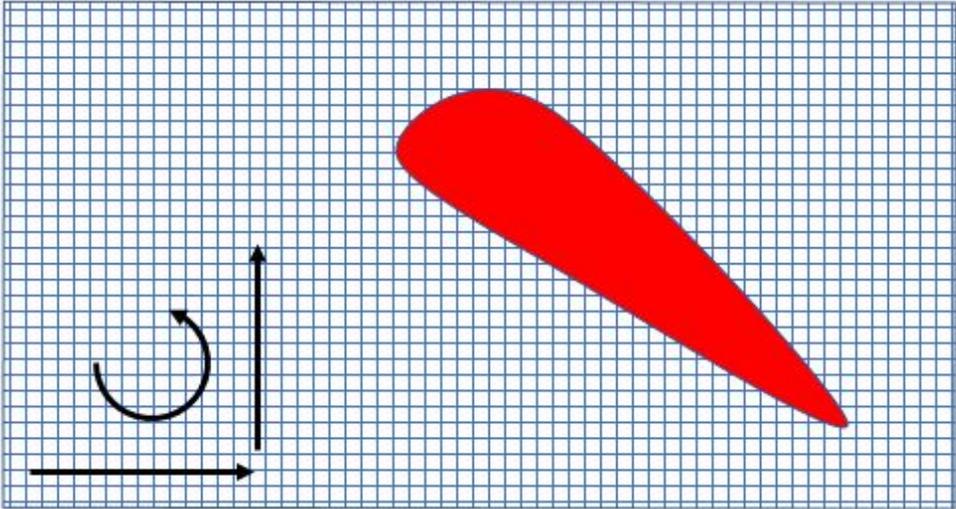
$$\mathbf{x}' = \mathbf{x} - \Delta \mathbf{x}$$



(a)



(b)



(c)

Figure 2.6: Update procedure of the Level Set function for mapping the displacements dues to beam deformation

2.4. Verification of the beam model

In order to verify the beam model, a simulation of a tapered rectangular beam has been carried (Figure 2.7). The beam is clamped at one end, and a vertical point load is acting at the free tip. The properties of the beam can be seen in Table 2.1.

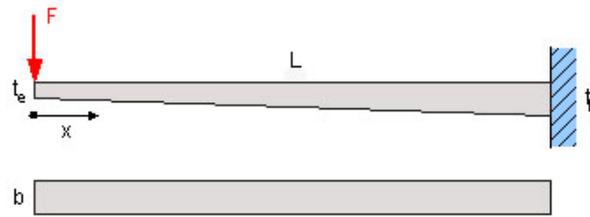


Figure 2.7: Representation of the tapered rectangular beam used for verifying the beam model

Applied vertical force [N]	10
Young Modulus [MPa]	69000
Width [mm]	0.4
Thickness at free end [mm]	0.2
Thickness at clamped end [mm]	0.4
Length [mm]	20
Cross-section mesh refinement	50×50

Table 2.1: Main properties of the tapered rectangular beam used for verifying the beam model

This problem can be analytically calculated. In order to test the 3D capabilities of the model, the cross-sections have been rotated 45° , and the applied load has been split into two components ($F_y = 10 \cos(\pi/4)$ and $F_z = -10 \sin(\pi/4)$) (Figure 2.8). Hence the displacement is also split into two components.

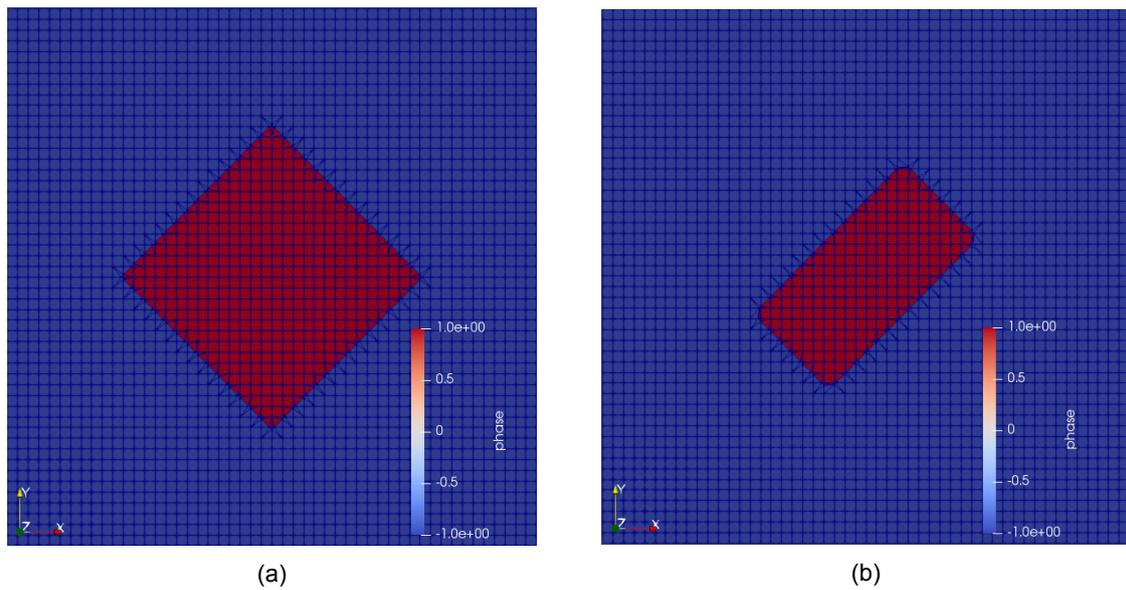


Figure 2.8: Detail of the beam clamped end cross-section (a). Detail of the beam free end cross-section (b).

A summary of the results from the convergence study can be seen on Table 2.2.

Mesh refinement	Tip deflection [<i>mm</i>]	Slope at beam end [<i>rad</i>]
1 beam element	242.4	-22.04
5 beam elements	288.3	-26.41
10 beam elements	291.5	-26.71
20 beam elements	292	-26.77
Analytical solution	296.3	-27.17

Table 2.2: Results for the convergence study carried during the verification process of the beam model.

Limitations of the Level Set Method and XFEM to reproduce the geometry in a more accurate way, joint with the interpolation scheme and model limitations may explain the fact that the convergence rate is not very fast.

3

Fluid-Structure Interaction

Fluid-Structure Interaction (FSI) techniques are employed at every cross-section in order to obtain the aerodynamic load resultants to be used as an input for the 1D beam model. FSI analysis considers simultaneously the perturbation of the flow caused by solid structures immersed within a fluid as well as the forces acting on the solid structures caused by the fluid flow. In the case of this project it will be considered every blade cross-section as a 2D solid rigid body immersed on a 2D fluid flow, that applies a certain pressure around the boundary of the geometry.

3.1. Description of the problem

In order to describe the physical properties, an Eulerian description is used. In this case physical properties are defined on fixed points within the mesh, and this mesh does not deform during the update procedure. Both fluid (Ω_F) and solid phases (Ω_S) need to be defined over this mesh. This Eulerian description is opposite to Lagrangian description, traditionally used in classical solid mechanics problems, where the physical properties are defined on the material points by the use of the so-called body-fitted meshes.

Eulerian FSI method is the one considered for carrying optimization problems as it can handle large deformation and contact phenomena. In addition, Eulerian FSI method is not restricted by the breakdown of body-fitted mesh and can compute more accurate deformation at every optimization step. Furthermore it allows the use of a strong coupling monolithic scheme to solve the problem, computing both phases phenomena at each step, hence increasing the accuracy and stability.

The interface between the solid and the fluid Γ_{SF} is defined by the Interface Capturing method. Based on the level set method, the interface is computed by the advection of the Level Set field Φ , allowing a clearer interface representation.

3.2. Governing Equations

The propagation of the advective velocity of the Level Set function Φ through both solid and fluid phases can be expressed as Equation 3.1 and Equation 3.2.

$$\frac{\partial \Phi}{\partial t} + v_s^i \frac{\partial \Phi}{\partial x_i} = 0 \quad \text{in } \Omega_S \quad (3.1)$$

$$\frac{\partial \Phi}{\partial t} + v_f^i \frac{\partial \Phi}{\partial x_i} = 0 \quad \text{in } \Omega_F \quad (3.2)$$

Where v^s and v^f are the solid and fluid propagation velocities, respectively. Fluid velocities are usually faster than solid velocities. Eventually leakage of the LSF may appear in the case this difference is wide enough. To prevent this phenomena, an artificial advective velocity (Equation 3.5) may be introduced in the fluid phase, using Helmholtz's smoothing (Equation 3.3) and Nietzsche's method techniques (Equation 3.4) [20].

$$-\Delta \hat{v}_i^f + \hat{v}_i^f = 0 \quad \text{in } \Omega_F \quad (3.3)$$

$$v_i^s - \hat{v}_i^f = 0 \quad \text{in } \Gamma_{SF} \quad (3.4)$$

$$\frac{\partial \Phi}{\partial t} + \hat{v}_f^i \frac{\partial \Phi}{\partial x_i} = 0 \quad \text{in } \Omega_F \quad (3.5)$$

Assuming incompressible material for Ω_S and an incompressible Navier-Stokes fluid for Ω_F , the strong form of the governing equations take the following shape for the full-Eulerian FSI problem (Equations 3.6 to 3.14).

$$\text{Solid momentum equation:} \quad \frac{\partial v_i^s}{\partial t} + v_j^s \frac{\partial v_i^s}{\partial x_j} + \alpha v_i^s = \frac{1}{\rho^s} \frac{\partial \sigma_{ij}^s}{\partial x_j} + b_i^s \quad \text{in } \Omega_S \quad (3.6)$$

$$\text{Solid continuity equation:} \quad \frac{\partial v_i^s}{\partial x_i} = 0 \quad \text{in } \Omega_S \quad (3.7)$$

$$\text{Advection of displacement:} \quad \frac{\partial u_i^s}{\partial t} + v_j^s \frac{\partial u_i^s}{\partial x_j} = v_i^s \quad \text{in } \Omega_S \quad (3.8)$$

$$\text{Fluid momentum equation:} \quad \frac{\partial v_i^f}{\partial t} + v_j^f \frac{\partial v_i^f}{\partial x_j} = \frac{1}{\rho^f} \frac{\partial \sigma_{ij}^f}{\partial x_j} + b_i^f \quad \text{in } \Omega_F \quad (3.9)$$

$$\text{Fluid continuity equation:} \quad \frac{\partial v_i^f}{\partial x_i} = 0 \quad \text{in } \Omega_F \quad (3.10)$$

$$\text{Helmholtz smoothing for } \hat{v}^f: \quad -\Delta \hat{v}_i^f + \hat{v}_i^f = 0 \quad \text{in } \Omega_F \quad (3.11)$$

$$\text{Advection of } \Phi: \quad \frac{\partial \Phi}{\partial t} + v_s^i \frac{\partial \Phi}{\partial x_i} = 0 \quad \text{in } \Omega_S \quad (3.12)$$

$$\frac{\partial \Phi}{\partial t} + \hat{v}_f^i \frac{\partial \Phi}{\partial x_i} = 0 \quad \text{in } \Omega_S \quad (3.13)$$

$$\text{Nitsche's method:} \quad v_i^s - \hat{v}_i^f = 0 \quad \text{in } \Gamma_{SF} \quad (3.14)$$

The Cauchy momentum equation (Equations 3.6 and 3.10) governs velocities in both phases Ω_S and Ω_F . The main difference of these momentum equations is the definition of the stress tensor. Stress tensor for the solid phase σ_{ij}^s is defined by displacements u_s and solid pressure p_s in Ω_S . Stress tensor for the fluid phase σ_{ij}^f is defined by fluid velocities v_f and pressure p_f in Ω_F . In addition, the artificial viscous term αv_i is introduced in Ω_S to get equilibrium deformation and α has been set to 1 in this case. Gravitational forces are applied only to the solid phase Ω_S . As the displacement fields are defined only in Ω_S , the advection equation of the displacement is performed only in Ω_S . The continuity equation 3.7 is essential to satisfy the compressibility. The level set function Φ is defined in the entire domain $\Omega = \Omega_S \cup \Omega_F$. In addition, the reinitialization technique is applied to Φ at certain regular time steps. The discretized form of the governing equations, and their derivation, can be found on [21, 22, 30, 43, 44], including SUPG and PSPG stabilization, and Nietzsche's Method penalization terms. Furthermore, ghost penalty factors are applied to suppress the instabilities caused by the intersection of XFEM meshes.

In the turbine blade model we are treating the cross-sections as solid rigid bodies for the FSI analysis. Hence the advective solid velocity vanishes and only Equations 3.9 and 3.10 need to be considered.

This FSI analysis is carried using the code implemented by Toshiki Nagai [34].

3.3. Resultant aerodynamic loads

As a result of the FSI analysis, the pressure distribution p and velocity distribution \vec{v} are obtained along the fluid mesh. By integrating the pressure distribution and the viscous friction along the cross-section outer boundary the resultant aerodynamic load \vec{F} can be

obtained. The aerodynamic load can be split into an horizontal load F_y and a vertical load F_z , which correspond to the normal and stream-wise projections of \vec{F} , respectively (Equations 3.15 and 3.16) [24].

$$F_y = - \oint_{\Gamma_{SF}} (\rho v \frac{\partial \vec{v}_t}{\partial \vec{n}} n_z - p n_y) d\Gamma_{SF} \quad (3.15)$$

$$F_z = \oint_{\Gamma_{SF}} (\rho v \frac{\partial \vec{v}_t}{\partial \vec{n}} n_y + p n_z) d\Gamma_{SF} \quad (3.16)$$

Where \vec{n} is the vector normal to the solid-fluid boundary, ν is the kinematic viscosity, ρ is the fluid density, and \vec{v}_t is the fluid velocity tangential to the surface. The resultant load is applied on the neutral point O of the section. Hence the aerodynamic moment with respect to x axis needs to be computed as well (Equation 3.17).

$$M_x = \oint_{\Gamma_{SF}} (\rho v \frac{\partial \vec{v}_t}{\partial \vec{n}} n_z - p n_y) (z - z_O) d\Gamma_{SF} + \oint_{\Gamma_{SF}} (\rho v \frac{\partial \vec{v}_t}{\partial \vec{n}} n_y + p n_z) (y - y_O) d\Gamma_{SF} \quad (3.17)$$

Where θ is the angle between the normal vector and the stream-wise direction of the flow.

The forces and moments need to be translated to the neutral point of the geometry, which is the reference point of the beam model. Hence the load vector for a certain beam model node is specified as follows (Equation 3.18).

$$\mathbf{f}^i = \begin{Bmatrix} f_{x'_i} \\ f_{y'_i} \\ f_{z'_i} \\ m_{x'_i} \\ m_{y'_i} \\ m_{z'_i} \end{Bmatrix} = \begin{Bmatrix} 0 \\ F_y \\ F_z \\ M_x \\ 0 \\ 0 \end{Bmatrix} \quad (3.18)$$

The boundary integral presented above can be performed using numerical integration. The problem domain is discretized into finite elements, where the values of p and v at each node are known. Hence it is possible to interpolate the pressure and velocity values of the corner nodes of those elements that are intersected by the solid-fluid boundary in order to approximate the pressure and velocity distribution along the boundary (Equation 3.19).

$$\{p(\chi, \eta), v(\chi, \eta)\} = \begin{bmatrix} N_1^{int} & 0 & 0 & 0 \\ 0 & N_2^{int} & 0 & 0 \\ 0 & 0 & N_3^{int} & 0 \\ 0 & 0 & 0 & N_4^{int} \end{bmatrix} \begin{bmatrix} \{p_1, v_1\} \\ \{p_2, v_2\} \\ \{p_3, v_3\} \\ \{p_4, v_4\} \end{bmatrix} \quad (3.19)$$

Where χ and η are the local coordinates of the element and N_n^{int} are the shape functions for linear integration associated to each node of the finite element (Equations 3.20 to 3.23).

$$N_1^{int} = \frac{(1 - \chi)(1 - \eta)}{4} \quad (3.20)$$

$$N_2^{int} = \frac{(1 + \chi)(1 - \eta)}{4} \quad (3.21)$$

$$N_3^{int} = \frac{(1 + \chi)(1 + \eta)}{4} \quad (3.22)$$

$$N_4^{int} = \frac{(1 - \chi)(1 + \eta)}{4} \quad (3.23)$$

The integration is performed using 3 Gauss points per element, located at the intersection line. Integration of equations 3.15, 3.16, and 3.17 are approximated by equations 3.24, 3.25, and 3.26.

$$F_y = \sum_i^{n_{be}} \sum_{gp}^3 -\rho v \left[\left(\frac{\partial u_y}{\partial z} \right)_i^{gp} - \left(\frac{\partial u_z}{\partial y} \right)_i^{gp} \right] W_i^{gp} \sin(\arctan(\frac{\vec{n}_i^z}{\vec{n}_i^y})) + p_i^{gp} W_i^{gp} \cos(\arctan(\frac{\vec{n}_i^z}{\vec{n}_i^y})) \quad (3.24)$$

$$F_z = \sum_i^{n_{be}} \sum_{gp}^3 \rho v \left[\left(\frac{\partial u_y}{\partial z} \right)_i^{gp} - \left(\frac{\partial u_z}{\partial y} \right)_i^{gp} \right] W_i^{gp} \cos(\arctan(\frac{\vec{n}_i^z}{\vec{n}_i^y})) + p_i^{gp} W_i^{gp} \sin(\arctan(\frac{\vec{n}_i^z}{\vec{n}_i^y})) \quad (3.25)$$

$$M_x = \sum_i^{n_{be}} \sum_{gp}^3 \{ \rho v \left[\left(\frac{\partial u_y}{\partial z} \right)_i^{gp} - \left(\frac{\partial u_z}{\partial y} \right)_i^{gp} \right] \cos(\arctan(\frac{\vec{n}_i^z}{\vec{n}_i^y})) + p_i^{gp} \sin(\arctan(\frac{\vec{n}_i^z}{\vec{n}_i^y})) \} W_i^{gp} (y_{gp} - y_0) \\ + \sum_i^{n_{be}} \sum_{gp}^3 \rho v \left[\left(\frac{\partial u_y}{\partial z} \right)_i^{gp} - \left(\frac{\partial u_z}{\partial y} \right)_i^{gp} \right] \sin(\arctan(\frac{\vec{n}_i^z}{\vec{n}_i^y})) + p_i^{gp} \cos(\arctan(\frac{\vec{n}_i^z}{\vec{n}_i^y})) \} W_i^{gp} (z_{gp} - z_0) \quad (3.26)$$

Where n_{be} is the number boundary elements, \vec{n}_i^y and \vec{n}_i^z are the projections of the normal vector of the interface line of each element on the stream-wise and normal directions, respectively, and p_i^{gp} is the interpolated value of the pressure at each Gauss point of the interface line of each element. Velocity derivatives $\frac{\partial u_y}{\partial z}$ and $\frac{\partial u_z}{\partial y}$ are numerically performed by the derivation of the shape functions (Equations 3.27 and 3.28).

$$\frac{\partial u_y}{\partial z} = \frac{\partial N_n^{int}}{\partial z} u_y^n \quad (3.27)$$

$$\frac{\partial u_z}{\partial y} = \frac{\partial N_n^{int}}{\partial y} u_z^n \quad (3.28)$$

3.4. Verification of the FSI model

In order to verify the FSI model, a simulation of a solid circular disc immersed on a laminar flow has been carried (Figure 3.1). The inlet velocity distribution is parabolic. This is a classical benchmark problem used in CFD [24, 33, 37]. The properties of both solid and fluid phases can be seen in Table 3.1.

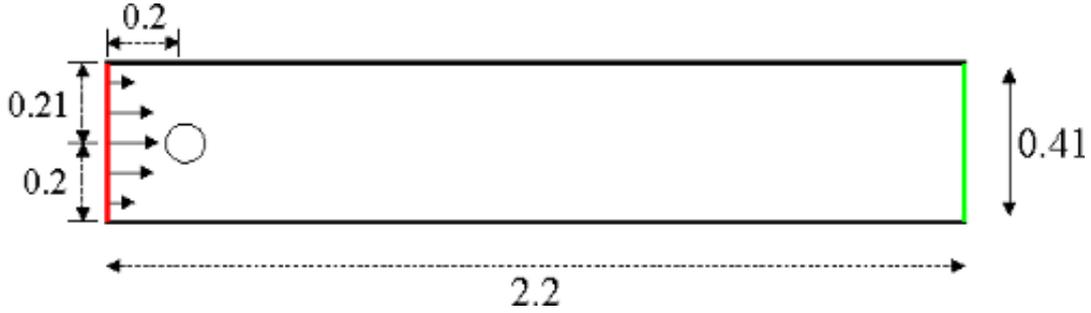


Figure 3.1: Verification of the FSI analysis. Geometry of the problem [33].

Circle radius [m]	0.05
Reference length [m]	0.1
Fluid average speed [m/s]	0.2
Fluid maximum speed [m/s]	0.3
Fluid Density [kg/m ³]	1
Reynolds number	20

Table 3.1: Main properties of Ω_s and Ω_f used for verifying the FSI analysis tool

The dimensionless Lift (C_l) and Drag (C_d) coefficients can be calculated using Equations 3.29 and 3.30.

$$C_l = \frac{2F_z}{U_{mean}^2 L_{ref}} \quad (3.29)$$

$$C_d = \frac{2F_y}{U_{mean}^2 L_{ref}} \quad (3.30)$$

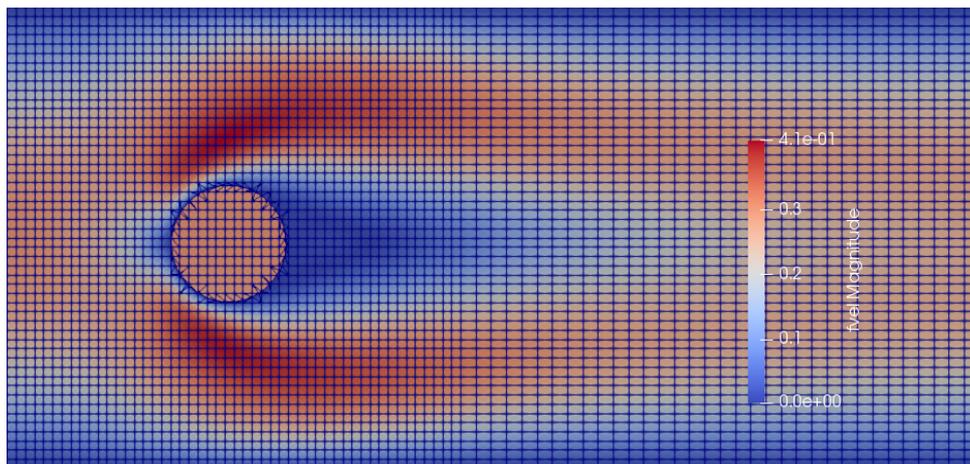
According to the benchmark [24, 33, 37], the expected results are: $C_l = 0.0106$, $C_d = 5.5795$, and $\Delta p = 0.1175$

A summary of the results from the convergence study can be seen on Table 3.2.

