

**A Coupled Adjoint Method
for Optimal Design
in Fluid-Structure Interaction
problems with Large Displacements**

by
Rubén Sánchez Fernández

**Department of Aeronautics
December 2017**

A Coupled Adjoint Method for Optimal Design in Fluid-Structure Interaction Problems with Large Displacements

by

Rubén Sánchez Fernández

submitted in part fulfilment of the requirements for the degree of Doctor of Philosophy
in Aeronautical Engineering and the Diploma of Imperial College London,
December 2017

Student number:	01012290
Project duration:	July 2014 – September 2017
Thesis supervisor:	Dr. R. Palacios, Imperial College London
Thesis committee:	Prof. V. Toropov, Queen Mary University of London Dr. P. Vincent, Imperial College London
Thesis submission:	Monday, 18 September, 2017
Viva-voce examination:	Monday, 27 November, 2017
Submitted in revised form:	Friday, 8 December, 2017

Department of Aeronautics, Imperial College London
SW7 2AZ, London, United Kingdom

O verdadeiro heroísmo consiste em trocar os anseios em realidades, as ideias em feitos.

Alfonso Daniel Rodríguez Castelao

Declaration

I hereby declare that I am the author of this thesis, and that it is the product of my own work. This dissertation has not been submitted for consideration towards any other degree or award. Where appropriate, I have fully acknowledged the ideas and results from the work of other people.

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Rubén Sánchez Fernández

Acknowledgements

First, and above all, I would like to express my sincerest gratitude to my academic supervisor, Dr. Rafael Palacios. Thank you, Rafa, not only for your outstanding technical guidance and vision, but also for your trust, your support, your commitment and your counsel. If there is a reason why I have made it this far, it is definitely because I have had an incredible mentor.

I would also like to thank the examiners of this thesis, Prof Vassili Toropov of Queen Mary University of London and Dr Peter Vincent of Imperial College London, for their time and interest in reading and discussing this work.

The work presented in this thesis has been funded by the Department of Aeronautics at Imperial College London, the UK Engineering and Physical Sciences Research Council Grant EP/J002070/1, and the European Office of Aerospace Research & Development of the US Air Force Office of Scientific Research. Their support is gratefully acknowledged.

I would like to thank the SU2 community, for the excellent framework in which I have been able to develop this thesis. Thank you to Professor Juan Alonso at Stanford, and to Professor Nico Gauger at TU Kaiserslautern, for the opportunities they have granted me to do collaborative research, and also to Francisco Palacios and Thomas Economon for their vision in helping putting this community together. Many thanks to my SU2 friends, Tom, Tim, Salvo, Beckett, for the invaluable discussions and the nice times together.

Finally, many thanks to all my London friends (you know who you are), for helping me discover and enjoy this city during these three and a half years. Muchas gracias a mis amigos en España, por vuestro apoyo, vuestra acogida cada vez que volvía a casa de visita, y vuestra preocupación por mi. A Rosa, porque gracias a tu increíble personalidad, a tu apoyo, a tus cuidados y a tu cariño, he podido sobrevivir a la tesis, y me iré de Londres con mucho más de lo que me esperaba cuando llegué.

Non quería rematar sen adicar o resultado deste traballo a toda a miña familia, a miña avoa e avós, tías e tíos, curmáns e curmás, e a Alberto en particular para darlle moita forza. A meus pais, Alonso e Eugenia, grazas por ensinarme a non rendirme nunca, e por dar todo o que tedes e un pouco máis para que puidese chegar a onde estou. A miña irmá, Rosalía, por escoitarme, por apoiarme, e por facer que me sinta cada día máis orgulloso de ti. Os meus éxitos son vosos.

Abstract

Computational Fluid-Structure Interaction (FSI) methods have reached a significant level of maturity which has led to their incorporation into the analysis stage of industrial applications. However, optimising the structural and/or the aerodynamic performance in highly non-linear coupled FSI problems remains a challenging task, due to the high computational cost of evaluating the functions in this problem and their gradients. Adjoint methods have shown to be an efficient methodology for this latter task, as they can compute sensitivities with a computational cost independent of the number of design variables. On the other hand, their implementation is complex, particularly when the linearisation of the system equations is convoluted.