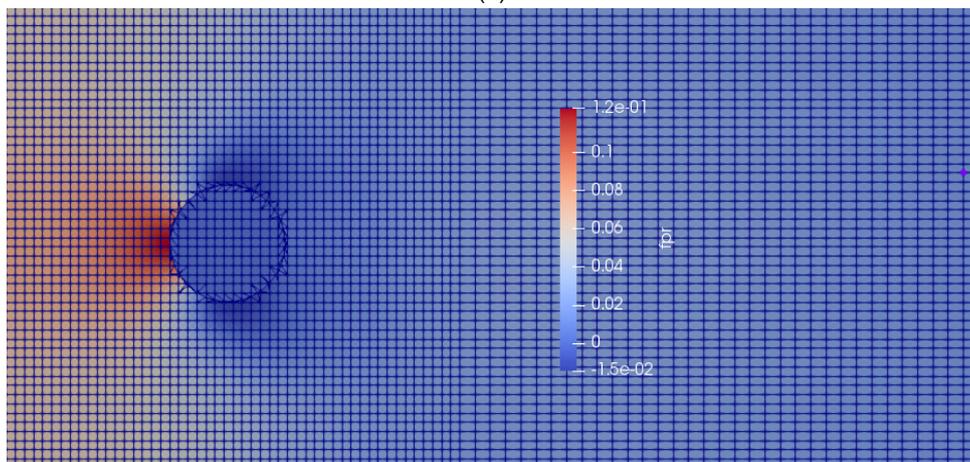
Element length	C_d	C_l	Δp
4.1e-2	4.0878	-0.0578	0.0970
2.05e-2	5.6093	0.00894	0.1194
1.37e-2	5.5372	-0.01821	0.1157
1.03e-2	5.4961	0.01423	0.1158
8.2e-3	5.5763	-0.0128	0.1156
Benchmark solution	5.5795	0.0106	0.1175

Table 3.2: Results for the convergence study carried during the verification process of the FSI analysis.

The pressure and fluid distribution can be seen on Figure 3.2.

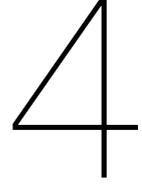


(a)



(b)

Figure 3.2: Verification of the FSI analysis. Fluid velocity (a) and fluid pressure (b) distributions around the disk.



Global System of Equations. Monolithic solver

In the previous sections it has been described every part of the turbine blade model. As it can be seen on Figure 1.4, there is a non linear behavior of the problem set. In order to deal with this non-linearity it has been proposed to use a Full-Newton solver for solving the global system of equations with a single numerical iterative solving procedure.

4.1. Strong form of Governing Equations

Considering the sub-models and assumptions presented in sections 2 and 3, the following System of Equations governs the problem (Equations 4.1 to 4.7), given an initial Level Set field $\Phi_0(x_i)$.

$$\text{Beam model:} \quad K_{ll}^{glob} \mathbf{u}^l - \mathbf{f}^l(F_y, F_z, M_x) = 0 \quad \text{in beam} \quad (4.1)$$

$$\text{Fluid momentum equation:} \quad \frac{\partial v_i^f}{\partial t} + v_j^f \frac{\partial v_i^f}{\partial x_j} = \frac{1}{\rho^f} \frac{\partial \sigma_{ij}^f}{\partial x_j} + b_i^f \quad \text{in } \Omega_F \quad (4.2)$$

$$\text{Fluid continuity equation:} \quad \frac{\partial v_i^f}{\partial x_i} = 0 \quad \text{in } \Omega_F \quad (4.3)$$

$$\text{Projection of } \Phi: \quad \Phi(x_i) - \Phi_0(x_i - \Delta x_i) = 0 \quad \text{in } \Omega \quad (4.4)$$

$$\text{Horizontal force:} \quad dF_y + (\rho v \frac{\partial \vec{v}_t}{\partial \vec{n}} n_z - p^f n_y) = 0 \quad \text{in } \Gamma_{SF} \quad (4.5)$$

$$\text{Vertical force:} \quad dF_z - (\rho v \frac{\partial \vec{v}_t}{\partial \vec{n}} n_y + p^f n_z) = 0 \quad \text{in } \Gamma_{SF} \quad (4.6)$$

$$\begin{aligned} \text{Moment:} \quad & dM_x - (\rho v \frac{\partial \vec{v}_t}{\partial \vec{n}} n_y + p^f n_z)(y - y_0) \\ & - (\rho v \frac{\partial \vec{v}_t}{\partial \vec{n}} n_z - p^f n_y)(z - z_0) = 0 \quad \text{in } \Gamma_{SF} \quad (4.7) \end{aligned}$$

4.2. Weak form of Governing Equations

The system of equations presented in the previous section can be integrated into 4 set of residuals ($R_{beam}, R_{fluid}, R_\Phi$, and R_{aero}). 4 unknown fields are considered: $\mathbf{u}_{beam} = \mathbf{u}^l$, $\mathbf{w}_{fluid} = \{\mathbf{v}^f = \{v_y, v_z\}, p^f\}$, Φ , and $\mathbf{f}_{aero} = \{F_y, F_z, M_x\}$. Hence, the solution vector is $\vec{b} = \{\mathbf{u}^l, \mathbf{v}^f, p^f, \Phi, F_y, F_z, M_x\}$.

The following spaces of trial solutions \mathcal{S} (Equation 4.8) and test functions \mathcal{V} are considered (Equation 4.9).

$$\text{Trial solutions:} \quad \{\mathbf{u}^l, \mathbf{v}^f, p^f, \Phi\} \quad \text{in } \mathcal{S} \quad (4.8)$$

$$\text{Test functions:} \quad \{\delta \mathbf{u}^l, \delta \mathbf{v}^f, \delta p^f, \delta \Phi\} \quad \text{in } \mathcal{V} \quad (4.9)$$

Hence the system of equations to be solved can be stated as follows (Equations 4.10 to 4.13).

$$R_{beam}\{\mathbf{u}^l, F_y, F_z, M_x\} = K_{lm}^{glob} \mathbf{u}^l - f_l(F_y, F_z, M_x) = 0 \quad (4.10)$$

$$R_{fluid}\{\mathbf{v}^f, p^f, \Phi\} = \int_{\Omega} d\Omega \delta v_i^f \left(\rho \frac{\partial v_i}{\partial t} + \rho v_j \frac{\partial v_i}{\partial x_j} - \frac{\partial \sigma_{ij}(v^f, p^f)}{\partial x_j} - \rho g_i \right) + \int_{\Omega} d\Omega \delta p^f \frac{\partial v_j}{\partial x_j} = 0 \quad (4.11)$$

$$R_{\Phi}\{\mathbf{u}^l, \Phi, \Phi_0\} = \int_{\Omega} d\Omega N^T (\Phi(x_i) - \Phi^{ref}(x_i - Z_i(x_i) \mathbf{u}_{sec}^l)) = 0 \quad (4.12)$$

$$\begin{aligned} R_{aero}\{\mathbf{v}^f, p^f, \Phi, F_y, F_z, M_x\} &= R_{F_y}\{\mathbf{v}^f, p^f, \Phi, F_y\} + R_{F_z}\{\mathbf{v}^f, p^f, \Phi, F_z\} + R_{M_x}\{\mathbf{v}^f, p^f, \Phi, M_x\} = \\ &= \oint_{\Gamma_{SF}} [dF_y + d\Gamma_{SF}(\rho v \frac{\partial \vec{v}_t}{\partial \vec{n}} n_z - p^f n_y)] + \oint_{\Gamma_{SF}} [dF_z - d\Gamma_{SF}(\rho v \frac{\partial \vec{v}_t}{\partial \vec{n}} n_y + p^f n_z)] \\ &+ \oint_{\Gamma_{SF}} [dM_x - d\Gamma_{SF}(\rho v \frac{\partial \vec{v}_t}{\partial \vec{n}} n_y + p^f n_z)(y - y_0) - d\Gamma_{SF}(\rho v \frac{\partial \vec{v}_t}{\partial \vec{n}} n_z - p^f n_y)(z - z_0)] \end{aligned} \quad (4.13)$$

Which can be linearized as Equation 4.14.

$$\mathbf{J} \Delta \vec{b} = \mathbf{J} \Delta \begin{Bmatrix} \mathbf{u}^l \\ \{\mathbf{v}^f, p^f\} \\ \Phi \\ \{F_y, F_z, M_x\} \end{Bmatrix} = - \begin{Bmatrix} R_{beam} \\ R_{fluid} \\ R_{\Phi} \\ R_{aero} \end{Bmatrix} \quad (4.14)$$

Where \mathbf{J} is the Jacobian matrix (Equation 4.15).

$$\mathbf{J} = \begin{bmatrix} \frac{\partial R_{beam}}{\partial \mathbf{u}^l} & \frac{\partial R_{beam}}{\partial \{\mathbf{v}^f, p^f\}} & \frac{\partial R_{beam}}{\partial \Phi} & \frac{\partial R_{beam}}{\partial \{F_y, F_z, M_x\}} \\ \frac{\partial R_{fluid}}{\partial \mathbf{u}^l} & \frac{\partial R_{fluid}}{\partial \{\mathbf{v}^f, p^f\}} & \frac{\partial R_{fluid}}{\partial \Phi} & \frac{\partial R_{fluid}}{\partial \{F_y, F_z, M_x\}} \\ \frac{\partial R_{\Phi}}{\partial \mathbf{u}^l} & \frac{\partial R_{\Phi}}{\partial \{\mathbf{v}^f, p^f\}} & \frac{\partial R_{\Phi}}{\partial \Phi} & \frac{\partial R_{\Phi}}{\partial \{F_y, F_z, M_x\}} \\ \frac{\partial R_{aero}}{\partial \mathbf{u}^l} & \frac{\partial R_{aero}}{\partial \{\mathbf{v}^f, p^f\}} & \frac{\partial R_{aero}}{\partial \Phi} & \frac{\partial R_{aero}}{\partial \{F_y, F_z, M_x\}} \end{bmatrix} \quad (4.15)$$

The derivation of the terms of the Jacobian can be seen on Appendix A.

4.3. Verification of the Full Newton Solver

In order to verify the Full Newton solver model, a simulation of a 3D hollowed beam immersed on a laminar flow has been carried (Figure 4.1). The fluid conditions for every section are those from the benchmark problem (Section 3.4). The Young modulus has been set to $69GPa$. At every section, an element length of $1.37e-2$ has been used, as a trade-off between accuracy and computing time. The aspect ratio of the beam is 20.

The beam has been split into 2 element, with different geometries at each of the 3 sections (Figure 4.2): A square cross-section located at clamped end, a star cross-section located at the middle of the beam, and a circle cross-section located at the beam tip.

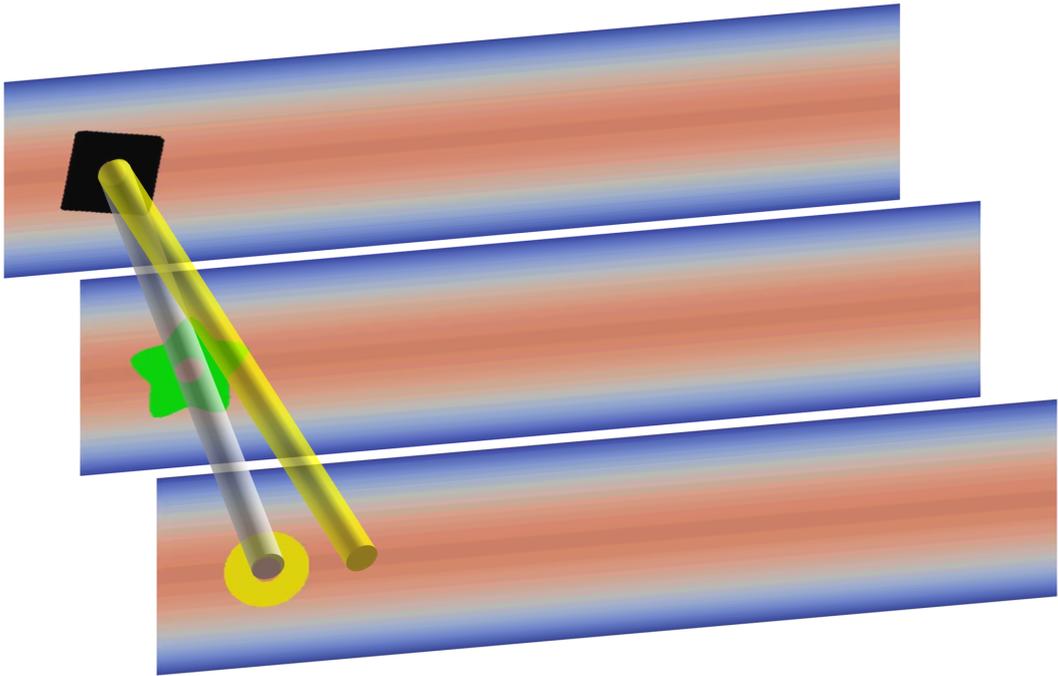


Figure 4.1: Verification of the Monolithic Newton solver. Initial geometry and deformed configuration of the beam.

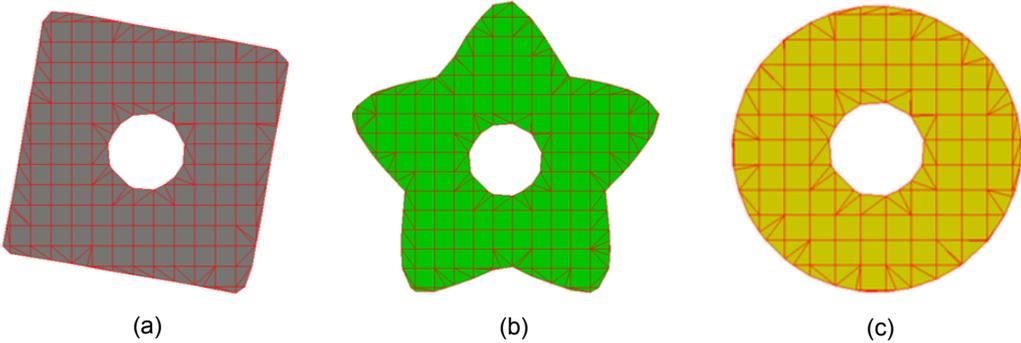


Figure 4.2: Different cross-section geometries used along the beam length. (a) Square cross-section (Located at clamped end), (b) Star cross-section (Located at middle of the beam), and (c) Circle cross-section (Located at tip).

A convergence plot of the monolithic Newton solver can be seen on Figure 4.3.

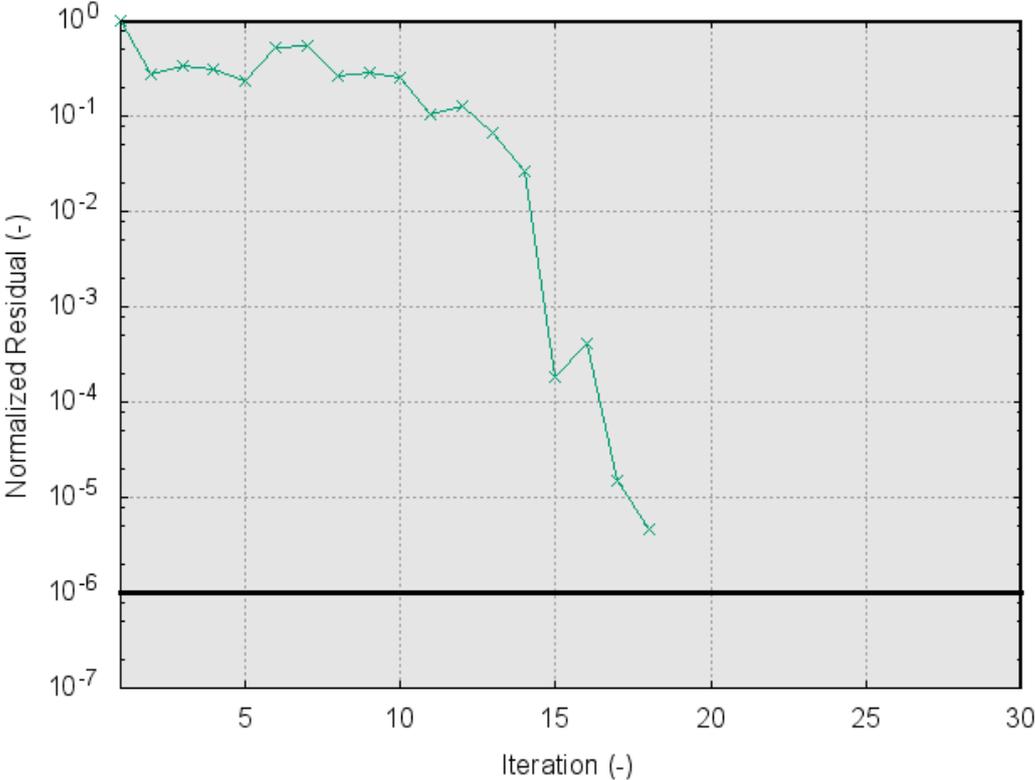


Figure 4.3: Verification of the Monolithic Newton solver. Convergence of the normalized residual with respect to the number of iterations.

A Newton relaxation parameter of 0.8 is applied to the beam displacements, and a Newton convergence criteria of $1e-5$ is used in all FSI analyses here. The visual results of this verification FSI example can be seen on Appendix B.

5

Optimization

The FSI problem described in Chapter 4 can be optimized with respect to a certain objective function q and subject to certain constraints g .

In this case, the optimization consists on finding the optimal cross-section geometry, i.e. the optimal initial Level Set function that defines the geometry, which is parametrized as a function of optimization variables s_i that minimizes q , as it can be seen on the following schema [16].

$$\text{Find optimum } s_i \begin{cases} \min(q) \\ g \leq 0 \end{cases}$$

5.1. Sensitivity Analysis

The first step is to perform the sensitivity analysis $\frac{dq}{ds_i}$, this is to compute relationship between the objective function, q , and the optimization variables, s_i . The sensitivity analysis is performed using the Adjoint method [26]. A transient non linear analysis is assumed (Equation 5.1). The dynamic residual can be obtained after FE discretization and integrating in time using a 1st order backward Euler method (Equation 5.2).

$$\frac{\partial b_i}{\partial t} + r_i(b_i) \simeq \frac{1}{\Delta t}(b_i^n - b_i^{n-1}) + r_i(b_i^n) = 0 \quad (5.1)$$

$$\mathbf{R}_{dyn}^n = \int d\Omega \delta b_i \left[\frac{1}{\Delta t}(b_i^n - b_i^{n-1}) + r_i(b_i^n) \right] = \frac{1}{\Delta t} M^n (\mathbf{b}^n - \mathbf{b}^{n-1}) + \mathbf{R}^n \quad (5.2)$$

Where b_i is the vector of state variables ($\vec{b} = \{\mathbf{u}^l, \mathbf{v}^f, p^f, \Phi, F_y, F_z, M_x\}$), r_i static residual in strong form, \mathbf{b}^n is the solution vector at t^n , M^n is the mass matrix at t^n , and \mathbf{R}_{dyn}^n and \mathbf{R}^n are the dynamic and static residuals at t^n , on their weak form, respectively.

From equilibrium (Equation 5.3), the derivative of the solution vector with respect to the optimization variables can be expressed as Equation 5.4.

$$\frac{d\mathbf{R}_{dyn}}{ds_i} = \frac{\partial \mathbf{R}_{dyn}}{\partial s_i} + \frac{\partial \mathbf{R}_{dyn}}{\partial \mathbf{b}} \frac{d\mathbf{b}}{ds_i} \quad (5.3)$$

$$\frac{d\mathbf{b}}{ds_i} = - \left(\frac{\partial \mathbf{R}_{dyn}}{\partial \mathbf{b}} \right)^{-1} \frac{\partial \mathbf{R}_{dyn}}{\partial s_i} \quad (5.4)$$

Hence, the sensitivity can be expressed as follows (Equation 5.5).

$$\frac{dq}{ds_i} = \frac{\partial q}{\partial s_i} + \frac{\partial q}{\partial \mathbf{b}} \frac{d\mathbf{b}}{ds_i} = \frac{\partial q}{\partial s_i} - \frac{\partial q}{\partial \mathbf{b}} \left(\frac{\partial \mathbf{R}_{dyn}}{\partial \mathbf{b}} \right)^{-1} \frac{\partial \mathbf{R}_{dyn}}{\partial s_i} = \frac{\partial q}{\partial s_i} - \mathbf{A}^T \frac{\partial \mathbf{R}_{dyn}}{\partial s_i} \quad (5.5)$$

Where $\mathbf{A} = \left(\frac{\partial \mathbf{R}_{dyn}}{\partial \mathbf{b}} \right)^T \frac{\partial q}{\partial \mathbf{b}}$ is the Adjoint vector. It can be computed using a backward time integrator (Equation 5.6). In this case, only 2 time steps are considered, as only stationary conditions are considered on this analysis.

$$\begin{bmatrix} (M^0/\Delta t + J^0)^T & (-M^1/\Delta t)^T & 0 & \ddots & 0 & 0 & 0 \\ 0 & (M^1/\Delta t + J^1)^T & (-M^2/\Delta t)^T & \ddots & 0 & 0 & 0 \\ 0 & 0 & (M^2/\Delta t + J^2)^T & \ddots & 0 & 0 & 0 \\ 0 & 0 & 0 & \ddots & (-M^{n-1}/\Delta t)^T & 0 & 0 \\ 0 & 0 & 0 & \ddots & (M^{n-1}/\Delta t + J^{n-1})^T & (-M^n/\Delta t)^T & 0 \\ 0 & 0 & 0 & \ddots & 0 & (M^n/\Delta t + J^n)^T & 0 \end{bmatrix} \begin{Bmatrix} A^0 \\ A^1 \\ A^2 \\ \vdots \\ A^{n-1} \\ A^n \end{Bmatrix} = \begin{Bmatrix} \frac{\partial q^0}{\partial \mathbf{b}} \\ \frac{\partial q^1}{\partial \mathbf{b}} \\ \frac{\partial q^2}{\partial \mathbf{b}} \\ \vdots \\ \frac{\partial q^{n-1}}{\partial \mathbf{b}} \\ \frac{\partial q^n}{\partial \mathbf{b}} \end{Bmatrix}$$

$$\frac{\partial \hat{\mathbf{R}}_{dyn}}{\partial \hat{\mathbf{b}}} \hat{\mathbf{A}} = \frac{\partial \hat{q}}{\partial \hat{\mathbf{b}}} \quad (5.6)$$

Hence, the adjoint vector terms are computed as follows:

$$\begin{aligned} A^n &= ((M^n/\Delta t + J^n)^T)^{-1} \frac{\partial q^n}{\partial \mathbf{b}} \\ A^{n-1} &= ((M^{n-1}/\Delta t + J^{n-1})^T)^{-1} \left(\frac{\partial q^{n-1}}{\partial \mathbf{b}} - (-M^n/\Delta t)^T A^n \right) \\ &\vdots \\ &\vdots \\ A^1 &= ((M^1/\Delta t + J^1)^T)^{-1} \left(\frac{\partial q^1}{\partial \mathbf{b}} - (-M^2/\Delta t)^T A^2 \right) \end{aligned}$$

The Jacobians and the mass matrices are the same ones as the ones analytically computed on the forward analysis, at equilibrium, whereas the $\frac{\partial \mathbf{R}_{dyn}}{\partial s_i}$ term is computed using finite differences by locally perturbing the optimization parameters.

Furthermore, the partial derivatives of the residual, objective, and constraints with respect to the geometrical optimization variables are also computed by a centered finite difference scheme.

Due to the use of a monolithic solver scheme for the forward analysis (Section 4.2), the Jacobian of the forward analysis (Derivative of the residual with respect to the state variable vector) is already computed and can be reused in the gradient computation required for the optimization process [23].

5.2. Verification of the sensitivities

In order to verify the Sensitivity Analysis as described in Section 5.1, a FSI analysis has been carried over a hollowed cylindrical beam of aspect ratio equal to 20.