In this thesis, we develop a novel technique for the evaluation of the coupled adjoint problem. It is based on the consistent application of Algorithmic Differentiation to the fixed-point iterators of the subproblems. This approach makes the computation of the adjoint independent from the solution methods employed for the primal problem, and overcomes the usual limitation for most realistic applications, which is the need for an explicit construction of the analytic Jacobian of the coupled problem. The method poses no restrictions to the non-linearity of the physics in either the fluid or structural field, and it is amenable to partitioned solution methods for the adjoint equations.

We have implemented this method in the open-source SU2 suite, by incorporating structural analysis and FSI capabilities and combining them with the available AD tool for the solution of the coupled adjoint. Finally, the validity of the implementations is assessed through a number of studies. We test the structural and FSI solvers using commercial codes and benchmark cases. Then, we demonstrate the accuracy of the sensitivities obtained using the proposed method. The advantages of our adjoint method for gradient-based optimisation are finally shown for a range of cases, from structural-only to fully coupled optimal design problems.

To those who know me well.

Contents

Declaration	i
Acknowledgements	iii
Abstract	v
1 Introduction	1
1.1 Motivation	3
1.2 Problem overview	4
1.2.1 Fluid-Structure Interaction (primal) problem	4
1.2.2 Adjoint-based gradient calculation for coupled FSI problems	7
1.3 Research Approach	13
1.4 Thesis outline	15
1.5 Publications	17
2 High-fidelity Fluid-Structure Interaction Model	19
2.1 Fluid Dynamics	20
2.1.1 Solution Strategy	20
2.1.2 Solver description	23
2.2 Structural Mechanics	24
2.2.1 Nonlinear equilibrium equations	25
2.2.2 Constitutive equations: Hyperelastic materials	26
2.2.3 Linearisation	28
2.2.4 Discretisation	29
2.2.5 Time integration	31
2.2.6 Solver implementation	32

2.3	Coupled Fluid-Structure Interaction Problem	34
2.3.1	Coupling Strategy	35
2.3.2	Solver Implementation	41
3	Gradient-based Optimisation Using Coupled Adjoints and AD	44
3.1	Conventional Methods to Compute Gradients	45
3.1.1	Finite differences	45
3.1.2	Complex step method	46
3.1.3	Direct method	47
3.1.4	Adjoint method	48
3.2	Gradient Computation using Algorithmic Differentiation	49
3.2.1	Forward mode	50
3.2.2	Reverse mode	51
3.3	The Adjoint Fluid Solver in SU2	53
3.3.1	Adjoint Methods Based on Fixed-Point Iterators and AD	56
3.3.2	AD-based Adjoint for Time-Domain Simulations	58
3.3.3	Code Structure of the AD-Based Adjoint Method in SU2	61
3.4	Adjoint Methods in Structural Mechanics	65
3.4.1	Structural Adjoint Based on Fixed-Point Iterators and AD	67
3.4.2	Extension of the AD-based Adjoint to Structural Dynamics	68
3.4.3	Solver Implementation	70
3.5	Coupled Adjoint Methods	71
3.5.1	A Novel Coupled Adjoint Method for FSI	74
3.5.2	Implementation of the Coupled Adjoint Method using AD	76
3.5.3	Extension of the Coupled Adjoint Method to time-domain FSI	81
4	Optimal Electromechanical Actuation using Dielectric Elastomers	84
4.1	Dielectric Elastomer Modelling	85
4.2	DE Actuation Sensitivities using Conventional Adjoints	88
4.3	DE Sensitivities using AD-based Adjoints	90
5	Coupled Analysis of FSI Problems With Large Deformations	92
5.1	Finite Element Solver Validation and Verification	92
5.1.1	Linear Solver	93

5.1.2	Non-linear Solver for Large Displacements	95
5.2	Coupled FSI Solver Validation and Verification	97
5.2.1	Membrane-Wing in Reattached Flow	97
5.2.2	Flexible Cantilever in Vortical Flow	100
6	Verification of the Sensitivities Computed with the AD-Based Adjoint Method	107
6.1	Flexible Structures In Vacuum	108
6.1.1	Static FEA	108
6.1.2	Actuated FEA	112
6.1.3	Dynamic FEA	116
6.2	Coupled FSI Problems with Large Deformations	121
6.2.1	Vertical Wall in Viscous Flow	121
6.2.2	Electro-Mechanically-Actuated Membrane Wing	126
7	Optimal Design and Actuation of Structures with the ADDA Method	128
7.1	Software architecture	129
7.2	Optimal Parameter Identification	130
7.3	Optimal Electromechanical Actuation	133
7.4	Optimal Mass Distribution in Dynamic Conditions	137
8	Optimal Design and Actuation of Structures under Fluid Loads	142
8.1	Reference configuration	142
8.2	Optimal Design of Vertical Wall in Viscous Flow	144
8.2.1	Wide channel (D16)	146
8.2.2	Narrow channel (D2)	149
8.3	Optimal Actuation of Vertical Wall in Viscous Flow	153
9	Conclusions	157
9.1	Summary of Thesis Main Achievements	157
9.2	Main Research Outcomes	159
9.3	Recommendations for Future Work	162
	Bibliography	164

List of Tables

3.1	Computational evaluation of the function $y = J(\alpha_1)$	49
3.2	Forward mode of AD.	50
3.3	Reverse mode of AD with one input and one output.	51
3.4	Number of evaluations per iteration of a problem with $n_\alpha = 200$ in an optimization framework. *The complex step method requires operator overloading, which increases the cost per evaluation. **The direct method does not evaluate the primal solver, but a linear system with the same number of variables.	55
5.1	Mesh discretisations	95
5.2	Problem definition	101
5.3	Summary of results and convergence study	105
6.1	Performance study of the ADDA solver. Memory footprint, computational time and convergence.	124
7.1	Value of the objective function for different initial guesses and number of actuators.	137
8.1	Geometrical description.	143
8.2	Structural properties.	143
8.3	Values of the objective function J for α_0	145
8.4	Results for the D16E20 case.	148
8.5	Results for the D16E40 case.	148
8.6	Results for the D2E20 case.	150
8.7	Results for the D2E40 case.	152
8.8	Results for the actuated case. (** indicates that the simulation started from the initial guess $\alpha_i = 5.0 \forall i$ in α_0)	155

List of Figures

1.1	Fluid-Structure Interaction problem.	5
1.2	Monolithic Fluid-Structure Interaction solver.	6
1.3	Partitioned Fluid-Structure Interaction solver.	6
2.1	Dual grid discretization	21
2.2	Class hierarchy for solving a Navier-Stokes problems.	23
2.3	Reference and current configuration.	25
2.4	Class hierarchy for solving a two-dimensional, non-linear structural problem. . .	32
2.5	List of child classes for the CElement class.	33
2.6	Child class structure for the definition of geometrical non-linearities.	34
2.7	Child class structure for the definition of material non-linearities.	34
2.8	Introduction of the CIteration and CDriver classes.	41
2.9	Sample FSI problem run in multiple cores.	42
2.10	Transfer structure for Fluid-Structure Interaction problems.	43
3.1	Computational graph for the expression $y = v_0 \cos(v_{-1})$	52
3.2	Propagation of gradients in reverse mode for the expression $y = v_0 \cos(v_{-1})$. . .	53
3.3	Sample airfoil NACA 2412 discretised with 100 points in the surface.	54
3.4	Fix-point adjoint iteration.	61
3.5	Primal (a) and reverse (b) computational paths for the solver $\mathbf{F}$	62
3.6	Primal (a) and reverse (b) computational paths for the solver $\mathbf{F}_{\mathbf{z}}$, where the subindex indicated the path is derived with respect to the mesh variables $\mathbf{z}$. . .	63
3.7	Primal (a) and reverse (b) computational paths for the calculation of the gradient. .	63
3.8	Primal (a) and reverse (b) computational paths for the time-domain solver $\mathbf{F}_n$, where the subindex indicated the path is derived with respect to the mesh vari- ables $\mathbf{z}$	64