The cross-section is constant along the beam length, consisting of a circle of constant radius, and an inner concentric circular hole. The radius of the inner hole is parametrized as a the optimization variable s_i of the problem (Figure 5.1).

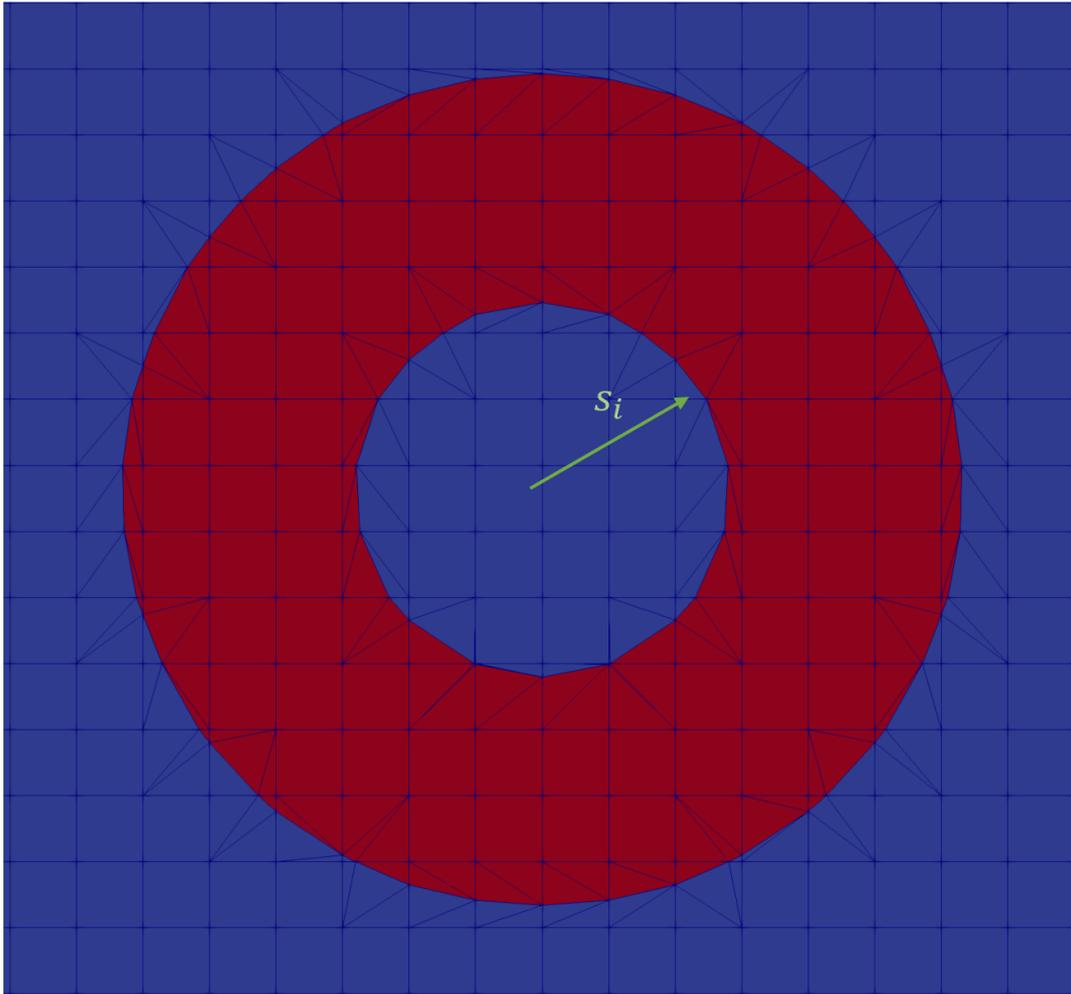


Figure 5.1: Hollowed circular shape used at the cross-sections in order to verify the Sensitivity Analysis.

The fluid properties and boundary conditions are the same as the ones used in 3.4. The position of the centroid of the cross-section as well as the most refined are of the mesh have been moved to the mid part of the flow channel in order to avoid fluid perturbation from the inlet (Figure 5.2).

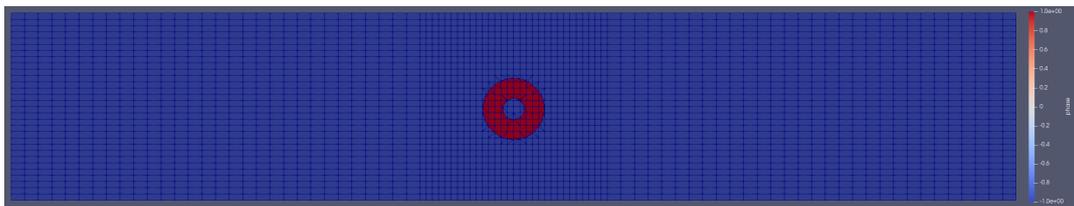


Figure 5.2: Position of the cross-section at the middle of the flow channel.

The beam is decomposed into a single beam element (Only sections at both ends are considered) in order to reduce computation consumption.

In this case, the optimization consists on finding the optimal value for the inner radius that minimizes q , which in this case is chosen as the displacement at the tip, subject to a minimum perimeter constraint, as it can be seen on the following schema [16].

$$\text{Find optimum } s_i \left\{ \begin{array}{l} q = \min \left(\sqrt{u_{x_i'}^2 + u_{y_i'}^2 + u_{z_i'}^2} \right) \text{ at section } = n \\ g = g_0 - \text{perimeter} \leq 0 \end{array} \right.$$

Where g_0 is a constant reference value for the minimum perimeter constraint.

In this case, the following terms have been used for the computation of the adjoint and the sensitivities (Equations 5.11 and 5.12).

$$\frac{\partial \hat{q}}{\partial \mathbf{b}} = \frac{\left[0 \quad \vdots \quad 2u_{x_i'} \quad 2u_{y_i'} \quad 2u_{z_i'} \quad 0 \quad 0 \quad 0 \quad \vdots \quad 0 \right]^T}{2q} \quad (5.7)$$

$$\frac{\partial q}{\partial s_i} = 0 \quad (5.8)$$

The variation $\frac{\partial R^{dyn}}{\partial s_i}$ is computed by finite differences taking into account the explicit effect of s_i on the original Level Set field. This means it has an influence on the calculation of K_{beam} and the position of the centroid $\{x_0, y_0\}$, and the reference Level Set field used on the Level Set projection module Φ^{ref} .

In order to do the verification, the sensitivities obtained by the Adjoint method have been compared with those computed by a global finite differences scheme, i.e. the initial Level Set field is perturbed and then the forward analysis is carried on to obtain z and g for the perturbed field [39].

5.2.1. Influence of Finite Difference perturbation size

To have accurate results when using a finite difference scheme, the result needs to be invariable with respect to the perturbation size, hence the values belonging to the plateau region of a $\frac{\partial q}{\partial s_i}$ with respect to h graph are those who need to be used. Where h is the perturbation value.

In this case (Figure 5.3), $1e-6$ is the value to be used in all the finite difference schemes employed (Corresponding to the global finite difference scheme, as well as the $\frac{\partial \mathbf{R}_{dyn}}{\partial s_i}$ terms used for the Adjoint method. It is also employed for computing the $\frac{\partial \mathbf{R}}{\partial \Phi}$ crossed terms of the Jacobian).

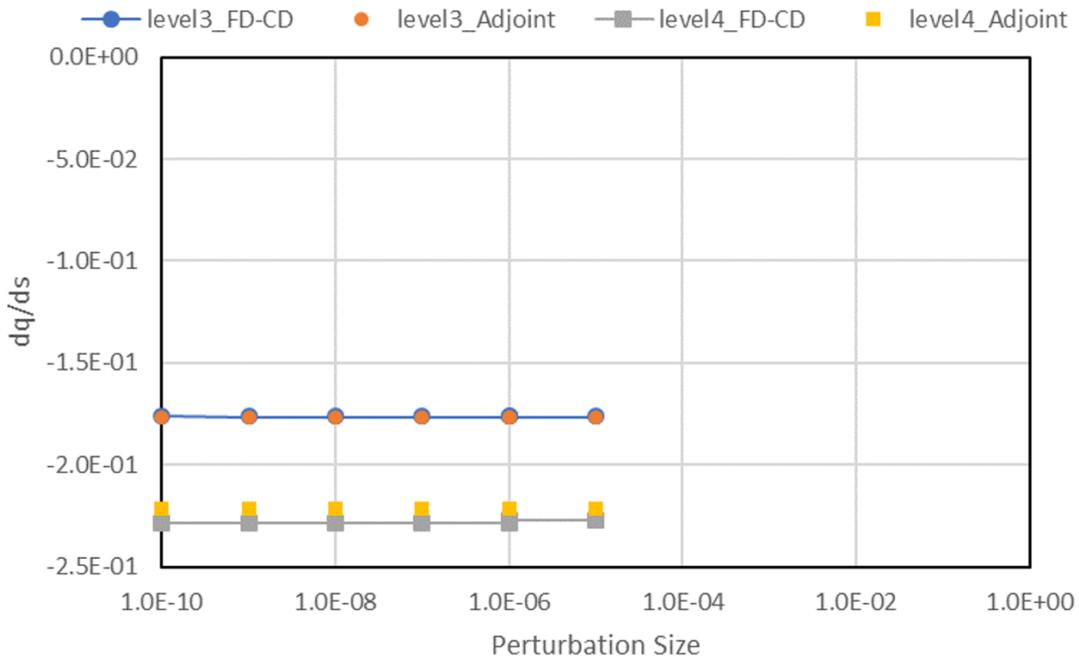


Figure 5.3: Influence of the perturbation on the computation of the sensitivities for the Adjoint method and global finite difference schemes. Different mesh refinement values (Circles: $1.37e-2$. Squares: $1.03e-2$) are used.

Note: A center finite difference scheme is used in all cases [45].

5.2.2. Influence of Mesh Refinement

To have reliable results, the obtained values need to be invariable with respect to the mesh refinement too. In this case, the obtained objective function q tend to converge when using element lengths smaller or equal than $1.03e-2$ (Figure 5.4).

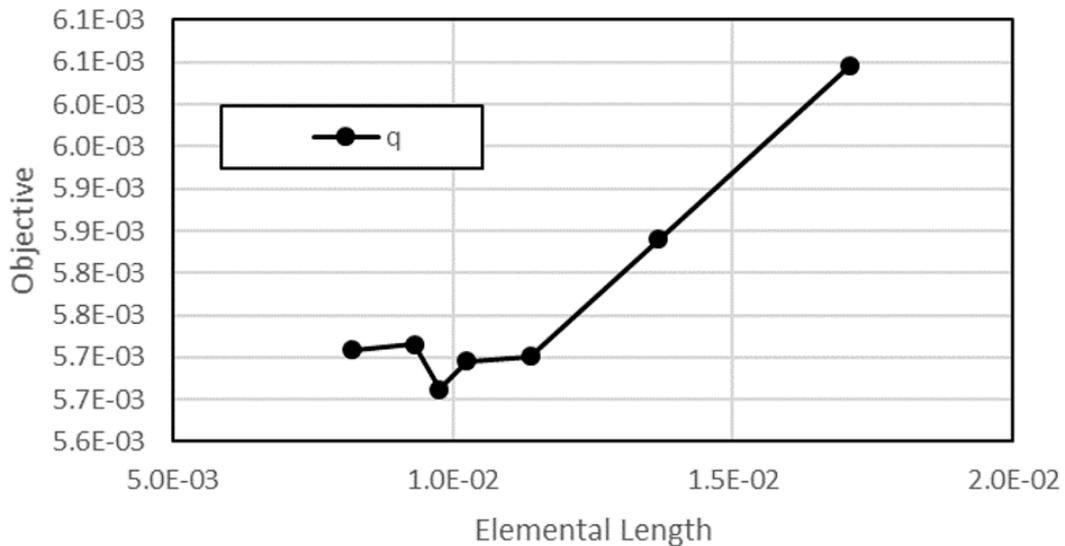
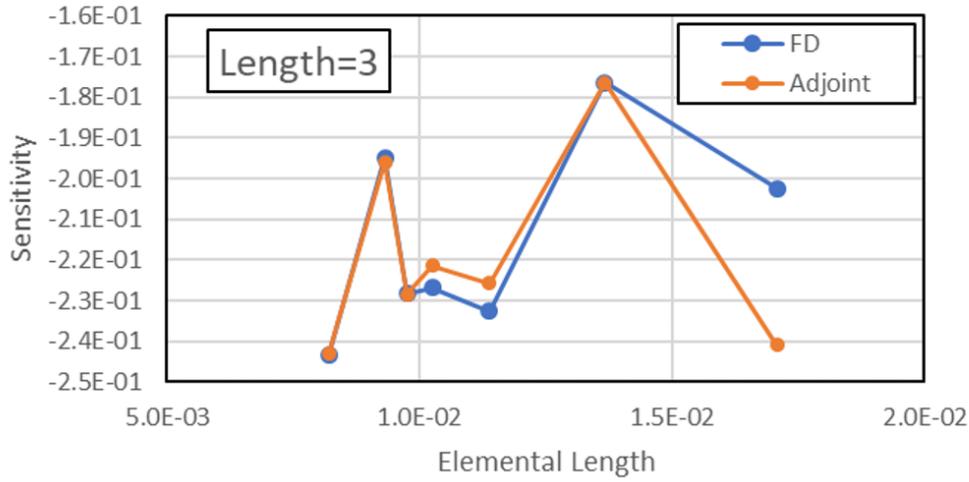
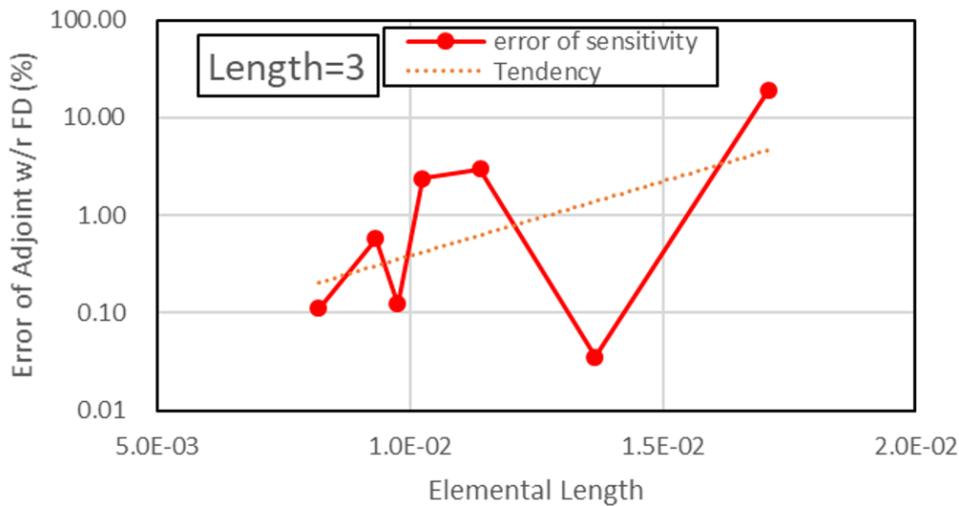


Figure 5.4: Influence of the mesh refinement on the objective function.

It can also be seen that when using more refined meshes the relative difference between the sensitivities computed using finite differences and the adjoint method become smaller (Figure 5.5).



(a)



(b)

Figure 5.5: Influence of the mesh refinement on the Sensitivities. 5.5a) Obtained sensitivities from both methods. 5.5b) Relative difference between the obtained sensitivities from both methods

In order to limit the required computation time for the optimization, an element length of $9.76e-3$ has been chosen for carrying on the optimization.

5.3. Results

An example of optimization has been carried on using the same setup (Hollowed cylindrical beam) as the one used for verifying the sensitivity analysis (Section 5.2).

The optimization has been performed using the Globally Convergent Method of Moving Asymptotes (GCMMA). More information about GCMMA and how to implement its algorithm can be found on [40, 42].

5.3.1. Minimum Tip Displacement

Following the same setup as in Section 5.2, the purpose is to minimize the norm of the beam displacement at the tip, measured at the centroid of the cross-section, subject to minimum perimeter constraint $g_0 = 1.14$ (Equation 5.9).

$$\text{Find optimum } s_i \begin{cases} q = \min \left(\sqrt{u_{x'_i}^2 + u_{y'_i}^2 + u_{z'_i}^2} \right) & \text{at section } = n \\ g = g_0 - \text{perimeter} \leq 0 \end{cases} \quad (5.9)$$

In Figure 5.6 it can be seen the evolution of q and g with respect to the optimization variable s_i .

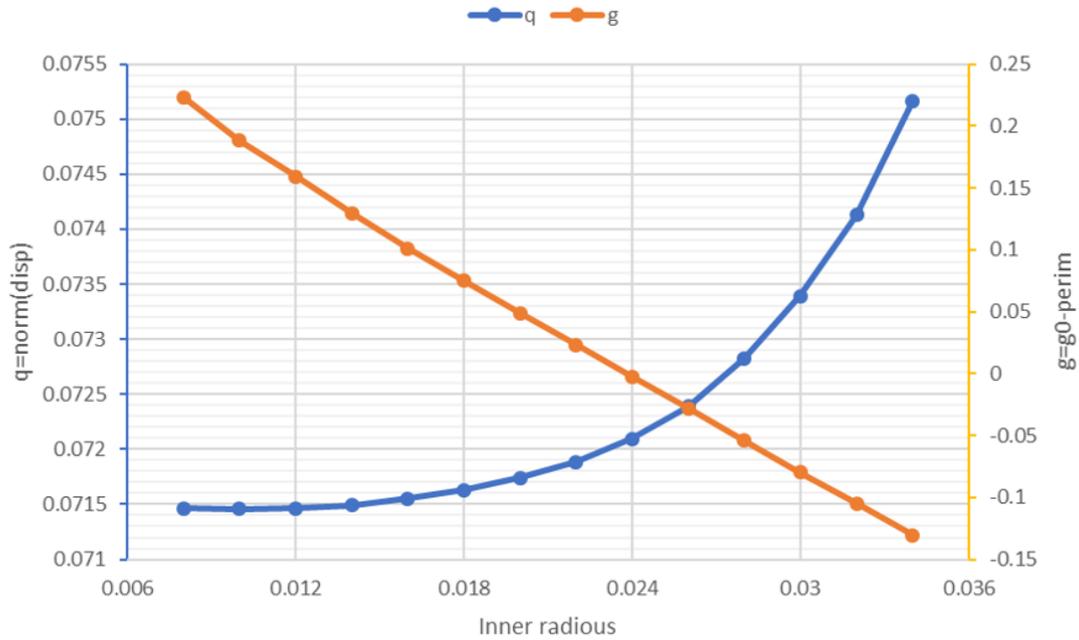


Figure 5.6: Influence of the Optimization variable on the objective and constraint.

As it can be seen on Figure 5.6, the optimum value of s_i that satisfies the constraint is somewhere below 0.024.

The obtained results from GCMMA are shown below. In Figure 5.7 it can be seen the convergence rate of the residual of the Karush-Kuhn-Tucker (KKT) conditions [42] and the obtained optimum value for the optimization variable.

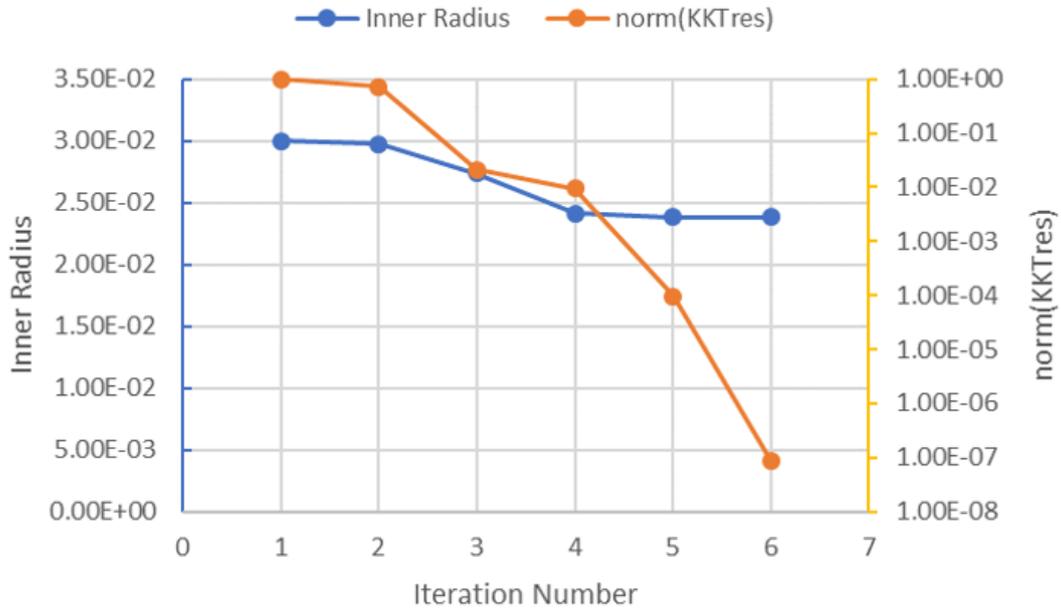


Figure 5.7: Convergence rate of the norm of the KKT residual for the minimum tip displacement case. Obtained optimum variable for the inner radius. Initial geometry conditions were satisfying the constraint.

In Figure 5.8 it can be seen the variation of the objective and constraint functions through the optimization iterations.

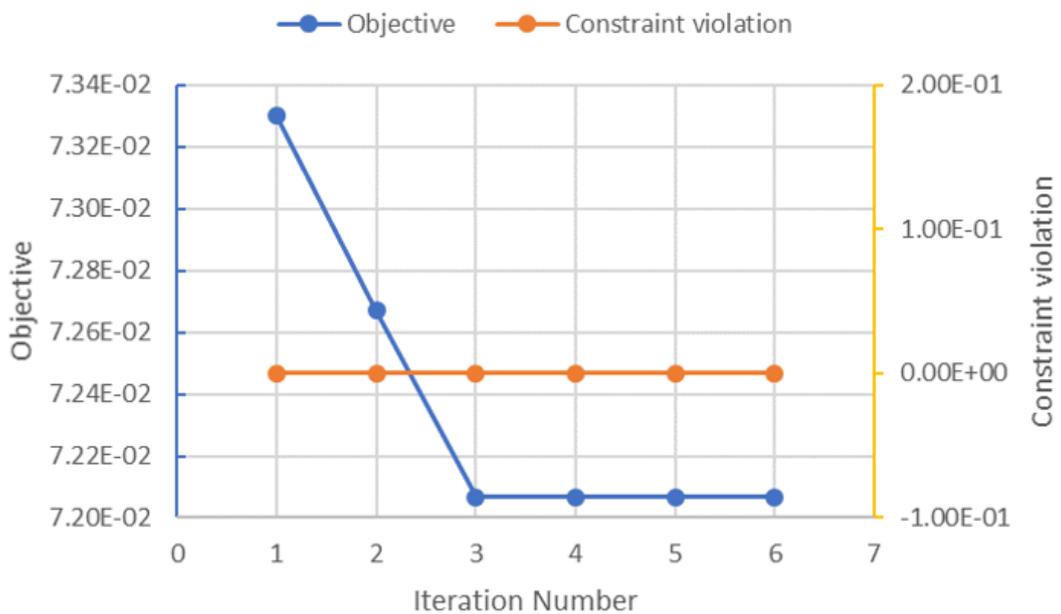


Figure 5.8: Variation of the objective and constraint through the iterations for the minimum tip displacement case. Initial geometry conditions were satisfying the constraint.

In the case above the initial geometric conditions were satisfying the constraint. On the other hand, in Figures 5.9 and 5.10 the same graphs are shown, this time starting the optimization process using a geometry that does not satisfy the constraint.

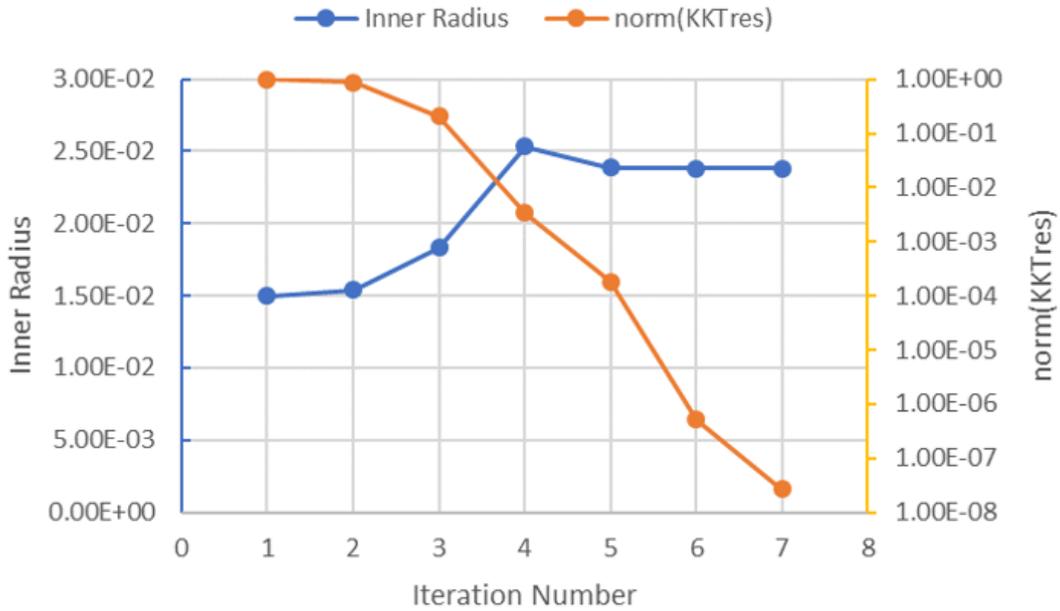


Figure 5.9: Convergence rate of the norm of the KKT residual for the minimum tip displacement case. Obtained optimum variable for the inner radius. Initial geometry conditions were not satisfying the constraint.

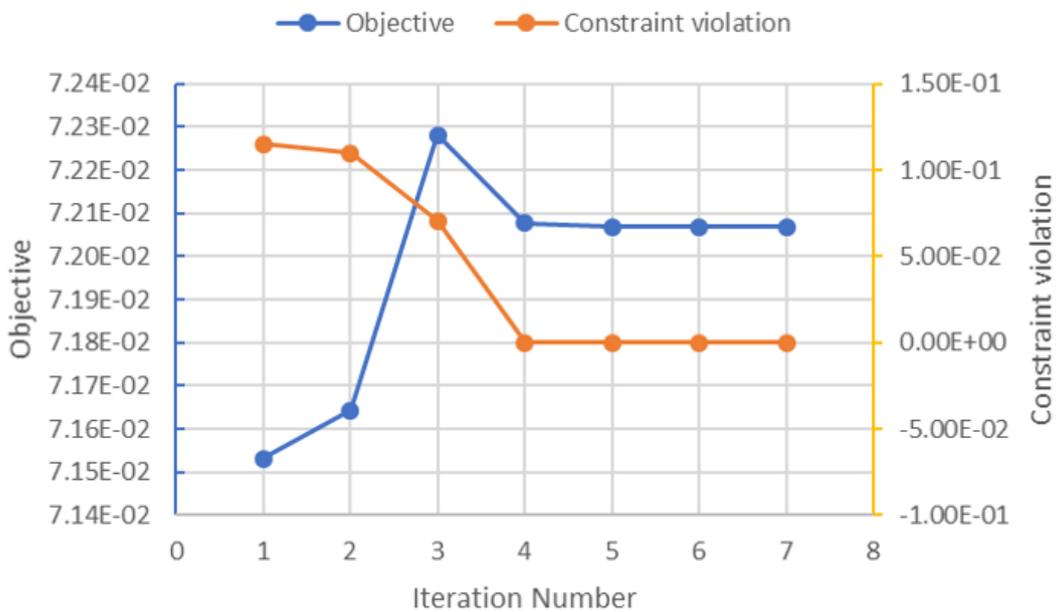


Figure 5.10: Variation of the objective and constraint through the iterations for the minimum tip displacement case. Initial geometry conditions were not satisfying the constraint.

It can be appreciated in Figures 5.7 to 5.10 that in both cases the optimization problem converges to same point, which is the one that was estimated on Figure 5.6. Furthermore, the convergence rate is slightly faster if the initial geometric conditions already satisfy the constraint.

The graphical evolution of the cross-section geometry can be seen on Appendix C.

5.3.2. Minimum Strain energy

Following the same setup as in Section 5.2, the purpose is to minimize the strain energy of the beam, subject to minimum perimeter constraint $g_0 = 1.14$ (Equation 5.10).

$$\text{Find optimum } s_i \begin{cases} q = \min \left[\frac{1}{2} (\mathbf{u}^T \mathbf{K}^{glob} \mathbf{u}^l) \right] \\ g = g_0 - \text{perimeter} \leq 0 \end{cases} \quad (5.10)$$

In this case, the following terms have been used for the computation of the adjoint and the sensitivities (Equations 5.11 to 5.12).

$$\frac{\partial \hat{q}}{\partial \hat{\mathbf{b}}} = [[\mathbf{K}^{glob} \mathbf{u}^l] \quad 0 \quad \dots \quad 0]^T \quad (5.11)$$

$$\frac{\partial q}{\partial s_i} = \frac{1}{2} \left(\mathbf{u}^l{}^T \frac{\partial \mathbf{K}^{glob}}{\partial s_i} \Big|_{FD} \mathbf{u}^l \right) \quad (5.12)$$

Obtained results from GCMMA are shown below. In Figure 5.11 it can be seen the convergence rate of the residual of the Karush-Kuhn-Tucker (KKT) conditions [42] and the obtained optimum value for the optimization variable.

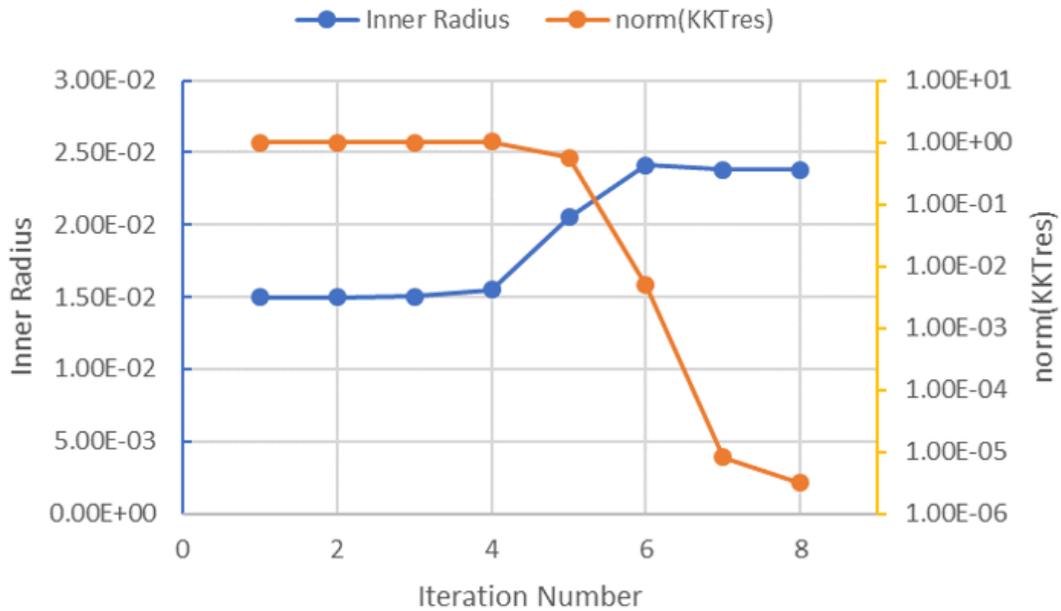


Figure 5.11: Convergence rate of the norm of the KKT residual for the minimum strain energy case. Obtained optimum variable for the inner radius. Initial geometry conditions were not satisfying the constraint.

In Figure 5.12 it can be seen the variation of the objective and constraint functions through the optimization iterations.

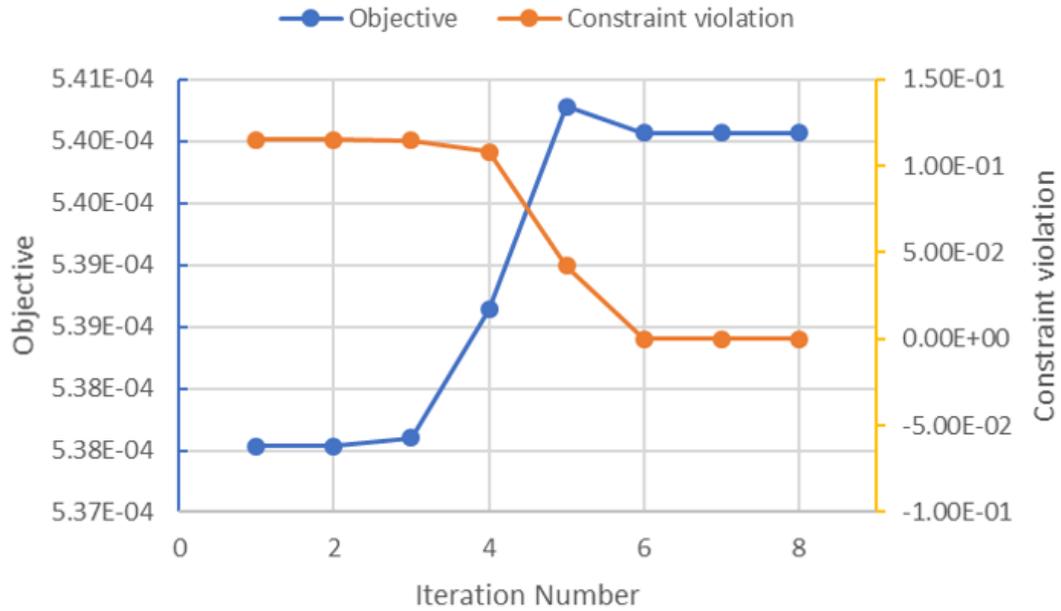


Figure 5.12: Variation of the objective and constraint through the iterations for the minimum strain energy case. Initial geometry conditions were not satisfying the constraint.

In this case the initial geometric conditions were not satisfying the constraint. As it can be appreciated, convergence rate is slower on the first steps, as compared with Section 5.3.1 case. Nevertheless, the algorithm leads to the same optimum point than in the previous case, as the same constraint function is used.

5.3.3. Other cases

The sensitivity analysis has been performed over more complex cross-section shapes, including multiple optimization variables, such as the one that can be seen on Figure 5.13.

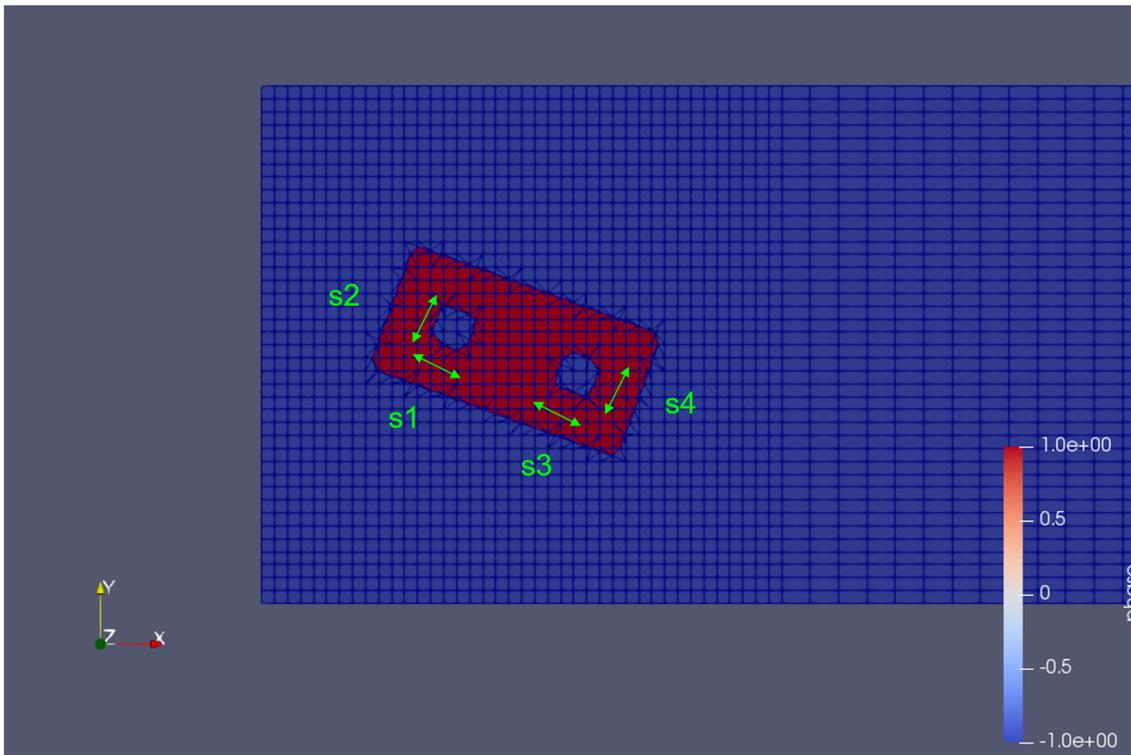


Figure 5.13: Initial Level Set parametrization of a hollowed rectangular cross-section, using 4 optimization variables s_i .

However, in this case the required mesh refinement to reproduce the geometric details with enough accuracy, which lead to Matlab limitations in terms of memory usage, joint with the large computation needs for carrying on the optimization with multiple variables, had made not applicable the optimization of these more complex shapes under the scope and time frame of the Thesis project.

This is also applicable to the topology optimization of the cross-sections, as in that case the optimization variables correspond to the the full set of Level Set nodal values, increasing a lot the computation time. In addition, the computation of the $\frac{\partial R^{dyn}}{\partial s_i}$ term using finite differences, becomes very slow for such a large amount of optimization variables (Equal to the total number of nodes).

Conclusions and Recommendations

6.1. Discussion

The 3D model developed for this research is conceptually very complex, owning strong coupling and non-linearities, and its accurate implementation has needed a careful analytical and numerical analysis, joint with extended coding sessions that were followed by long debugging periods, in order to obtain a robust monolithic solver that is able to be handled on the optimization stage. In the next section some the points that were tackled during the project are described, joint with a critical reflection on how they impacted on the overall work.

6.1.1. Modeling

On the one hand, the beam model (Section 2.1) needed to adapt the 2D information from the cross-sections onto the 3D space through interpolation. Different locations for the cross-sections along the beam length were considered, such as the beam element ends or the beam element Gauss points. For simplicity, the beam element ends were chosen as points where the cross-sections are located along the beam length.

In terms of interpolation, both linear interpolation and averaging of the cross-sections were considered. Linear interpolation was finally chosen, as the required shape functions and numerical integration parameters were already calculated for the computation of the beam element stiffness matrix (Section 2.1.5).

Regarding the 2D fluid mesh, which was also used for computing the cross-section inertia properties, main discussion was about how to handle the two fields (Fluid and Solid) and the interface into one single Eulerian mesh. The mesh is supposed to stay invariable through the iterations. A Level Set function was used to define the fluid and solid fields, and the interface location. An interface capturing method is necessary to define the intersection points within the mesh due to the Eulerian description of the problem, and the boundary elements were enriched using XFEM to accurately define the interface without expensive re-meshing steps.

In addition, a trade-off between computation efficiency and accuracy was done in order to choose the mesh refinement level for each mesh (Fluid and beam). In the case of the beam mesh, larger increments of mesh refinement increased a lot the required computation time, as it is related with the total number of cross-sections, thus related with the total fluid mesh size. As it can be seen on Section 2.4, large increments on the number of beam elements do not largely increment the accuracy of the benchmark problem. Hence, to reduce the time of the calculations, only one or two beam elements were considered in most of the cases for the sensitivity analysis and optimization tests. It is assumed that this approach is more accurate the smoother the cross-section properties change along the beam length.

On the other hand, in the case of the fluid mesh, a local refinement of the fluid mesh area that is closer to the cross-section within the channel flow helps to increase the accuracy without increasing a lot the memory consumption and the computation time. Furthermore, the local meshes associated to the geometry and fluid field of each cross-section are treated as one big single mesh within the Matlab code, sharing the local system of coordinates (i.e.

the local meshes are superposed on on top of each other), in order to make use of the already developed settings for single-mesh problems within TransFEM, which needed to be adapted in order to avoid superposed/duplicated node errors and plotting issues due to the difficulty to separate the meshes back in Paraview viewer for visualization purposes.

Larger computation times where a big deal, especially at the debugging phases, joint with Matlab out of memory errors (Max. memory error in Matlab). To alleviate the problem the computation of the Jacobian terms in Matlab has been performed using sparse notation, in order to increase the efficiency of the code, allowing the use of larger values for the mesh refinement before running out of memory.

6.1.2. Level Set field

The Level Set field definition has an influence over all the stages of the project (Modeling, solver and optimization)

The Level Set function is usually defined as a signed distance function (e.g. a cone) as it is easier to capture the interface within the mesh and to perform Heaviside smoothing steps, due to the clear definition of the LSF zero isocontour.

While this LS definition is able to accurately reproduce circular shapes, it leads to inaccuracies when dealing with more complex shapes, especially those including sharp corners and edges (Figure 5.13). It also makes more complex the creation of parametrized internal "holes", which are necessary for the definition of the initial geometry guess when carrying shape and topology optimization simulations.

In order to solve this issue, a Matlab script is existing within TransFEM, which is able to generate a consistent Level Set field from a black & white bitmap file. This allows the use of more complex shapes, such as NACA profiles, and swiss-cheese hollowed structures (Necessary for defining the hollowed geometric initial conditions required by topology optimization) within the TransFEM framework (Figure 6.1).

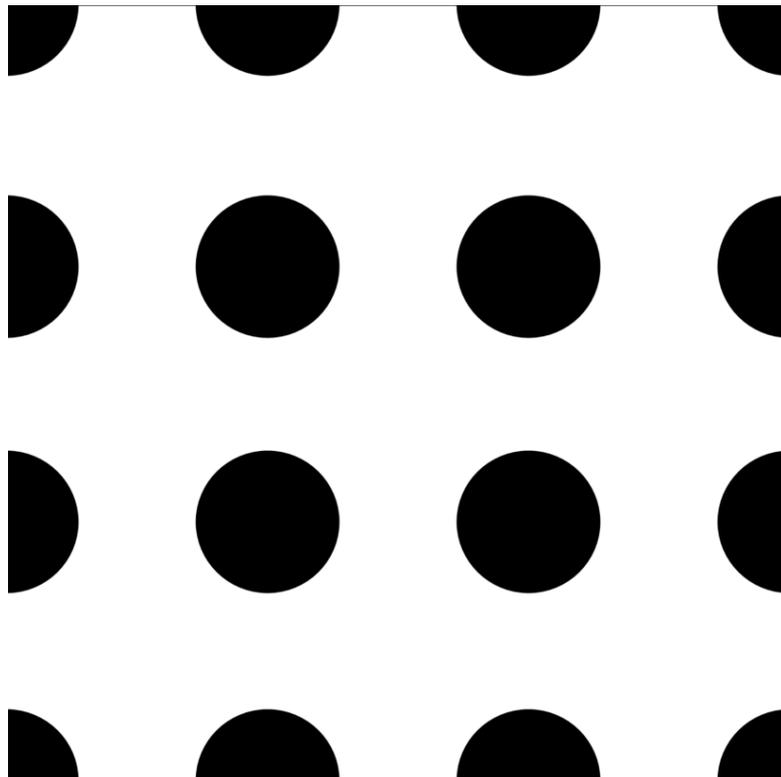


Figure 6.1: Swiss-cheese hollowed pattern black & white bitmap file. Used to define hollowed geometries necessary as an initial guess for the topology optimization procedure.

It is also noticeable the fact that the Level Set function values at the nodes $\Phi(\mathbf{X})$ are treated

as degrees of freedom (DOFs) for the Fluid-Structure Interaction problem. It is important to distinguish between the Level Set function as a DOF, and the original Level Set field that defines the initial geometric properties of the cross-sections. This original Level Set field $\Phi^{ref}(\mathbf{X})$ is necessary to compute the stiffness matrix of the cross-sections, as well as the position of centroid of each cross-section. It is also used as reference Level Set field for the Level Set projection module.

It is of extreme importance to take this difference into consideration, especially when computing the $\frac{\partial}{\partial \Phi}$ terms of the Jacobian that are influenced only by $\Phi(\mathbf{X})$ as a DOF, or the $\frac{\partial}{\partial s_i}$ terms of the sensitivity analysis, which are only influenced by $\Phi^{ref}(\mathbf{X})$. This becomes more complex in the case of topology optimization, as all the $\Phi^{ref}(\mathbf{X})$ values are considered as optimization variables.

In the case of the Level Set projection module (Section 2.3) is it also important to consider a threshold value to filter the nodes corresponding to the far fluid field (i.e. Those nodes owning a very low value of Φ and $\frac{\partial}{\partial \Phi} = 0$) in order to speed up the projection process, by focusing only on those nodes that lie within the solid field or in its surroundings. In this case those values of Φ that are equal or smaller than -4.95 lie below the threshold are filtered, thus being ignored by the Level Set projection scheme.

Other Level Set projection schemes were considered, such as a mirrored translation and rotation of the background mesh, or a dual mesh system (Fluid + Solid one). However, the actual development was the most simple and efficient one in terms of computing the Jacobian terms while keeping the mesh invariable and unique at all times.

6.1.3. Numerical Instabilities

During the implementation of the code and the check phases, different numerical issues have been faced that affected negatively the stability of the results, the convergence rate or the accuracy of the obtained solution.

In the case of the Newton solver, there were some important convergence issues, which used to appear at the initial iterations of the loop in those cases where the allowed displacement of the cross-section was too large (e.g. Long slender beam, soft material, faster fluid flow, etc). At those occasions the initial guess for the Level Set field (i.e. Equivalent to the straight beam) was too far away from the final converged solution, leading to meaningless intermediate states that especially affect the Level Set field and the fluid distribution (e.g. Appearance of strange deformed geometries), which may eventually generate the blow up of the computation or very slow convergence rates.

In order to deal with it, a predictor for the Level Set field was used for the first stages. It consisted of a separate Level Set projection step that was run at the beginning of each Newton iteration, taking into account the beam displacements calculated on the previous iteration as an input, in order to update the Level Set field to a consistent shape before updating the fluid field and the rest of the solution vector. The use of the predictor, and its efficiency, is problem dependent, and it is only used until the convergence path is reached, i.e. It is used until the convergence criteria (Norm of the residual) is smaller than a certain value, generally when $norm(res) < 0.01$.

Another way to improve the convergence of the Newton solver is the addition of a Newton relaxation parameter. This is a scalar α that multiplies some of the terms of the update vector (Equation 6.1) to improve the convergence of the solution.

$$\mathbf{X}_{n+1} = \mathbf{X}_n - \alpha \frac{f'(\mathbf{X}_n)}{f(\mathbf{X}_n)} \quad (6.1)$$

In this case the relaxation is only applied to the beam displacements and $0 < \alpha < 1$ is the Newton relaxation parameter, which is problem dependent. After some testing $\alpha = 0.5$ was chosen as the most efficient one in most cases of this project.