3.9	Primal (a) and reverse (b) computational paths for the solver S	71
3.10	Primal (a) and reverse (b) computational paths for the solver S	72
3.11	Primal and reverse computational paths for the FSI solver, using a structural-based objective function.	79
3.12	Primal and reverse computational paths for the FSI solver, using a fluid-based objective function.	80
4.1	Dielectric Elastomer transducer.	85
4.2	Primal (a) and reverse (b) computational paths for the solver S with electric actuation.	91
5.1	Beam validation case	93
5.2	Nodal displacement magnitude. Results from ABAQUS (a) and SU2 (b), for 13311 elements. Displacements scaled 3000 times.	94
5.3	Nodal displacement magnitude, convergence study. Relative error in the maximum displacements between SU2 and ABAQUS.	94
5.4	Convergence study definition for non-linear structural problems.	96
5.5	Nodal displacement magnitude, convergence study for non-linear problems. Relative error in the maximum displacements between SU2 and ABAQUS. The <i>continuous</i> lines correspond to the dead load, while the <i>dashed</i> lines correspond to the uniform, follower pressure.	96
5.6	Membrane wing test case (not to scale).	98
5.7	Midline surface deflection of the membrane for an angle of attack of 4°	99
5.8	Velocity fields and membrane deformation.	99
5.9	Fluid-Structure Interaction test case (not to scale).	100
5.10	Fluid discretisations: <i>coarse</i> (top) and <i>fine</i> (bottom) meshes for the FSI benchmark	101
5.11	Time histories of the vertical displacements at the tip of the cantilever and FFT plots, for the a) <i>coarse</i> and b) <i>fine</i> fluid discretisations, and the two time steps considered, $\Delta t = 0.0075$ s and $\Delta t = 0.005$ s.	102
5.12	Pressure contours and structural displacements for a cycle of vibration for the <i>coarse</i> mesh (a) and for the <i>fine</i> mesh (b). $T=0$ (1), $T=\pi/2$ (2), $T=\pi$ (3), $T=3\pi/2$ (4). The arrow in (b) points out some vortical structures that are damped out by the coarser discretisation.	104

5.13	Time histories of the vertical tip displacements for different values of the relaxation parameter for the BGS FSI solver. The base solution (in grey) is obtained with $\omega_{fixed} = 0.5$, and no predictor.	106
6.1	Test cases for the computation of the static adjoint sensitivity in flexible structures.	109
6.2	Objective function (top left), forward mode sensitivity (bottom left) and relative error with respect to Forward Mode of the AD-based Discrete Adjoint, Central Differences ($2h = 500$) and Conventional Adjoint, for an uniform body load. . .	110
6.3	Objective function (top left), forward mode sensitivity (bottom left) and relative error with respect to Forward Mode of the AD-based Discrete Adjoint, Central Differences ($2h = 500$) and Conventional Adjoint (**indicates it is obtained using and approximate tangent matrix), for a follower, uniform pressure.	111
6.4	Mean deflection of the membrane for a dead weight W , $\rho = 200 \text{ kg/m}^3$, and for a (follower) pressure $p = 0.4 \text{ Pa}$	112
6.5	Mean deflection of the membrane for a follower pressure $p = 0.4 \text{ Pa}$ under different levels of prestretch, ϵ_0	113
6.6	Mean deflection of the membrane in vacuum with a prestretch $\epsilon_0 = 5\%$ under different levels of electromechanical actuation. E_F is the modulus of the electric field, V/t	113
6.7	Objective function and sensitivities for different prestretch ϵ_0 and actuation conditions, E_F . The values of J marked with a diamond correspond to the geometries presented in Fig. 6.6	114
6.8	Relative error of the sensitivities computed using the conventional discrete adjoint with an approximate Jacobian, with respect to the forward mode of AD. Five different magnitudes of prestretch are considered. Load conditions: dead load (continuous line) and follower pressure (dashed line).	114
6.9	Relative error of the sensitivities computed using a Central Differences scheme, with a constant increment $2h = 0.2 \text{ MV/m}$, with respect to the forward mode. Five different magnitudes of prestretch are considered. Load conditions: dead load (continuous line) and follower pressure (dashed line)	115