This technique is very useful, especially when facing an oscillatory behavior of the convergence criteria on the first iterations, as it considerably reduces the amplitude of the solution oscillations (Figure 6.2).

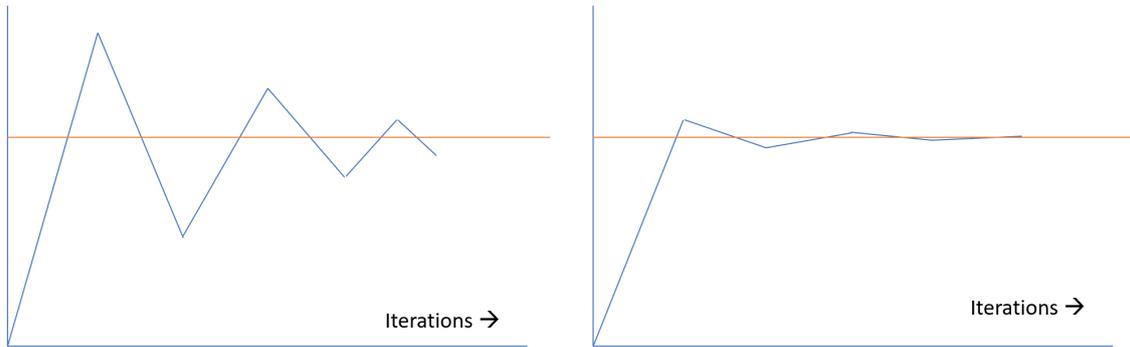


Figure 6.2: Simplified idealization of the performance increment when using Newton relaxation stabilization to help to improve convergence under strong oscillatory behaviors. Without (Left) and with (Right) Newton relaxation. In red the expected converged solution.

There are other situations where due to the limited size of the design domain, an intermediate solution for the Level Set field may eventually lay out of the bounds of the design domain, specially when large cross-section displacements are allowed (i.e. When using a low stiffness beam). When this phenomena occurs the simulation crashes, as the Matlab code is not able to treat correctly the FSI model. To mitigate the problem, a penalty factor is added to the update vector of the beam displacements, to scale the displacement within the design domain. However, when this situation continuously repeats throughout the iterations, the simulation shall be aborted, as it may be necessary to increase the size of the design domain.

Another important topic to check during the forward analysis and the computation of the sensitivities is the condition number of the Jacobian. Ill-conditioned systems of equations usually negatively affect the convergence when dealing with non-linearities, thus reducing the performance of linear iterative solvers such the Newton-Raphson developed for this project [48]. Poorly conditioned matrices may lead to inaccuracies on the solution vector or on the obtained sensitivities. The influence of this numerical error is larger the larger the condition number of the Jacobian is.

One of the main causes of ill-conditioning of the Jacobian is the difference in terms of order of magnitude between the different components of the matrix, due to the multiphysics of the problem. The accurate non-dimensionalization of the equations and variables may solve this issue. In this case it was tested the application of a Jacobi preconditioner [27] to the Jacobian matrix, which makes the diagonal terms equal to 1. However, despite reducing the condition number of the global Jacobian, the effect on the obtained results and the convergence rate was almost negligible in most cases.

An ill-conditioned matrix system also appears in XFEM when the ratio of fluid to solid phase within an intersected element is very close to zero or to the unity (i.e. The Interface is very small and it is very close to one of the nodes) [48]. This configuration makes appearance either due to an incorrect initial definition of the geometry, through the Level Set mapping during the LS projection stage, or as a result of the geometry update during the optimization process. Face-oriented ghost-penalty methods are chosen to mitigate the problem in TransFEM [13, 14].

Another source of ill-conditioning of the Jacobian is the apparition of large peaks of fluid pressure on hollowed structures when performing the FSI analysis, due to the difficult mathematical treatment of the fluid phase that lies within the holes of the cross-section. The apparition of this extremely high pressure peaks causes the crash of the computation. The solution taken was to temporary remove the fluid pressure and velocity degrees of freedom from the active DOFs list of those nodes that lie inside the outer interface of the cross-section, at beginning of each Newton iteration.

As it is only considered the steady state of the fluid, a large enough time step needs to be chosen in order to avoid transient states on the fluid field through the iterations. It has been observed that the choice of small time steps considerably increases the global condition number of the Jacobian. At the same time it has a negative effect on convergence, as the

fluid field of the partial solutions is perturbed by the fluid field of previous iterations, and this perturbation takes a long time to vanish through the iterations. Hence, the correct choice of the time step is of high importance in this case.

Finally, it has been observed a big influence of the $\frac{\partial}{\partial \Phi}$ crossed terms of the Jacobian. As stated in Appendix A, these terms are very difficult to account analytically, and they are numerically derived by using Finite Differences. Furthermore, within the Newton loop, when the current solution vector for the fluid field variables is very different from the converged one (Such as at the initial steps), the simulation tends to blow up, with abnormal values of v_f and p_f at the boundary elements, when these $\frac{\partial}{\partial \Phi}$ crossed terms are active on the Jacobian. The way chosen to deal with this issue is to temporarily disable these $\frac{\partial}{\partial \Phi}$ crossed terms on the Jacobian for the first iterations, until the convergence criteria is below a certain value. At this point the $\frac{\partial}{\partial \Phi}$ crossed terms are reactivated and the simulation continues as usual, with a good convergence rate. After some testing, it was chosen $norm(res) = 0.01$ as a good threshold for the activation of these terms. These $\frac{\partial}{\partial \Phi}$ crossed terms are also included in the Jacobian when performing the sensitivity analysis.

6.2. Conclusions

The initial goal of the MSc. Thesis project was to obtain optimal wind turbine blade cross-sections, using topology optimization techniques in addition to modern tools (e.g. LSM and X-FEM), presenting a crisp definition of its geometry and contour, in order to potentially make applicable this technique in the industrial design process.

As a result of this research, it has been implemented a 3D Fluid-Structure Interaction (FSI) model applicable to slender structures, such as wind turbine blades, by combining a 3D beam model with a 2D fluid model, under low Reynolds number flows. The main advantage of this setup is a considerable reduction of the computational cost, as compared with a full 3D FSI setup. In addition, the use of XFEM in combination of the Level Set method allows obtaining a better resolution of the geometry and crisp interfaces between solid and fluid, without the necessity of re-meshing, as in the case of classical FEM.

The proposed 3D beam is partially based on the formulation of the classical beam element for slender beams (Euler-Bernoulli), including Saint-Venant torsional effects for isotropic materials, and with the addition of the terms related with the coupling between axial and torsion, and bending and torsion contributions, which may arise when using non-linear materials. The stiffness information of the beam is interpolated from its cross-section geometries and materials, which can vary along the beam length.

The cross-section geometries are defined on a XFEM mesh. The fluid and solid domains are specified using a Level Set Function. A 2D fluid simulation based on Incompressible Navier Stokes flow at low Reynolds number is carried around each cross-section, in order to obtain the aerodynamic loading over its contour. This aerodynamic loading serves as an input for the beam model, to compute deformation of the beam. This deformation is mapped onto the cross-sections, obtaining the updated displacements and rotations of the geometry. With the updated geometry the fluid field is altered and it needs to be updated as well, forming a non-linear iterative process that loops until a converged structure is obtained.

The 3D FSI model is solved on a monolithic Newton-Raphson solver that treats all the equations involved at once. The Jacobian terms derived for the monolithic solving scheme that has been developed for the forward analysis allow a straightforward computation of the sensitivities using adjoint method. This sensitivity analysis makes possible the optimization of the geometry of the cross-sections based on certain criteria and constraints.

The different models used have been independently tested for their validation, such as the beam model (Section 2.4) and the fluid model (Section 3.4). This done, the full 3D FSI model was tested using an iterative staggered scheme, where the fluid and beam parts were separately computed before updating the geometry of the cross-section at each iteration.

The staggered solving scheme was evolved into a monolithic Newton solver that treats the whole system of equations at once (Section 4), where all the dependencies between the system equations and variables needed to be accounted. The main reason for choosing a monolithic solver is the possibility to use the same derivatives involved in the construction of

the Jacobian matrix of the forward analysis for the sensitivity analysis, as they are necessary for the implementation of the Adjoint Method too. As seen in the verification example of Section 4.3, a consistent Jacobian matrix has been obtained, as the Newton solver is able to converge to an accurate enough solution. This fact makes suitable the use of this Jacobian for performing further steps, such as optimization.

A sensitivity analysis was carried based on the adjoint method (Section 5.1), obtaining similar results when comparing the sensitivities obtained through the adjoint method with those obtained from the finite difference analysis. Despite this difference is smaller than 1% in most of the cases, this number is still quite large as compared with other sensitivity analysis from the literature, which could be related with the strong non-linearity of the problem, XFEM mesh related issues or another unknown sources of error. A lot of work during the project has been invested on improving the sensitivity analysis and debugging, without being able to obtain better results.

Finally, several optimization examples have been performed (Section 5.3), where simple shapes with a small number of optimization variables have been optimized under certain criteria (e.g. Minimum tip displacement for a hollowed circular beam). However, due to the computing limitations of Matlab and time constraints, only simple shape optimization cases could be carried during the scope of this MSc. Thesis. Performing topology optimization of the beam looks theoretically feasible, however the amount of time and resources needed (i.e. The amount of optimization variables equals the size of the the Level Set function $\Phi(\mathbf{X})$ - equal to the total number of nodes) has not made it possible during this MSc. Thesis work.

No similar works have been found on the literature, hence no valid benchmarks for the full 3D FSI optimization model have been used to compare the obtained results on this research, and they have been only validated by convergence studies and modular benchmark examples.

Nevertheless it has been a very enriching experience where I have learned a lot on how the research world works and I have improved a lot my knowledge on FSI and structural optimization. The project has shown it is worth the effort to develop a monolithic solver for dealing with complex multiphysics problems in order to allow further optimization stages to improve current system's performance to meet the desired objectives under certain constraints. The use of an XFEM model in combination with the Level Set method in a monolithic approach for the optimization of FSI problems [51], allows obtaining clearer interfaces and a more crisp representation of the geometry [45], as compared with other optimization techniques, such as the density-based approach [7], which is a very important fact to account for during the industrial design process.

Hence, this method has a great potential for the future, however still some improvements on its implementation need to be performed, in order to boost the performance and reliability of the model to more complex academic and industrial standards. As stated on the next section (Section 6.3), the following research topics and recommendations are suggested to improve the current method in future works.

6.3. Recommendations for Future Research

- The actual 2D XFEM model could be improved to accommodate more complex displacements and deformations. For example, the solid phase could be allowed to deform based on the aerodynamic loading, besides translating and rotating as it is allowed so far. The current 2D cross-section mesh could be also improved to allow wrapping displacements of the cross-sections in order to improve the reliability of the model.
- The actual fluid model, based on Incompressible Navier Stokes formulation, could be improved into a more complex one. The most important improvement would be the accommodation of higher Reynolds numbers within the model, as this would allow the simulation of more realistic fluid flows. Another recommendation is the addition of out of plane loading and/or fluid flows, as they can be treated by the 3D beam model, in order to increase the accuracy of the fluid analysis. The treatment of more complex phenomena, such as turbulence or boundary layer, would be interesting as well.
- The computation efficiency of the code could be improved (e.g. Using more efficient

software tools rather than Matlab), in order to reduce the computation cost of the optimization, allowing the optimization of multiple variable problems, and reducing the time required for the simulations. A better way to manage the memory for storing the Jacobian matrix would also be beneficial, as it would allow the use of finer meshes.

- An more efficient way to perform the sensitivity analysis with respect to the actual implementation could be developed within the code, in order to perform a complete topology optimization of the cross-section, as the actual number of optimization variables - equal to the number of nodes, which correspond to the size of the Level Set function array - becomes too large and consumes lots of resources and time.
- The addition of a more complex material model for the cross-section, by including more complex non linear material models, plasticity, multiple layers and/or composites, to increase the freedom of the design.
- The use of a better analytical model for the derivation of the Jacobian, as some of the terms could not be derived using analytical methods on this research, treating them numerically by finite differences. This applies especially to those terms that are derived with respect to the Level Set field ($\frac{\partial}{\partial \phi}$ crossed terms).
- The inclusion of better benchmarks, both from the literature and from software modeling, in order to check the reliability and accuracy of the obtained results, as they could not be found on this research.
- The tune up of the Newton-Raphson solver, for example with the addition of more sophisticated predictors and preconditioners, in order to improve the stability of the results and the convergence rate.
- Finding a better way to define the initial geometry of the cross-sections as a Level Set field is also important to be developed, as it would allow the use of more complex shapes on the model (e.g. NACA profiles). The method of mapping a black & white bitmap file on the mesh, as used in other parts of TransFEM could be valid for this task.
- The treatment of the temporal evolution of the problem would be interesting too, by dealing with the transient states of the fluid and beam models.

$$\frac{\partial R_{beam}}{\partial \Phi} = \begin{bmatrix} \vdots & 0 & \vdots \\ \vdots & 0 & \vdots \end{bmatrix} \quad (A.5)$$

$$\frac{\partial R_{beam}}{\partial F_y} = \begin{bmatrix} \vdots \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix} \quad (A.6)$$

$$\frac{\partial R_{beam}}{\partial F_z} = \begin{bmatrix} \vdots \\ 0 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix} \quad (A.7)$$

$$\frac{\partial R_{beam}}{\partial M_x} = \begin{bmatrix} \vdots \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \\ 0 \\ \vdots \end{bmatrix} \quad (A.8)$$

$$\frac{\partial R_{beam}}{\partial \eta} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \quad (A.9)$$

Note: A clamping boundary condition has been added to the displacements of the first node.

A.1.2. Fluid model terms

These are the terms associated with the fluid model. From Equation 4.11, using PSPG and SUPG stabilization methods (δv_i^f is the stabilized test function):

$$\begin{aligned}
R_{fluid}\{\mathbf{v}^f, p^f, \Phi\} &= \\
&= \int_{\Omega} d\Omega \delta v_i \left(\rho \frac{\partial v_i}{\partial t} + \rho v_j \frac{\partial v_i}{\partial x_j} - \frac{\partial \sigma_{ij}(\mathbf{v}^f, p^f)}{\partial x_j} - \rho g_i \right) + \int_{\Omega} d\Omega \delta p^f \frac{\partial v_j}{\partial x_j} = \\
&= \int_{\Omega} d\Omega \left(\delta v_i + \tau_{SUPG} v_k \frac{\partial \delta v_i}{\partial x_k} + \frac{\tau_{PSPG}}{\rho} \frac{\partial \delta p}{\partial x_i} \right) \left(\rho \frac{\partial v_i}{\partial t} + \rho v_j \frac{\partial v_i}{\partial x_j} - \frac{\partial \sigma_{ij}(\mathbf{v}^f, p^f)}{\partial x_j} - \rho g_i \right) + \int_{\Omega} d\Omega \delta p^f \frac{\partial v_j}{\partial x_j} = \\
&= \int_{\Omega} d\Omega \left(\delta v_i + \tau_{SUPG} v_k \frac{\partial \delta v_i}{\partial x_k} + \frac{\tau_{PSPG}}{\rho} \frac{\partial \delta p}{\partial x_i} \right) \left(\rho \frac{\partial v_i}{\partial t} + \rho v_j \frac{\partial v_i}{\partial x_j} - \rho g_i \right) + \int_{\Omega} d\Omega \frac{\partial \delta v_i^f}{\partial x_j} \sigma_{ij}(\mathbf{v}^f, p^f) \\
&- \int_{\Omega} d\Omega \left(\tau_{SUPG} v_k \frac{\partial \delta v_i}{\partial x_k} + \frac{\tau_{PSPG}}{\rho} \frac{\partial \delta p}{\partial x_i} \right) \frac{\partial \sigma_{ij}(\mathbf{v}^f, p^f)}{\partial x_j} + \int_{\Omega} d\Omega \delta p^f \frac{\partial v_j}{\partial x_j} - \int_{\Gamma_{SF}} d\Gamma \delta v_i \sigma_{ij}(\mathbf{v}^f, p^f) n_j + \alpha \int_{\Gamma} d\Gamma \delta v_i v_i
\end{aligned} \tag{A.10}$$

This residual (Equation A.10) can be split into advective, body force, stress, stabilized stress, incompressibility and surface terms (Equation A.11):

$$\begin{aligned}
R_{fluid}\{\mathbf{v}^f, p^f, \Phi\} &= R_{v_y}^{Adv} + R_{v_z}^{Adv} + R_{p^f}^{Adv} + R_{v_y}^{BF} + R_{v_z}^{BF} + R_{p^f}^{BF} + R_{v_y}^{Str} + R_{v_z}^{Str} + R_{p^f}^{Str} \\
&+ R_{v_y}^{Stabstr} + R_{v_z}^{Stabstr} + R_{p^f}^{Stabstr} + R_{v_y}^{IC} + R_{v_z}^{IC} + R_{p^f}^{IC} + \bar{R}_{v_y}^{Surf} + \bar{R}_{v_z}^{Surf} + \bar{R}_{p^f}^{Surf} + R_{v_y}^{Dyn} + R_{v_z}^{Dyn} + R_{p^f}^{Dyn}
\end{aligned} \tag{A.11}$$

Each term can be written as follows (Equations A.12 to A.32), where $|w_{fluid}\rangle$ is a basis about a certain trial solution, $A = v_i \frac{\partial N}{\partial x_i}$ are the advective shape functions, as a function of the shape function vector N , and $\sigma(\mathbf{v}^f, p^f) = -p\delta_{ij} + \bar{\sigma}_{ij}(\mathbf{v}^f)$ is the total stress tensor.

$$R_{v_y}^{Adv} = \int_{\Omega} d\Omega (\delta v_y + \tau_{SUPG} v_k \frac{\partial \delta v_y}{\partial x_k}) \rho v_j \frac{\partial v_y}{\partial x_j} = \int_{\Omega} d\Omega (N + \tau_{SUPG} v_k A)^T \rho v_j \frac{\partial v_y}{\partial x_j} |v_y\rangle \tag{A.12}$$

$$R_{v_z}^{Adv} = \int_{\Omega} d\Omega (N + \tau_{SUPG} v_k A)^T \rho v_j \frac{\partial v_z}{\partial x_j} |v_z\rangle \tag{A.13}$$

$$R_{p^f}^{Adv} = \int_{\Omega} d\Omega \frac{\tau_{PSPG}}{\rho} \frac{\partial \delta p}{\partial x_i} \rho v_j \frac{\partial v_i}{\partial x_j} = \int_{\Omega} d\Omega \tau_{PSPG} \frac{\partial N^T}{\partial x_i} v_j \frac{\partial v_i}{\partial x_j} |p^f\rangle \tag{A.14}$$

$$R_{v_y}^{BF} = - \int_{\Omega} d\Omega (N + \tau_{SUPG} v_k A)^T \rho g_y |v_y\rangle \tag{A.15}$$

$$R_{v_z}^{BF} = - \int_{\Omega} d\Omega (N + \tau_{SUPG} v_k A)^T \rho g_z |v_z\rangle \tag{A.16}$$

$$R_{p^f}^{BF} = - \int_{\Omega} d\Omega \tau_{PSPG} \frac{\partial N^T}{\partial x_i} g_i |p^f\rangle \tag{A.17}$$

$$R_{v_y}^{Str} = \int_{\Omega} d\Omega \frac{\partial \delta v_i}{\partial x_j} \sigma_{yj}(\mathbf{v}^f, p^f) = - \int_{\Omega} d\Omega \frac{\partial N^T}{\partial y} p^f |v_y\rangle + \int_{\Omega} d\Omega \frac{\partial N^T}{\partial x_j} \bar{\sigma}_{yj}(\mathbf{v}^f) |v_y\rangle \tag{A.18}$$

$$R_{v_z}^{Str} = - \int_{\Omega} d\Omega \frac{\partial N^T}{\partial z} p^f |v_z\rangle + \int_{\Omega} d\Omega \frac{\partial N^T}{\partial x_j} \bar{\sigma}_{zj}(\mathbf{v}^f) |v_z\rangle \tag{A.19}$$

$$R_{p^f}^{Str} = 0 \tag{A.20}$$

$$R_{v_y}^{Stabstr} = - \int_{\Omega} d\Omega \tau_{PSPG} v_k \frac{\partial \delta v_y}{\partial x_k} \frac{\partial \sigma_{yj}(\mathbf{v}^f, p^f)}{\partial x_j} = \int_{\Omega} d\Omega \tau_{PSPG} A^T \left(\frac{\partial p^f}{\partial y} - \frac{\partial \bar{\sigma}_{yj}(\mathbf{v}^f)}{\partial x_j} \right) |v_y\rangle \tag{A.21}$$

$$R_{v_z}^{Stabstr} = \int_{\Omega} d\Omega \tau_{PSPG} A^T \left(\frac{\partial p^f}{\partial z} - \frac{\partial \bar{\sigma}_{zj}(\mathbf{v}^f)}{\partial x_j} \right) |v_z\rangle \tag{A.22}$$

$$R_{p^f}^{Stabstr} = \int_{\Omega} d\Omega \frac{\tau_{PSPG}}{\rho} \frac{\partial N^T}{\partial x_i} \frac{\partial p^f}{\partial x_i} |p^f\rangle - \int_{\Omega} d\Omega \frac{\tau_{PSPG}}{\rho} \frac{\partial N^T}{\partial x_i} \frac{\partial \bar{\sigma}_{ij}(\mathbf{v}^f)}{\partial x_j} |p^f\rangle \tag{A.23}$$

$$R_{v_y}^{IC} = 0 \quad (\text{A.24})$$

$$R_{v_z}^{IC} = 0 \quad (\text{A.25})$$

$$R_{p^f}^{IC} = \int_{\Omega} d\Omega N^T \frac{\partial v_i}{\partial x_i} |p^f\rangle \quad (\text{A.26})$$

$$\overline{R}_{v_y}^{Surf} = \alpha \int_{\Gamma_I} d\Gamma N^T v_y |v_y\rangle \quad (\text{A.27})$$

$$\overline{R}_{v_z}^{Surf} = \alpha \int_{\Gamma_I} d\Gamma \delta N^T v_z |v_z\rangle \quad (\text{A.28})$$

$$\overline{R}_{p^f}^{Surf} = 0 \quad (\text{A.29})$$

$$R_{v_y}^{Dyn} = \int_{\Omega} d\Omega (\delta v_y + \tau_{SUPG} v_k \frac{\partial \delta v_y}{\partial x_k}) \rho \frac{\partial v_y}{\partial t} = \int_{\Omega} d\Omega (N + \tau_{SUPG} v_k A)^T \rho \frac{\partial v_y}{\partial t} |v_y\rangle \quad (\text{A.30})$$

$$R_{v_z}^{Dyn} = \int_{\Omega} d\Omega (N + \tau_{SUPG} v_k A)^T \rho \frac{\partial v_z}{\partial t} |v_z\rangle \quad (\text{A.31})$$

$$R_{p^f}^{Dyn} = \int_{\Omega} d\Omega \frac{\tau_{PSPG}}{\rho} \frac{\partial \delta p}{\partial x_i} \rho \frac{\partial v_i}{\partial t} = \int_{\Omega} d\Omega \tau_{PSPG} \frac{\partial N^T}{\partial x_i} \frac{\partial \delta v_i}{\partial t} |p^f\rangle \quad (\text{A.32})$$

With $e_{ij}(\mathbf{v}) = \frac{1}{2}(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i})$. Corresponding Jacobian terms are summarized as follows (Equations A.33 to A.77):

$$\frac{\partial R_{v_y}^{Adv}}{\partial v_y} = \int_{\Omega} d\Omega (N + \tau_{SUPG} A)^T \rho (A + \frac{\partial v_y}{\partial y} N) |v_y\rangle \langle v_y| + \int_{\Omega} d\Omega \tau_{SUPG} \rho v_j \frac{\partial v_y}{\partial x_j} \frac{\partial N^T}{\partial y} N |v_y\rangle \langle v_y| \quad (\text{A.33})$$

$$\frac{\partial R_{v_y}^{Adv}}{\partial v_z} = \int_{\Omega} d\Omega (N + \tau_{SUPG} A)^T \rho \frac{\partial v_y}{\partial z} N |v_y\rangle \langle v_z| + \int_{\Omega} d\Omega \tau_{SUPG} \rho v_j \frac{\partial v_y}{\partial x_j} \frac{\partial N^T}{\partial z} N |v_y\rangle \langle v_z| \quad (\text{A.34})$$

$$\frac{\partial R_{v_y}^{Adv}}{\partial p^f} = 0 \quad (\text{A.35})$$

$$\frac{\partial R_{v_z}^{Adv}}{\partial v_y} = \int_{\Omega} d\Omega (N + \tau_{SUPG} A)^T \rho \frac{\partial v_z}{\partial y} N |v_z\rangle \langle v_y| + \int_{\Omega} d\Omega \tau_{SUPG} \rho v_j \frac{\partial v_z}{\partial x_j} \frac{\partial N^T}{\partial y} N |v_z\rangle \langle v_y| \quad (\text{A.36})$$

$$\frac{\partial R_{v_z}^{Adv}}{\partial v_z} = \int_{\Omega} d\Omega (N + \tau_{SUPG} A)^T \rho (A + \frac{\partial v_z}{\partial z} N) |v_z\rangle \langle v_z| + \int_{\Omega} d\Omega \tau_{SUPG} \rho v_j \frac{\partial v_z}{\partial x_j} \frac{\partial N^T}{\partial z} N |v_z\rangle \langle v_z| \quad (\text{A.37})$$

$$\frac{\partial R_{v_z}^{Adv}}{\partial p^f} = 0 \quad (\text{A.38})$$

$$\frac{\partial R_{p^f}^{Adv}}{\partial v_y} = \int_{\Omega} d\Omega \tau_{PSPG} \left(\frac{\partial v_y}{\partial y} \frac{\partial N}{\partial y} + \frac{\partial v_z}{\partial y} \frac{\partial N}{\partial z} \right)^T N |p^f\rangle \langle v_y| + \int_{\Omega} d\Omega \tau_{PSPG} \frac{\partial N^T}{\partial y} A |p^f\rangle \langle v_y| \quad (\text{A.39})$$

$$\frac{\partial R_{p^f}^{Adv}}{\partial v_z} = \int_{\Omega} d\Omega \tau_{PSPG} \left(\frac{\partial v_y}{\partial z} \frac{\partial N}{\partial y} + \frac{\partial v_z}{\partial z} \frac{\partial N}{\partial z} \right)^T N |p^f\rangle \langle v_z| + \int_{\Omega} d\Omega \tau_{PSPG} \frac{\partial N^T}{\partial z} A |p^f\rangle \langle v_z| \quad (\text{A.40})$$

$$\frac{\partial R_{p^f}^{Adv}}{\partial p^f} = 0 \quad (\text{A.41})$$

$$\frac{\partial R_{v_y}^{Str}}{\partial v_y} = \int_{\Omega} d\Omega \frac{\partial N^T}{\partial x_j} \frac{\partial \overline{\sigma}_{yj}(\mathbf{v}^f)}{\partial v_y} |v_y\rangle \langle v_y| \quad (\text{A.42})$$

$$\frac{\partial R_{v_y}^{Str}}{\partial v_z} = \int_{\Omega} d\Omega \frac{\partial N^T}{\partial x_j} \frac{\partial \overline{\sigma}_{yj}(\mathbf{v}^f)}{\partial v_z} |v_y\rangle \langle v_z| \quad (\text{A.43})$$

$$\frac{\partial R_{v_y}^{Str}}{\partial p^f} = - \int_{\Omega} d\Omega \frac{\partial N^T}{\partial y} N |v_y\rangle \langle p^f| \quad (\text{A.44})$$

$$\frac{\partial R_{v_z}^{Str}}{\partial v_y} = \int_{\Omega} d\Omega \frac{\partial N^T}{\partial x_j} \frac{\partial \overline{\sigma}_{zj}(\mathbf{v}^f)}{\partial v_y} |v_z\rangle \langle v_y| \quad (\text{A.45})$$

$$\frac{\partial R_{v_z}^{Str}}{\partial v_z} = \int_{\Omega} d\Omega \frac{\partial N^T}{\partial x_j} \frac{\partial \bar{\sigma}_{zj}(\mathbf{v}^f)}{\partial v_z} |v_z\rangle\langle v_z| \quad (\text{A.46})$$

$$\frac{\partial R_{v_z}^{Str}}{\partial p^f} = - \int_{\Omega} d\Omega \frac{\partial N^T}{\partial z} N |v_z\rangle\langle p^f| \quad (\text{A.47})$$

$$\frac{\partial R_{v_y}^{Stabstr}}{\partial v_y} = - \int_{\Omega} d\Omega \tau_{SUPG} A^T \frac{\partial}{\partial v_y} \left(\frac{\partial \bar{\sigma}_{yj}(\mathbf{v}^f)}{\partial x_j} \right) |v_y\rangle\langle v_y| + - \int_{\Omega} d\Omega \tau_{SUPG} \left(-\frac{\partial p}{\partial y} + \frac{\partial \bar{\sigma}_{yj}(\mathbf{v}^f)}{\partial x_j} \right) \frac{\partial N^T}{\partial y} N |v_y\rangle\langle v_y| \quad (\text{A.48})$$

$$\frac{\partial R_{v_y}^{Stabstr}}{\partial v_z} = - \int_{\Omega} d\Omega \tau_{SUPG} A^T \frac{\partial}{\partial v_z} \left(\frac{\partial \bar{\sigma}_{yj}(\mathbf{v}^f)}{\partial x_j} \right) |v_y\rangle\langle v_z| + - \int_{\Omega} d\Omega \tau_{SUPG} \left(-\frac{\partial p}{\partial y} + \frac{\partial \bar{\sigma}_{yj}(\mathbf{v}^f)}{\partial x_j} \right) \frac{\partial N^T}{\partial z} N |v_y\rangle\langle v_z| \quad (\text{A.49})$$

$$\frac{\partial R_{v_y}^{Stabstr}}{\partial p^f} = \int_{\Omega} d\Omega \tau_{SUPG} A^T \frac{\partial N}{\partial y} |v_y\rangle\langle p^f| \quad (\text{A.50})$$

$$\frac{\partial R_{v_z}^{Stabstr}}{\partial v_y} = - \int_{\Omega} d\Omega \tau_{SUPG} A^T \frac{\partial}{\partial v_y} \left(\frac{\partial \bar{\sigma}_{zj}(\mathbf{v}^f)}{\partial x_j} \right) |v_z\rangle\langle v_y| + - \int_{\Omega} d\Omega \tau_{SUPG} \left(-\frac{\partial p}{\partial z} + \frac{\partial \bar{\sigma}_{zj}(\mathbf{v}^f)}{\partial x_j} \right) \frac{\partial N^T}{\partial y} N |v_z\rangle\langle v_y| \quad (\text{A.51})$$

$$\frac{\partial R_{v_z}^{Stabstr}}{\partial v_z} = - \int_{\Omega} d\Omega \tau_{SUPG} A^T \frac{\partial}{\partial v_z} \left(\frac{\partial \bar{\sigma}_{zj}(\mathbf{v}^f)}{\partial x_j} \right) |v_z\rangle\langle v_z| + - \int_{\Omega} d\Omega \tau_{SUPG} \left(-\frac{\partial p}{\partial z} + \frac{\partial \bar{\sigma}_{zj}(\mathbf{v}^f)}{\partial x_j} \right) \frac{\partial N^T}{\partial z} N |v_z\rangle\langle v_z| \quad (\text{A.52})$$

$$\frac{\partial R_{v_z}^{Stabstr}}{\partial p^f} = \int_{\Omega} d\Omega \tau_{SUPG} A^T \frac{\partial N}{\partial z} |v_z\rangle\langle p^f| \quad (\text{A.53})$$

$$\frac{\partial R_{p^f}^{Stabstr}}{\partial v_y} = - \int_{\Omega} d\Omega \frac{\tau_{PSPG}}{\rho} \frac{\partial N^T}{\partial x_i} \frac{\partial}{\partial v_y} \left(\frac{\partial \bar{\sigma}_{ij}(\mathbf{v}^f)}{\partial x_j} \right) |p^f\rangle\langle v_y| \quad (\text{A.54})$$

$$\frac{\partial R_{p^f}^{Stabstr}}{\partial v_z} = - \int_{\Omega} d\Omega \frac{\tau_{PSPG}}{\rho} \frac{\partial N^T}{\partial x_i} \frac{\partial}{\partial v_z} \left(\frac{\partial \bar{\sigma}_{ij}(\mathbf{v}^f)}{\partial x_j} \right) |p^f\rangle\langle v_z| \quad (\text{A.55})$$

$$\frac{\partial R_{p^f}^{Stabstr}}{\partial p^f} = \int_{\Omega} d\Omega \frac{\tau_{PSPG}}{\rho} \frac{\partial N^T}{\partial x_i} \frac{\partial N}{\partial x_i} |p^f\rangle\langle p^f| \quad (\text{A.56})$$

$$\frac{\partial R_{v_y}^{IC}}{\partial v_y} = 0 \quad (\text{A.57})$$

$$\frac{\partial R_{v_y}^{IC}}{\partial v_z} = 0 \quad (\text{A.58})$$

$$\frac{\partial R_{v_y}^{IC}}{\partial p^f} = 0 \quad (\text{A.59})$$

$$\frac{\partial R_{v_z}^{IC}}{\partial v_y} = 0 \quad (\text{A.60})$$

$$\frac{\partial R_{v_z}^{IC}}{\partial v_z} = 0 \quad (\text{A.61})$$

$$\frac{\partial R_{v_z}^{IC}}{\partial p^f} = 0 \quad (\text{A.62})$$

$$\frac{\partial R_{p^f}^{IC}}{\partial v_y} = \int_{\Omega} d\Omega N^T \frac{\partial N}{\partial y} |p^f\rangle\langle v_y| \quad (\text{A.63})$$

$$\frac{\partial R_{p^f}^{IC}}{\partial v_z} = \int_{\Omega} d\Omega N^T \frac{\partial N}{\partial z} |p^f\rangle\langle v_z| \quad (\text{A.64})$$

$$\frac{\partial R_{p^f}^{IC}}{\partial p^f} = 0 \quad (\text{A.65})$$

$$\frac{\partial \bar{R}_{v_y}^{surf}}{\partial v_y} = \alpha \int_{\Gamma_i} d\Gamma N^T N |v_y\rangle \langle v_y| \quad (\text{A.66})$$

$$\frac{\partial \bar{R}_{v_y}^{surf}}{\partial v_z} = 0 \quad (\text{A.67})$$

$$\frac{\partial \bar{R}_{v_y}^{surf}}{\partial p^f} = 0 \quad (\text{A.68})$$

$$\frac{\partial \bar{R}_{v_z}^{surf}}{\partial v_y} = 0 \quad (\text{A.69})$$

$$\frac{\partial \bar{R}_{v_z}^{surf}}{\partial v_z} = \alpha \int_{\Gamma_i} d\Gamma N^T N |v_z\rangle \langle v_z| \quad (\text{A.70})$$

$$\frac{\partial \bar{R}_{v_z}^{surf}}{\partial p^f} = 0 \quad (\text{A.71})$$

$$\frac{\partial R_{v_y}^{dyn}}{\partial v_y} = \int_{\Omega} d\Omega \tau_{SUPG} \rho \frac{\partial v_y}{\partial t} \frac{\partial N^T}{\partial y} N |v_y\rangle \langle v_y| \quad (\text{A.72})$$

$$\frac{\partial R_{v_y}^{dyn}}{\partial v_z} = \int_{\Omega} d\Omega \tau_{SUPG} \rho \frac{\partial v_y}{\partial t} \frac{\partial N^T}{\partial z} N |v_y\rangle \langle v_z| \quad (\text{A.73})$$

$$\frac{\partial R_{v_y}^{dyn}}{\partial p^f} = 0 \quad (\text{A.74})$$

$$\frac{\partial R_{v_z}^{dyn}}{\partial v_y} = \int_{\Omega} d\Omega \tau_{SUPG} \rho \frac{\partial v_z}{\partial t} \frac{\partial N^T}{\partial y} N |v_z\rangle \langle v_y| \quad (\text{A.75})$$

$$\frac{\partial R_{v_z}^{dyn}}{\partial v_z} = \int_{\Omega} d\Omega \tau_{SUPG} \rho \frac{\partial v_z}{\partial t} \frac{\partial N^T}{\partial z} N |v_z\rangle \langle v_z| \quad (\text{A.76})$$

$$\frac{\partial R_{v_z}^{dyn}}{\partial p^f} = 0 \quad (\text{A.77})$$

$$\frac{\partial R_{fluid}}{\partial u^l} = \vec{0} \quad (\text{A.78})$$

$$\frac{\partial R_{fluid}}{\partial F_y} = \vec{0} \quad (\text{A.79})$$

$$\frac{\partial R_{fluid}}{\partial F_z} = \vec{0} \quad (\text{A.80})$$

$$\frac{\partial R_{fluid}}{\partial M_x} = \vec{0} \quad (\text{A.81})$$

$$\frac{\partial R_{fluid}}{\partial \eta} = \vec{0} \quad (\text{A.82})$$

$$\frac{\partial R_{fluid}}{\partial \Phi} = \left. \frac{\partial R_{fluid}}{\partial \Phi} \right|_{FD}^{**} \quad (\text{A.83})$$

And the mass matrix terms are (Equation A.85):

$$\mathbf{m} = \mathbf{m}_{v_y}^{v_y} + \mathbf{m}_{v_z}^{v_z} + \mathbf{m}_{p^f}^{v_y} + \mathbf{m}_{p^f}^{v_z} \quad (\text{A.84})$$

$$(\text{A.85})$$

$$\mathbf{m}_{v_y}^{v_y} = \int_{\Omega} d\Omega (N + \tau_{SUPG} A)^T N |v_y\rangle \langle v_y| \quad (\text{A.86})$$

$$(\text{A.87})$$

$$\mathbf{m}_{v_z}^{v_z} = \int_{\Omega} d\Omega (N + \tau_{SUPGA})^T N |v_z\rangle \langle v_z| \quad (\text{A.88})$$

$$\mathbf{m}_{p^f}^{v_y} = \int_{\Omega} d\Omega \tau_{PSPG} \frac{\partial N^T}{\partial y} N |p^f\rangle \langle v_y| \quad (\text{A.89})$$

$$\mathbf{m}_{p^f}^{v_z} = \int_{\Omega} d\Omega \tau_{PSPG} \frac{\partial N^T}{\partial z} N |p^f\rangle \langle v_z| \quad (\text{A.90})$$

A.1.3. Level Set projections terms

These are the terms associated with the Level Set projection scheme. From Equation 4.12:

$$R_{\Phi}\{u^l, \Phi, \Phi_0\} = \int_{\Omega} d\Omega N^T (\Phi(x_i) - \Phi^{ref}(x_i - Z_i(x_i)u_{sect}^l)) = 0 \quad (\text{A.91})$$

$$\frac{\partial R_{\Phi}}{\partial u^l} = \left[\vec{0} \cdot \int_{\Omega} d\Omega N^T \frac{\partial \Phi}{\partial x_i'} Z_i(x_i) \cdot \vec{0} \right] \quad (\text{A.92})$$

$$\frac{\partial R_{\Phi}}{\partial v^f} = \vec{0} \quad (\text{A.93})$$

$$\frac{\partial R_{\Phi}}{\partial p^f} = \vec{0} \quad (\text{A.94})$$

$$\frac{\partial R_{\Phi}}{\partial \Phi} = \int_{\Omega} d\Omega N^T N \quad (\text{A.95})$$

$$\frac{\partial R_{\Phi}}{\partial F_y} = \vec{0} \quad (\text{A.96})$$

$$\frac{\partial R_{\Phi}}{\partial F_z} = \vec{0} \quad (\text{A.97})$$

$$\frac{\partial R_{\Phi}}{\partial M_x} = \vec{0} \quad (\text{A.98})$$

$$\frac{\partial R_{\Phi}}{\partial r_i} = \vec{0} \quad (\text{A.99})$$

Where $\mathbf{x}' = \mathbf{x} + \mathbf{Z}\mathbf{u}_{sect}^l$.

A.1.4. Aerodynamic forces terms

These are the terms associated with the aerodynamic forces. From Equation 4.13:

$$R_{F_y}\{v^f, p^f, \Phi, D\} = \oint_{\Gamma_{SF}} \left[dF_y + d\Gamma_{SF} \left(\rho v \frac{\partial \vec{v}_t}{\partial \vec{n}} n_z - p^f n_y \right) \right] = \oint_{\Gamma_{SF}} \left[dF_y + d\Gamma_{SF} \left(\rho v \left(\frac{\partial v_y}{\partial z} - \frac{\partial v_z}{\partial y} \right) n_z - p^f n_y \right) \right] \quad (\text{A.100})$$

$$R_{F_z}\{v^f, p^f, \Phi, L\} = \oint_{\Gamma_{SF}} \left[dF_z - d\Gamma_{SF} \left(\rho v \frac{\partial \vec{v}_t}{\partial \vec{n}} n_y + p^f n_z \right) \right] = \oint_{\Gamma_{SF}} \left[dF_z - d\Gamma_{SF} \left(\rho v \left(\frac{\partial v_y}{\partial z} - \frac{\partial v_z}{\partial y} \right) n_y + p^f n_z \right) \right] \quad (\text{A.101})$$

$$\begin{aligned} R_{M_x}\{v^f, p^f, \Phi, L\} &= \oint_{\Gamma_{SF}} \left[dM_x - d\Gamma_{SF} \left(\rho v \frac{\partial \vec{v}_t}{\partial \vec{n}} n_y + p^f n_z \right) (y - y_0) - d\Gamma_{SF} \left(\rho v \frac{\partial \vec{v}_t}{\partial \vec{n}} n_z - p^f n_y \right) (z - z_0) \right] = \\ &= \oint_{\Gamma_{SF}} d\Gamma_{SF} \left[dM_x - \left(\rho v \left(\frac{\partial v_y}{\partial z} - \frac{\partial v_z}{\partial y} \right) n_y + p^f n_z \right) (y - y_0) - \left(\rho v \left(\frac{\partial v_y}{\partial z} - \frac{\partial v_z}{\partial y} \right) n_z - p^f n_y \right) (z - z_0) \right] \end{aligned} \quad (\text{A.102})$$

$$\frac{\partial R_{F_y}}{\partial u^l} = \vec{0} \quad (\text{A.103})$$

$$\frac{\partial R_{F_y}}{\partial v_y} = \oint_{\Gamma_{SF}} d\Gamma_{SF} \rho v \frac{\partial N}{\partial z} n_z \quad (\text{A.104})$$

$$\frac{\partial R_{F_y}}{\partial v_z} = - \oint_{\Gamma_{SF}} d\Gamma_{SF} \rho v \frac{\partial N}{\partial y} n_z \quad (\text{A.105})$$

$$\frac{\partial R_{F_y}}{\partial p^f} = - \oint_{\Gamma_{SF}} d\Gamma_{SF} N n_y \quad (\text{A.106})$$

$$\frac{\partial R_{F_y}}{\partial \Phi} = \left. \frac{\partial R_{F_y}}{\partial \Phi} \right|_{FD}^{**} \quad (\text{A.107})$$

$$\frac{\partial R_{F_y}}{\partial F_z} = \vec{0} \quad (\text{A.108})$$

$$\frac{\partial R_{F_y}}{\partial F_y} = \begin{bmatrix} \cdot \\ \cdot \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ \cdot \\ \cdot \end{bmatrix} \quad (\text{A.109})$$

$$\frac{\partial R_{F_y}}{\partial M_x} = \vec{0} \quad (\text{A.110})$$

$$\frac{\partial R_{F_y}}{\partial r_i} = \vec{0} \quad (\text{A.111})$$

$$\frac{\partial R_{F_z}}{\partial u^l} = \vec{0} \quad (\text{A.112})$$

$$\frac{\partial R_{F_z}}{\partial v_y} = - \oint_{\Gamma_{SF}} d\Gamma_{SF} \rho v \frac{\partial N}{\partial z} n_y \quad (\text{A.113})$$

$$\frac{\partial R_{F_z}}{\partial v_z} = \oint_{\Gamma_{SF}} d\Gamma_{SF} \rho v \frac{\partial N}{\partial y} n_y \quad (\text{A.114})$$

$$\frac{\partial R_{F_z}}{\partial p^f} = - \oint_{\Gamma_{SF}} d\Gamma_{SF} N n_z \quad (\text{A.115})$$

$$\frac{\partial R_{F_z}}{\partial \Phi} = \left. \frac{\partial R_{F_z}}{\partial \Phi} \right|_{FD}^{**} \quad (\text{A.116})$$

$$\frac{\partial R_{F_z}}{\partial F_y} = \vec{0} \quad (\text{A.117})$$

$$\frac{\partial R_{F_z}}{\partial F_z} = \begin{bmatrix} \cdot \\ \cdot \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ \cdot \\ \cdot \end{bmatrix} \quad (\text{A.118})$$

$$\frac{\partial R_{F_z}}{\partial M_x} = \vec{0} \quad (\text{A.119})$$

$$\frac{\partial R_{F_z}}{\partial r_i} = \vec{0} \quad (\text{A.120})$$

$$\frac{\partial R_{M_x}}{\partial u^l} = \vec{0} \quad (\text{A.121})$$

$$\frac{\partial R_{M_x}}{\partial v_y} = - \oint_{\Gamma_{SF}} d\Gamma_{SF} \rho v \frac{\partial N}{\partial z} n_y (y - y_0) - \oint_{\Gamma_{SF}} d\Gamma_{SF} \rho v \frac{\partial N}{\partial z} n_z (z - z_0) \quad (\text{A.122})$$

$$\frac{\partial R_{M_x}}{\partial v_z} = \oint_{\Gamma_{SF}} d\Gamma_{SF} \rho v \frac{\partial N}{\partial y} n_y (y - y_0) + \oint_{\Gamma_{SF}} d\Gamma_{SF} \rho v \frac{\partial N}{\partial y} n_z (z - z_0) \quad (\text{A.123})$$

$$\frac{\partial R_{M_x}}{\partial p^f} = - \oint_{\Gamma_{SF}} d\Gamma_{SF} n_z N (y - y_0) + \oint_{\Gamma_{SF}} d\Gamma_{SF} n_y N (z - z_0) \quad (\text{A.124})$$

$$\frac{\partial R_{M_x}}{\partial \Phi} = \frac{\partial R_{M_x}}{\partial \Phi} \Big|_{FD}^{**} \quad (\text{A.125})$$

$$\frac{\partial R_{M_x}}{\partial F_y} = \vec{0} \quad (\text{A.126})$$

$$\frac{\partial R_{M_x}}{\partial F_z} = \vec{0} \quad (\text{A.127})$$

$$\frac{\partial R_{M_x}}{\partial M_x} = \begin{bmatrix} \vdots \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ \vdots \end{bmatrix} \quad (\text{A.128})$$

$$\frac{\partial R_{M_x}}{\partial r_l} = 0 \quad (\text{A.129})$$

** The computation of $\left\{ \frac{\partial R_{fluid}}{\partial \Phi}, \frac{\partial R_{F_y}}{\partial \Phi}, \frac{\partial R_{F_z}}{\partial \Phi}, \frac{\partial R_{M_x}}{\partial \Phi} \right\}$ terms has been performed using finite differences in order to achieve higher accuracy on the sensitivity analysis [39] and better convergence rate on the forward analysis. The original approach was to find an analytical definition of these terms as well, accounting for the influence of the Level Set on $\{\mathbf{v}^f, p^f, F_y, F_z, M_x\}$ as a function of changes of the normal vector, boundary length, and boundary position, for boundary integrals on intersected elements, and changes of fluid area and fluid area position, for area integrals on intersected elements. An example of this analytical derivation for a boundary integral term is described on Equation A.130 and an example for an area integral is described on Equation A.131.

$$\frac{\partial R_{M_x}}{\partial \Phi} = \frac{\partial R_{M_x}}{\partial n_i} \frac{\partial n_i}{\partial \Phi} + \frac{\partial R_{M_x}}{\partial \Gamma} \frac{\partial \Gamma}{\partial \Phi} + R_{M_x} \left\{ \frac{\partial p}{\partial \mathbf{x}}, \frac{\partial v_y}{\partial \mathbf{x}}, \frac{\partial v_z}{\partial \mathbf{x}} \right\}_{GP} \frac{\partial \mathbf{x}}{\partial \Phi} \Big|_{GP} \quad (\text{A.130})$$

$$\frac{\partial R_{v_y}^{Str}}{\partial \Phi} = \frac{\partial R_{v_y}^{Str}}{\partial \Omega} \frac{\partial \Omega}{\partial \Phi} + R_{v_y}^{Str} \left\{ \frac{\partial p}{\partial \mathbf{x}}, \frac{\partial v_y}{\partial \mathbf{x}}, \frac{\partial v_z}{\partial \mathbf{x}} \right\}_{GP} \frac{\partial \mathbf{x}}{\partial \Phi} \Big|_{GP} \quad (\text{A.131})$$

Where:

$$\frac{\partial n_y}{\partial \Phi} = \frac{n_z^2}{\sqrt{(\frac{\partial \Phi}{\partial y})^2 + (\frac{\partial \Phi}{\partial z})^2}} \frac{\partial N}{\partial y} - \frac{n_y n_z}{\sqrt{(\frac{\partial \Phi}{\partial y})^2 + (\frac{\partial \Phi}{\partial z})^2}} \frac{\partial N}{\partial z} \quad (\text{A.132})$$

$$\frac{\partial n_z}{\partial \Phi} = - \frac{n_y n_z}{\sqrt{(\frac{\partial \Phi}{\partial y})^2 + (\frac{\partial \Phi}{\partial z})^2}} \frac{\partial N}{\partial y} + \frac{n_y^2}{\sqrt{(\frac{\partial \Phi}{\partial y})^2 + (\frac{\partial \Phi}{\partial z})^2}} \frac{\partial N}{\partial z} \quad (\text{A.133})$$

$$\frac{\partial R_{M_x}}{\partial \Gamma} = \frac{R_{M_x}}{\Gamma} \quad (\text{A.134})$$

$$\frac{\partial \Gamma}{\partial \Phi} = \frac{\partial \Gamma}{\partial \mathbf{x}_{int}^i} \frac{\partial \mathbf{x}_{int}^i}{\partial \Phi} \quad (\text{A.135})$$

$$\frac{\partial R_{v_y}^{Str}}{\partial \Omega} = \frac{R_{v_y}^{Str}}{\Omega} \quad (\text{A.136})$$

$$\frac{\partial \Omega}{\partial \Phi} = \frac{\partial \Omega}{\partial \mathbf{x}} \Big|_{GP} \frac{\partial \mathbf{x}}{\partial \Phi} \Big|_{GP} \quad (\text{A.137})$$

$$\frac{\partial \mathbf{x}}{\partial \Phi} \Big|_{GP} = \frac{\partial \mathbf{x}}{\partial \mathbf{x}_{int}^i} \Big|_{GP} \frac{\partial \mathbf{x}_{int}^i}{\partial \Phi} \quad (\text{A.138})$$

$$\mathbf{x}_{int}^i = \mathbf{x}_m + \frac{\Phi_m}{\Phi_m - \Phi_n} (\mathbf{x}_n - \mathbf{x}_m) \quad (\text{A.139})$$

$$\frac{\partial \mathbf{x}_{int}^i}{\partial \Phi} = \begin{cases} \left(\frac{1}{\Phi_m - \Phi_n} - \frac{\Phi_m}{(\Phi_m - \Phi_n)^2} \right) (\mathbf{x}_n - \mathbf{x}_m) & \text{if } node = m \\ \frac{\Phi_m}{(\Phi_m - \Phi_n)^2} (\mathbf{x}_n - \mathbf{x}_m) & \text{if } node = n \\ 0 & \text{Otherwise} \end{cases} \quad (\text{A.140})$$

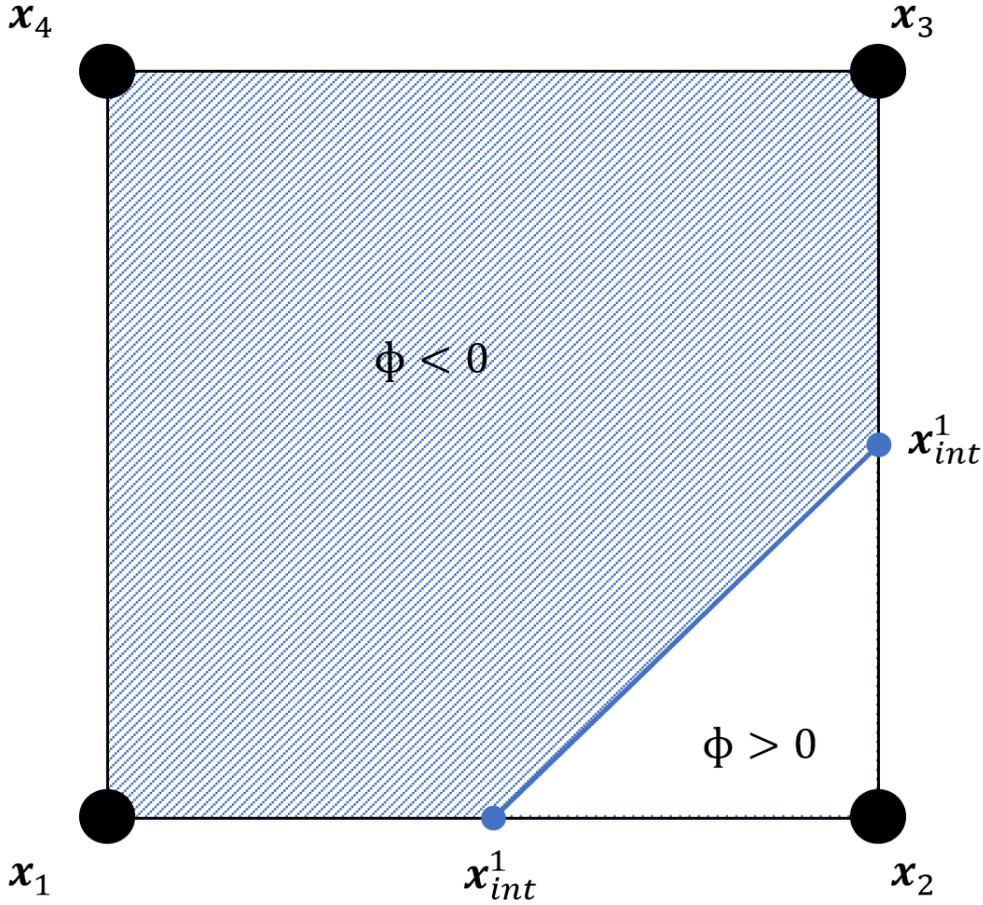
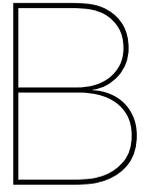


Figure A.1: Example of intersected element. Position of intersected nodes

However the analytical derivation of these terms had a difference of up to 20% with respect to their finite difference counterparts, and it showed a poor performance on the calculation of the sensitivities and the forward analysis convergence rate. Hence the finite difference derivation of these terms was finally chosen.



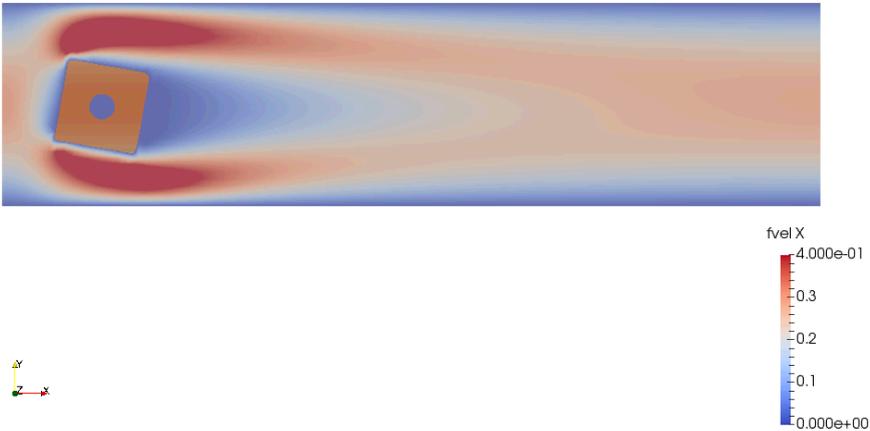
Verification of the monolithic Newton solver. Visual results

The following results have been obtained for the fluid horizontal velocity distribution at each cross-section on the verification example (Section 4.3). Displacements and twist along the beam length are also appreciated (Figures B.1 to B.3).

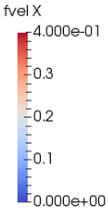
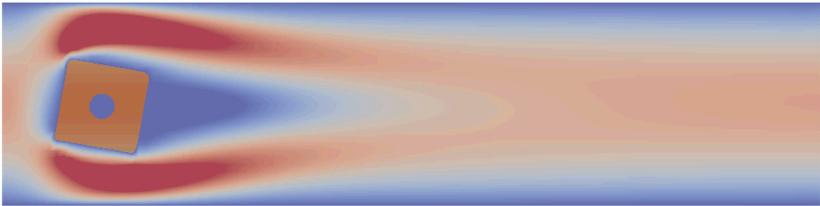
B.1. Clamped section: Square shape



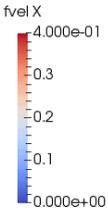
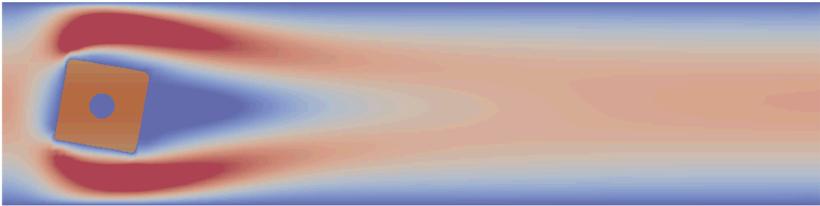
(a)



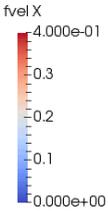
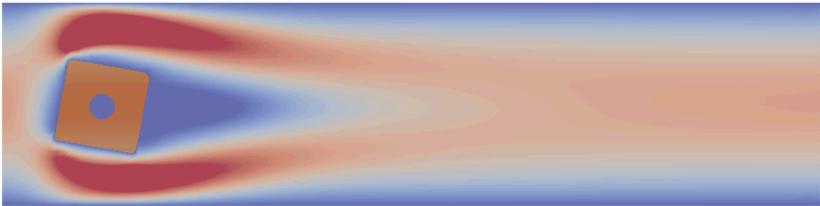
(b)



(c)



(d)



(e)

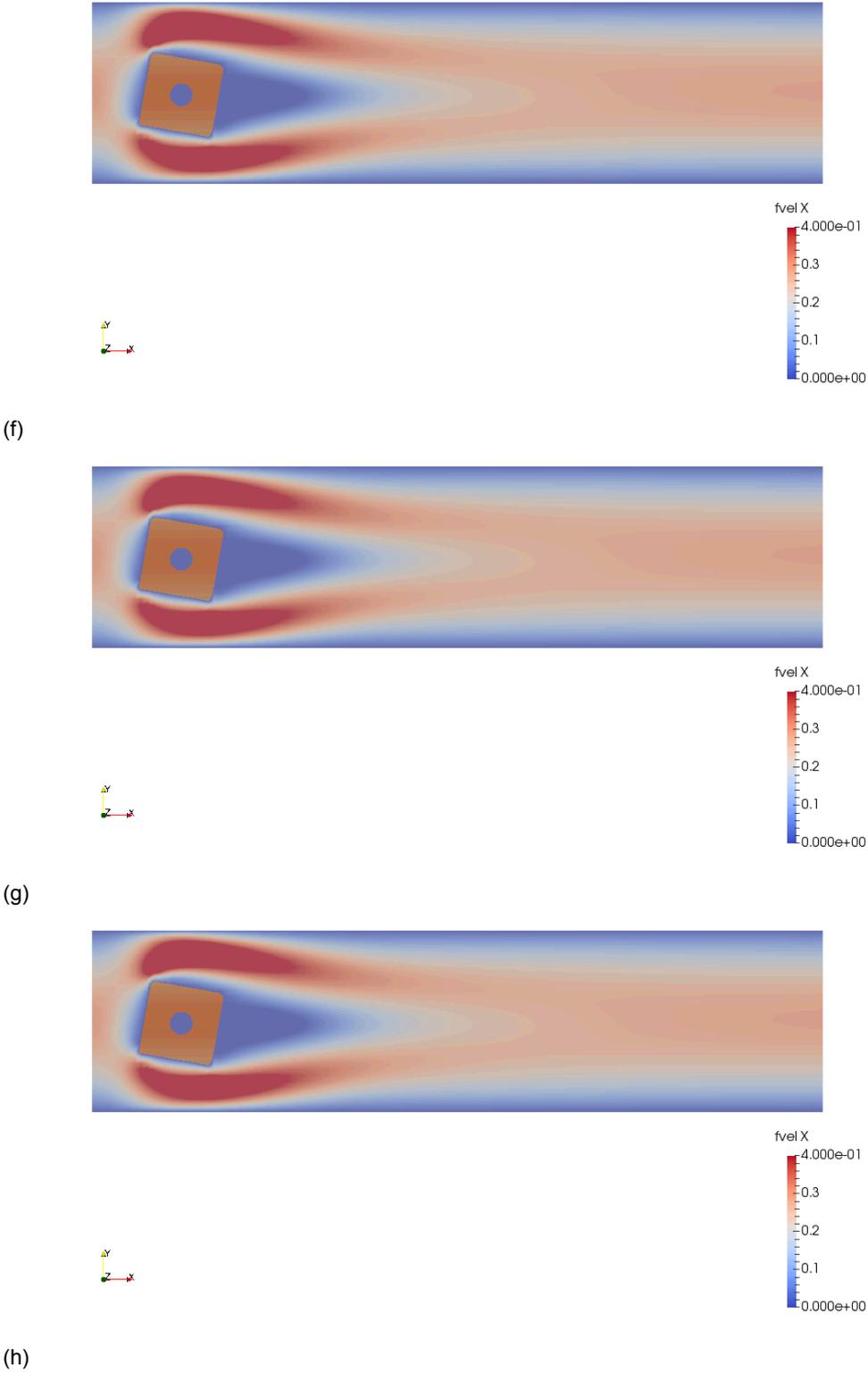
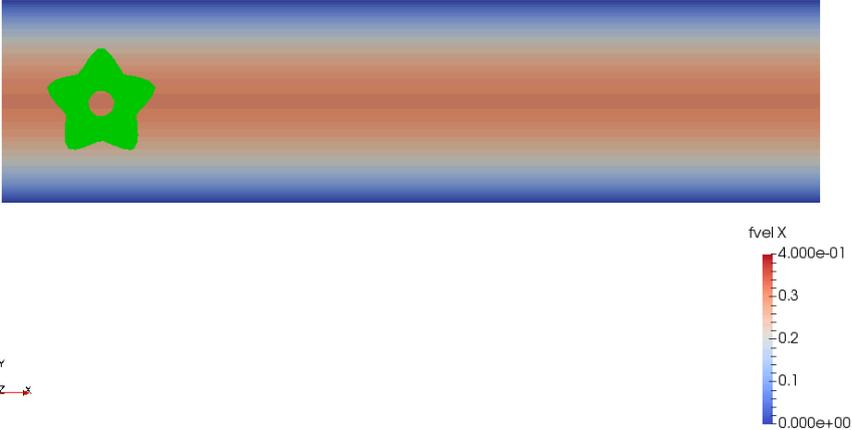
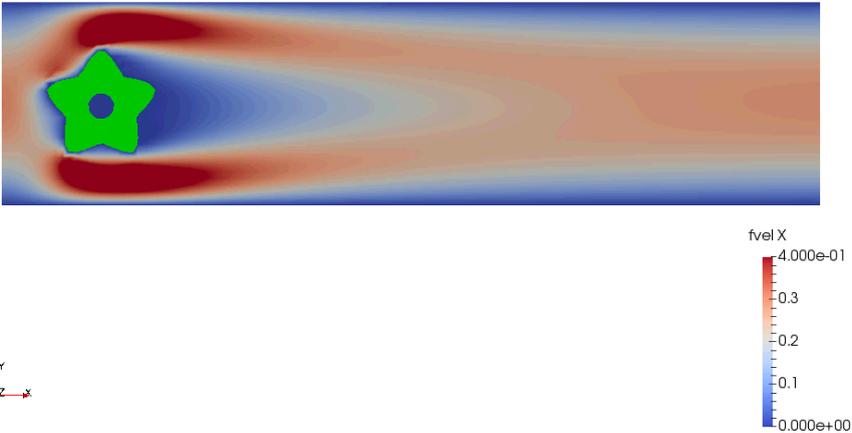


Figure B.1: Evolution of the horizontal velocity distribution and Level Set representation of the section 1 mesh (Clamped)

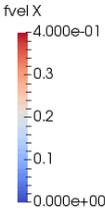
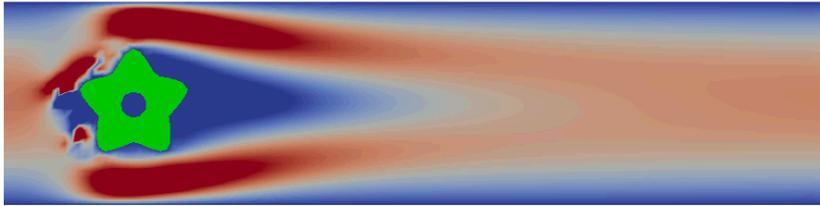
B.2. Mid section: Star shape



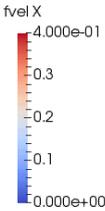
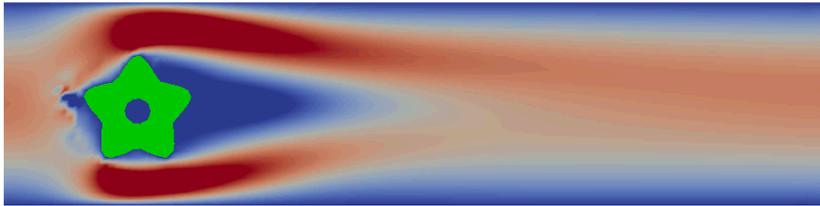
(a)



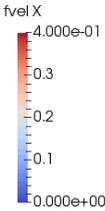
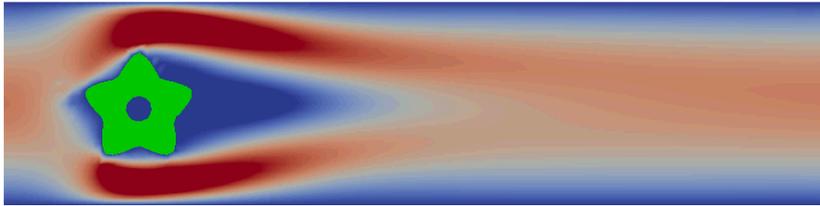
(b)



(c)



(d)



(e)

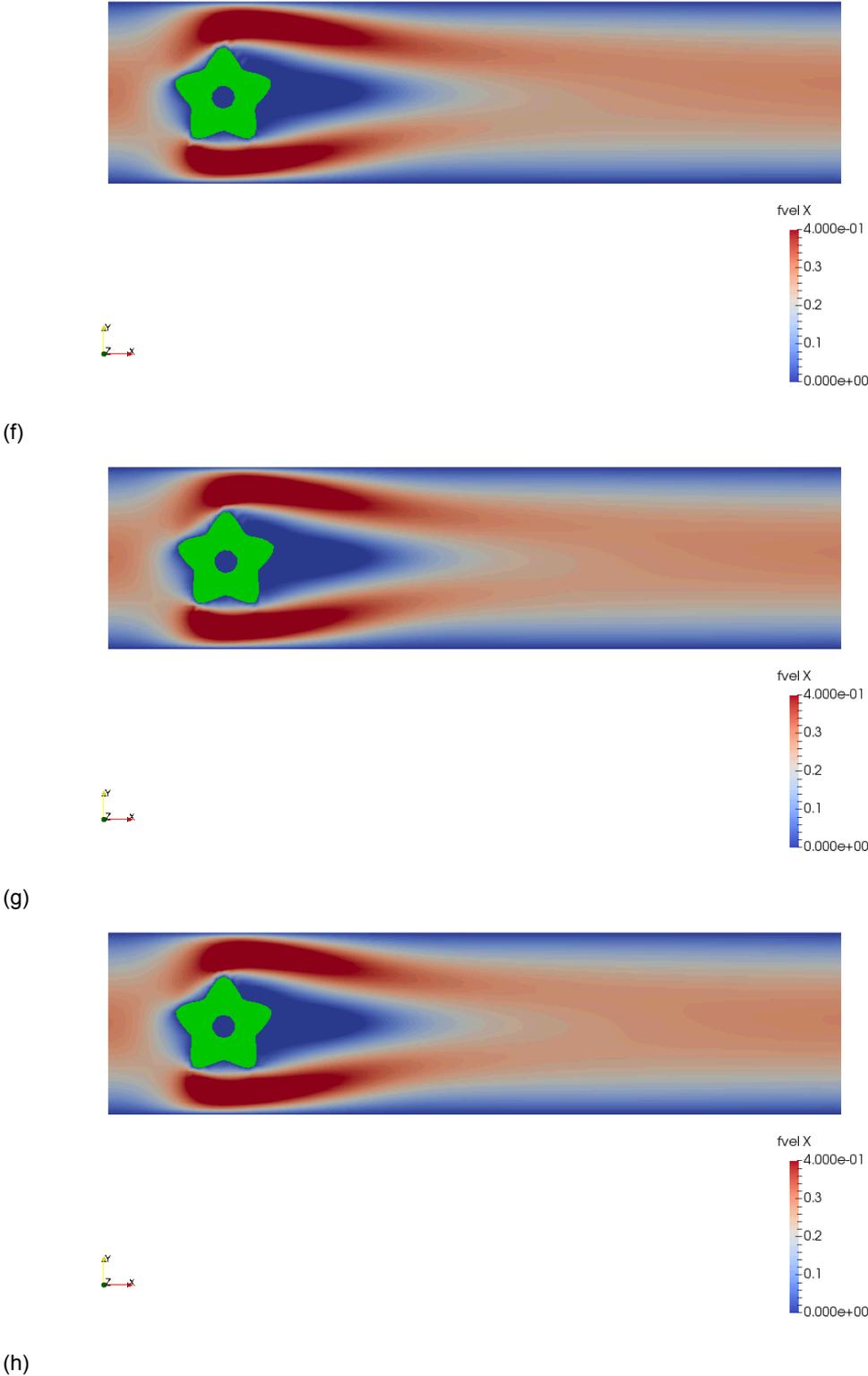
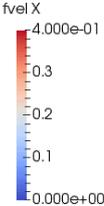
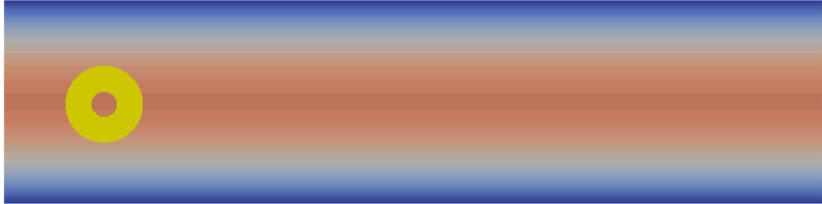
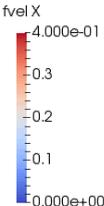
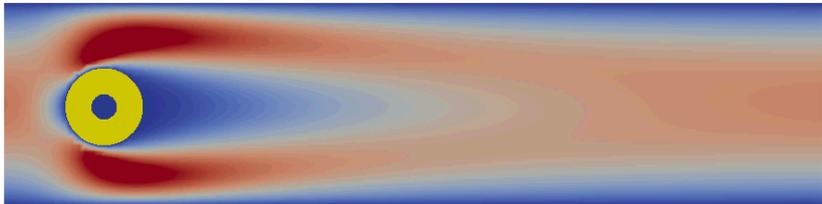


Figure B.2: Evolution through the Newton iterations of the horizontal velocity distribution and Level Set representation of the section 2 mesh (Mid)

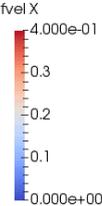
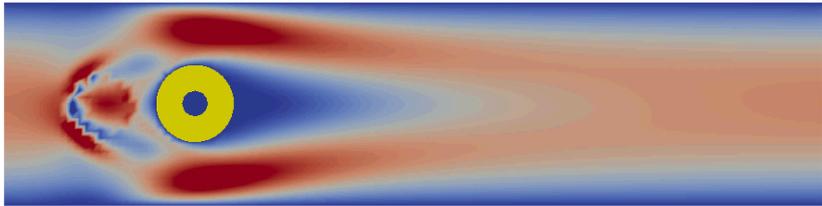
B.3. Tip section: Circular shape



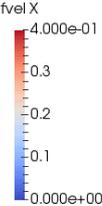
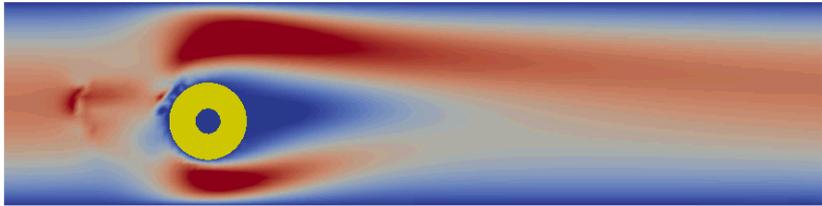
(a)



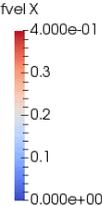
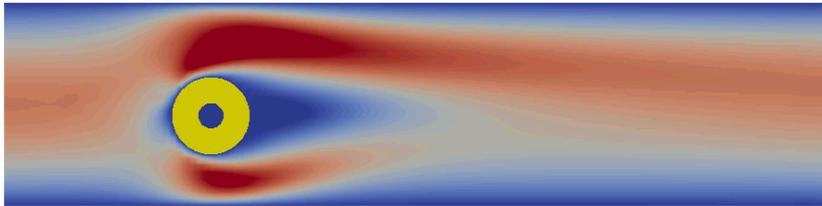
(b)



(c)



(d)



(e)

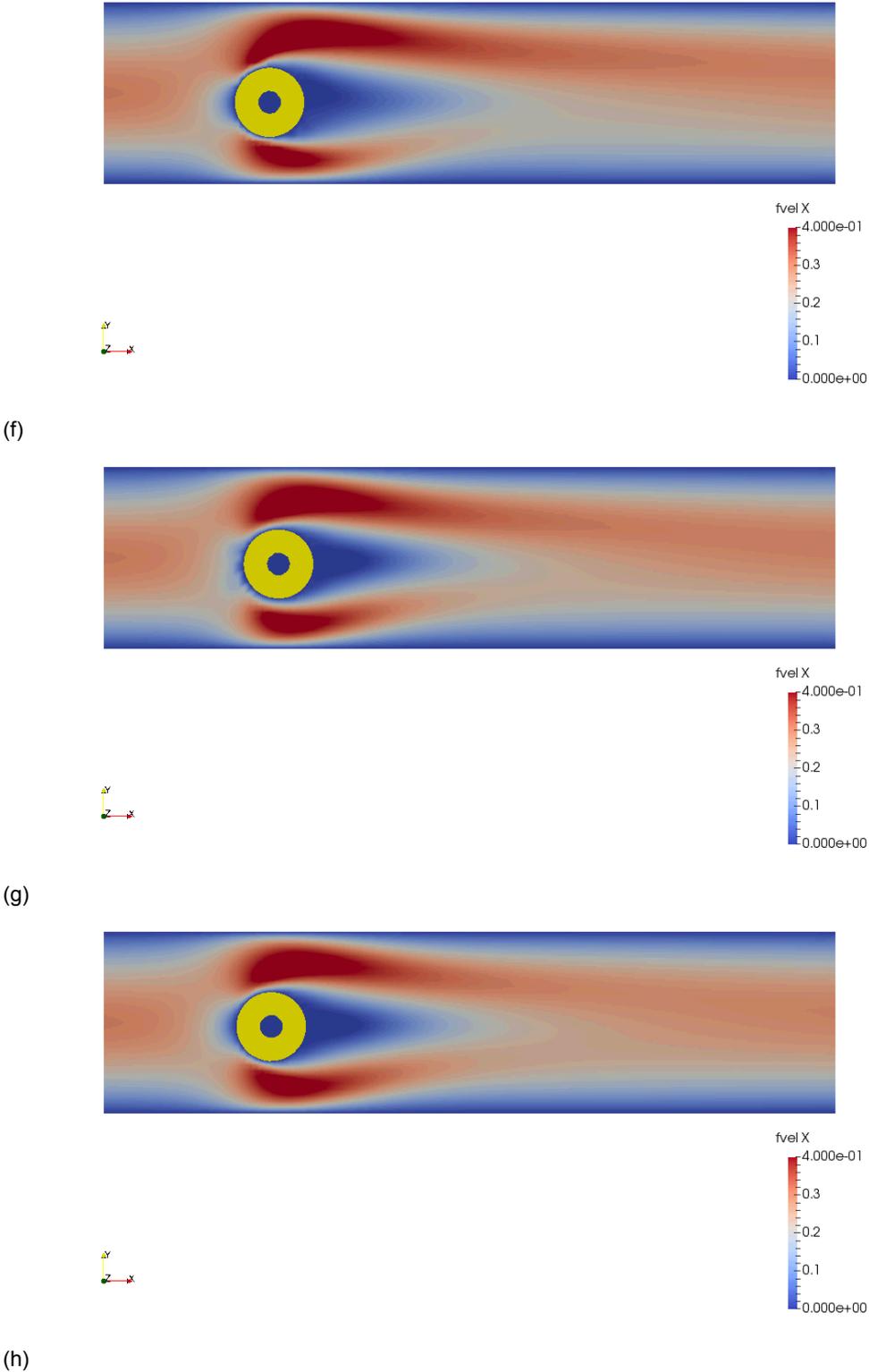
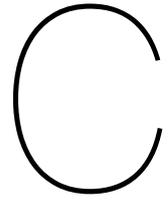
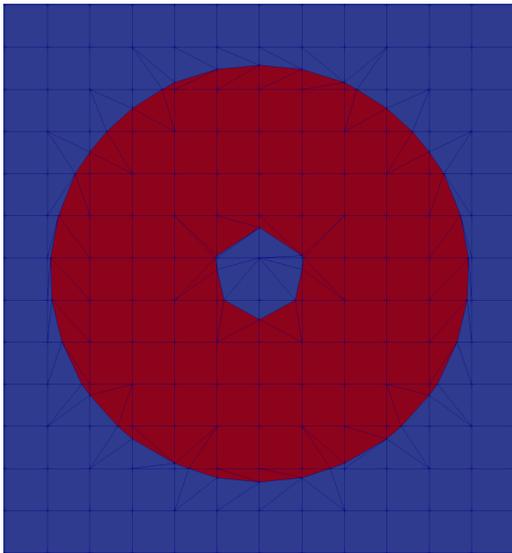


Figure B.3: Evolution through the Newton iterations of the horizontal velocity distribution and Level Set representation of the section 3 mesh (Tip)

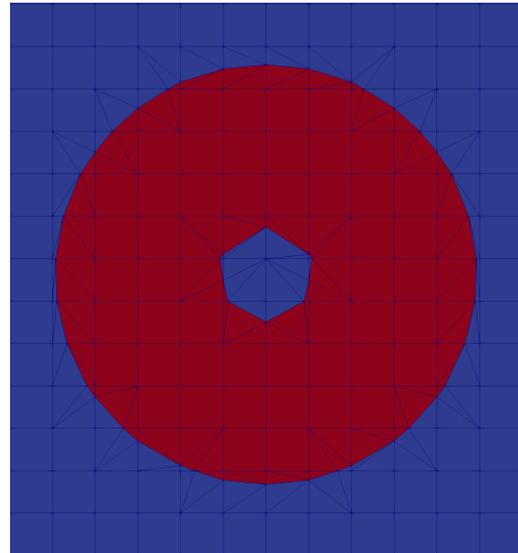


Optimization of hollowed cylinder beam. Visual results

The following images (Figures C.1a to C.1g) show the evolution of the cross-section geometry through the optimization iterations of the example of Section 5.3.1.



(a)



(b)

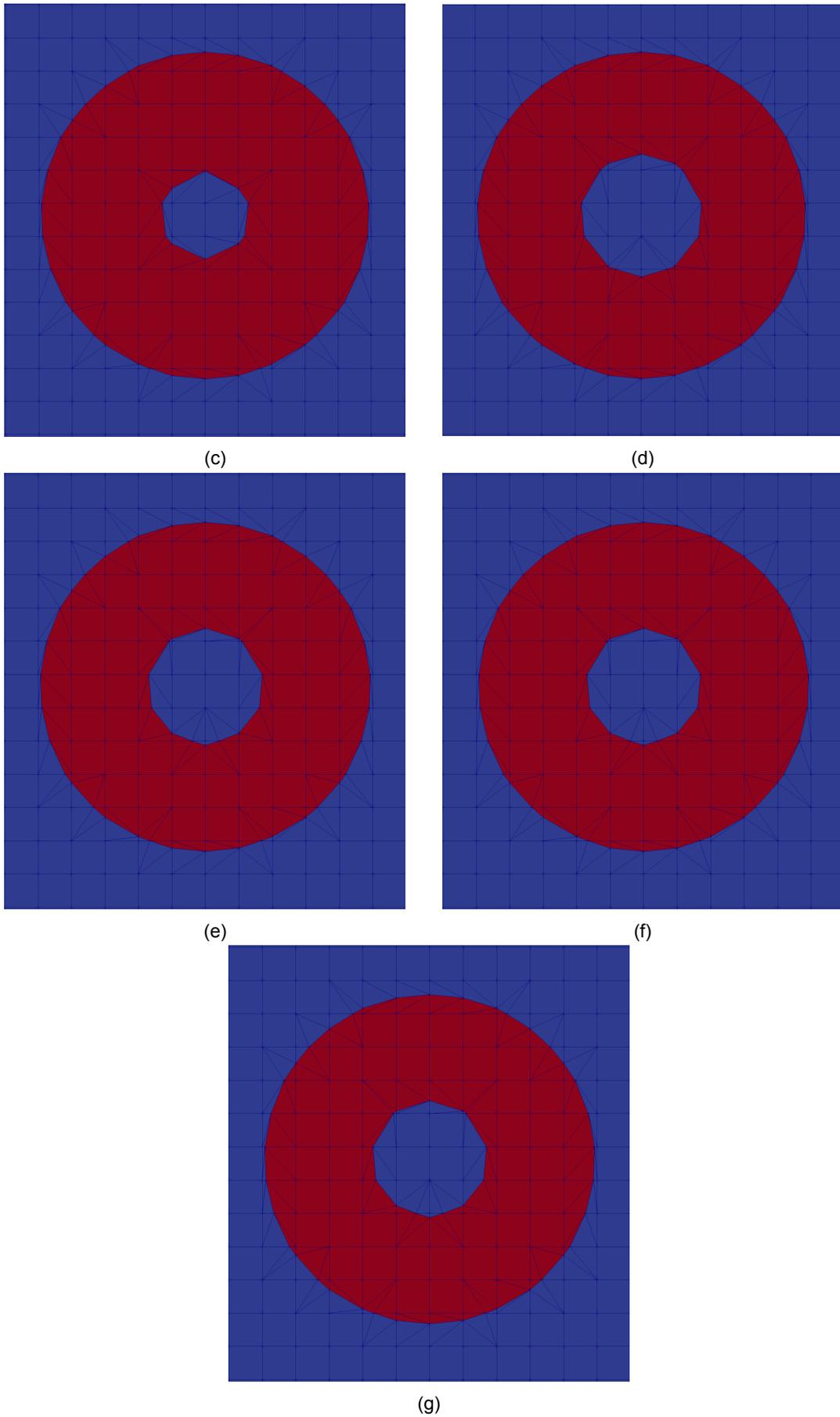


Figure C.1: Evolution of the cross-section geometry through the optimization iterations

Bibliography

- [1] Tu Delft Wind Energy Institute R & D Programme. 2015.
- [2] Grégoire Allaire, François Jouve, and AM Anca Maria Toader. Structural optimization using sensitivity analysis and a level-set method. *Journal of Computational Physics*, 2004.
- [3] K. Bassett, R. Carriveau, and D. S K Ting. 3D printed wind turbines part 1: Design considerations and rapid manufacture potential. *Sustainable Energy Technologies and Assessments*, 11:186–193, 2015.
- [4] Jean-Louis Batoz and Gouri Dhatt. *Modélisation des structures par éléments finis. Volume 2, Poutres et plaques*. Hermès, 1990.
- [5] T. Belytschko and T. Black. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering*, 45(5):601–620, 6 1999.
- [6] T Belytschko, SP Xiao, and C Parimi. Topology optimization with implicit functions and regularization. *Methods in Engineering*, 2003.
- [7] Martin P. Bendsøe and Ole. Sigmund. *Topology Optimization : Theory, Methods, and Applications*. Springer Berlin Heidelberg, 2004.
- [8] José Pedro Blasques and Mathias Stolpe. Maximum stiffness and minimum weight optimization of laminated composite beams using continuous fiber angles. *Structural and Multidisciplinary Optimization*, 2011.
- [9] José Pedro Blasques, Amaral Albergaria, Mathias ; Stolpe, Christian ; Berggreen, and Kim Branner. *Optimal Design of Laminated Composite Beams*. PhD thesis, 2011.
- [10] JP Blasques. User’s Manual for BECAS. *Technical University of Denmark*, 2012.
- [11] Thijs Bosma. *Levelset based fluid-structure interaction modeling with the eXtended Finite Element Method*. PhD thesis, TU Delft.
- [12] W. D. Brouwer, E. C F C Van Herpt, and M. Labordus. Vacuum injection moulding for large structural applications. *Composites Part A: Applied Science and Manufacturing*, 34 (6 SPEC.):551–558, 2003.
- [13] Erik Burman, Miguel Ángel Fernández, and Peter Hansbo. Edge Stabilization for the Incompressible Navier-Stokes Equations: a Continuous Interior Penalty Finite Element Method. (RR-5349):34, 2004.
- [14] Erik Burman, Miguel a Fernández, and Peter Hansbo. Continuous interior penalty finite element method for Oseen’s equations. *Siam J. Numer. Anal*, 44(3):1248–1274, 2006.
- [15] J. Chessa and T. Belytschko. An Extended Finite Element Method for Two-Phase Fluids. *Journal of Applied Mechanics*, 70(1):10, 2003.
- [16] Peter W. Christensen and A. Klarbring. *An introduction to structural optimization*. 2008.
- [17] Christophe Daux, Nicolas Moës, John Dolbow, Natarajan Sukumar, and Ted Belytschko. Arbitrary branched and intersecting cracks with the extended finite element method. *Int. J. Numer. Meth. Eng*, 2000.
- [18] Carlos Felippa. Implementation of One-Dimensional Elements. In *Introduction to Finite Element Methods*, chapter 20. 2016.

- [19] TP Fries and T Belytschko. The extended/generalized finite element method: an overview of the method and its applications. *International Journal for Numerical*, 2010.
- [20] Peter Hansbo, Joakim Hermansson, and Thomas Svedberg. Nitsche’s method combined with space–time finite elements for ALE fluid–structure interaction problems.
- [21] Nicholas Jenkins, Kurt Maute, N Jenkins, and K Maute. Level set topology optimization of stationary fluid-structure interaction problems. *Struct Multidisc Optim*, 52:179–195, 2015.
- [22] Nicholas Jenkins, Kurt Maute, and Kurt Maute KurtMaute. An immersed boundary approach for shape and topology optimization of stationary fluid-structure interaction problems. *Structural and Multidisciplinary Optimization*, 54:1191–1208, 2016.
- [23] Nicholas J Jenkins. Immersed Boundary Methods for Optimization of Strongly Coupled Fluid-Structure Systems. 2017.
- [24] Volker John. Higher order finite element methods and multigrid solvers in a benchmark problem for the 3D Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 2002.
- [25] Yoon Young Kim and Tae Soo Kim. Topology optimization of beam cross sections. *International Journal of Solids and Structures*, 37(3):477–493, 1 2000.
- [26] S Kreissl, G Pingen, and K Maute. An explicit level set approach for generalized shape optimization of fluids with the lattice Boltzmann method. *International Journal for*, 2011.
- [27] Christapher Lang, David Makhija, Alireza Doostan, and Kurt Maute. A simple and efficient preconditioning scheme for heaviside enriched XFEM. *Computational Mechanics*, 54(5):1357–1374, 11 2014.
- [28] Shutian Liu, Xiaomin An, and Haipeng Jia. Topology optimization of beam cross-section considering warping deformation. *Structural and Multidisciplinary Optimization*, 35(5): 403–411, 5 2008.
- [29] Erik Lund. Buckling topology optimization of laminated multi-material composite shell structures. *Composite Structures*, 91(2):158–167, 2009.
- [30] UM Mayer, A Popp, A Gerstenberger, and WA Wall. 3D fluid–structure-contact interaction based on a combined XFEM FSI and dual mortar contact approach. *Computational Mechanics*, 2010.
- [31] JM Melenk and I Babuška. The partition of unity finite element method: basic theory and applications. *Computer methods in applied mechanics and*, 1996.
- [32] Nicolas Moës, John Dolbow, and Ted Belytschko. A finite element method for crack growth without remeshing. *International journal for numerical*, 1999.
- [33] G Nabh. On high order methods for the stationary incompressible Navier-Stokes equations. 1998.
- [34] Toshiki Nagai. Living Document (2017 Spring) Modeling and Optimization of Fluid-Structure Interaction Problems based on Full-Eulerian Extended Finite Element Method. 2017.
- [35] Fuda Ning, Weilong Cong, Jingjing Qiu, Junhua Wei, and Shiren Wang. Additive manufacturing of carbon fiber reinforced thermoplastic composites using fused deposition modeling. *Composites Part B: Engineering*, 80:369–378, 2015.
- [36] Eugenio Oñate. *Structural Analysis with the Finite Element Method Linear Statics*. 2013.
- [37] M Schäfer, S Turek, F Durst, E Krause, and R Rannacher. Benchmark Computations of Laminar Flow Around a Cylinder.

- [38] James Albert. Sethian. *Level set methods and fast marching methods : evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*. Cambridge University Press, 1999.
- [39] Ashesh Sharma, Hernan Villanueva, and Kurt Maute. On shape sensitivities with heaviside-enriched XFEM. *Structural and Multidisciplinary Optimization*, 55(2):385–408, 2017.
- [40] Ole Sigmund and Kurt Maute. Topology optimization approaches: A comparative review. *Structural and Multidisciplinary Optimization*, 48(6):1031–1055, 2013.
- [41] N Sukumar, DL Chopp, N Moës, and T Belytschko. Modeling holes and inclusions by level sets in the extended finite-element method. *Computer methods in*, 2001.
- [42] Krister Svanberg. The method of moving asymptotes??a new method for structural optimization, 1987.
- [43] T E Tezduyar. Stabilized Finite Element Formulations for Incompressible Flow Computations.
- [44] T E Tezduyar, S Mittal, S E Ray, and R Shih. Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements*. *Computer Methods in Applied Mechanics and Engineering*, 95:221–242, 1992.
- [45] N. P. Van Dijk, K. Maute, M. Langelaar, and F. Van Keulen. Level-set methods for structural topology optimization: A review. *Structural and Multidisciplinary Optimization*, 48(3):437–472, 2013.
- [46] Paul S. Veers, Thomas D. Ashwill, Herbert J. Sutherland, Daniel L. Laird, Donald W. Lobitz, Dayton A. Griffin, John F. Mandell, Walter D. M??sial, Kevin Jackson, Michael Zuteck, Antonio Miravete, Stephen W. Tsai, and James L. Richmond. Trends in the design, manufacture and evaluation of wind turbine blades. *Wind Energy*, 6(3):245–259, 2003.
- [47] Carlos H. Villanueva and Kurt Maute. Density and level set-XFEM schemes for topology optimization of 3-D structures. *Computational Mechanics*, 2014.
- [48] Hernan Villanueva and Kurt Maute. CutFEM topology optimization of 3D laminar incompressible flow problems. pages 1–35, 2017.
- [49] MY Wang, X Wang, and D Guo. A level set method for structural topology optimization. *Computer methods in applied mechanics and*, 2003.
- [50] P Wei, MY Wang, and X Xing. A study on X-FEM in continuum structural optimization using a level set model. *Computer-Aided Design*, 2010.
- [51] Gil Ho Yoon. Topology optimization for stationary fluid-structure interaction problems using a new monolithic formulation. *International Journal for Numerical Methods in Engineering*, 82(5):591–616, 4 2010.
- [52] OC Zienkiewicz and RL Taylor. *The finite element method for solid and structural mechanics*. 2005.

Glossary

List of Acronyms

FSI	Fluid-Structure Interaction
FEM	Finite Element Method
XFEM	eXtended Finite Element Method
LS	Level Set
LSM	Level Set Method
LSF	Level Set Field/ Level Set Function
INS	Incompressible Navier-Stokes
SUPG	Streamline-Upwind / Petrov Galerkin
PSPG	Pressure-Stabilized / Petrov Galerkin
DOFs	Degrees of Freedom
FD	Finite Differences