6.10	Relative error of the sensitivities computed using the ADDA with respect to the Forward mode. Five different magnitudes of prestretch are considered. Load conditions: dead load ($\bullet$) and follower pressure ($+$). **Indicates that an approximate Jacobian is used.	116
6.11	Test case description for the dynamic adjoint.	117
6.12	Time history of the instantaneous (J_i) and the averaged (J) value of the objective function.	118
6.13	Values of the objective function in the region of study.	118
6.14	Sensitivities computed using the forward mode of AD.	118
6.15	Relative error ($\log_{10}$) of the sensitivities computed using the ADDA method with respect to the forward mode.	119
6.16	Relative error ($\log_{10}$) of the sensitivities computed using Central Differences with respect to the forward mode.	119
6.17	Objective function (a) and gradient with respect to the density (b) for a fixed value of $E = 250$ kPa.	120
6.18	Objective function (a) and gradient with respect to the Young's Modulus (b) for a fixed value of $\rho = 100$ kg/m ³	120
6.19	Test case description: vertical wall in viscous flow.	121
6.20	Flow streamlines and cantilever deformation with $E = 13$ and 28 kPa.	122
6.21	Objective function (top left), sensitivity (bottom left) and relative error between ADDA method and central differences with respect to the forward mode of AD.	122
6.22	Convergence of dJ/dE and $dJ/d\nu$ using ADDA.	123
6.23	Relative error for central differences for different values of h , compared with the relative error of the ADDA, with respect to the gradient computed using the forward mode of AD.	124
6.24	Geometrical discretisation of the cantilever into 10 regions, each corresponding to one design variable.	125
6.25	Far left: Design variable distribution. Left: Sensitivities for the design vector. Centre: Relative error of the sensitivities computed using ADDA and central differences (CD), $2h = 500$, with respect to the value computed using the forward mode (FM). Right: Absolute error of the same magnitude.	126

6.26	Objective function (top left), sensitivity (bottom left) and relative error between ADDA method and central differences with respect to the forward mode of AD, for an electromechanically actuated membrane wing.	127
7.1	Python framework available in SU2 for optimal aerodynamic shape design. . . .	129
7.2	New Python framework available in SU2 for structural and multiphysics problems.	130
7.3	Objective distribution of stiffness and deformed geometry for $p = 2$ Pa.	131
7.4	Optimisation evolution of the inverse problem for an uniform initial guess. . . .	131
7.5	Optimisation evolution of the inverse problem for a linear initial guess.	132
7.6	Optimisation evolution of the inverse problem for a parabolic initial guess. . . .	132
7.7	Optimisation history of the inverse problem for the uniform, linear and parabolic initial guesses.	133
7.8	Optimisation evolution of the inverse problem for an improved parabolic initial guess.	133
7.9	Optimisation history of the inverse problem for the improved parabolic initial guess.	134
7.10	Cantilever bimorph and distribution of DE actuators in the bottom half.	134
7.11	Deformed geometry under no actuation, $E = 2.5 \cdot 10^5$ Pa, $\nu = 0.35$, $\rho = 20$ kg/m ³ .	135
7.12	Cantilever bending moment, and actuation moments due to Electric Field using 1 (a) and 8 (b) design variables.	136
7.13	Time history of the optimisation (left), optimised actuation (centre) and final shape of the bimorph (right) for 1, 8 and 64 actuators.	136
7.14	Time history of the optimisation (left), optimised actuation (centre) and final shape of the bimorph (right) for an uniform, linear and parabollic initial guess in the actuation distribution, and 64 actuators.	137
7.15	Time history of the tip displacement (a) and optimised distribution of density (b), for the following values of c_d : (1) $c_d = 0.1\%$ (2) $c_d = 0.5\%$ (3) $c_d = 1\%$	138
7.16	Time history of the vertical tip displacement (left) and optimised distribution of density (right), for the $c_d = 2\%$	139
7.17	Time history of the vertical tip displacement (a) and optimised distribution of density (b), for the following values of c_d : (1) $c_d = 2.5\%$ (2) $c_d = 3.5\%$ (3) $c_d = 4.5\%$	140

7.18	Time history of the vertical tip displacement (a) and optimised distribution of mass (b), for the following values of c_d : (1) $c_d = 5E - 02$ (2) $c_d = 6E - 02$ (3) $c_d = 8E - 02$	140
7.19	Reference and optimised values of J (left), and % reduction in J obtained after the optimisation (right).	141
8.1	Parametric FSI domain.	142
8.2	Streamlines and wall deformation for the four nominal cases.	144
8.3	Design variables (a) and target geometry (b).	145
8.4	Deformed geometries for α_0	145
8.5	Optimisation history (a), initial state (b), and optimised geometries (top) and distribution of design variables (bottom) for the unconstrained (c) and constrained (d) test case D16E20	147
8.6	Optimisation history (a), initial state (b), and optimised geometries (top) and distribution of design variables (bottom) for the unconstrained (c) and constrained (d) test case D16E40	148
8.7	Optimisation history (a), initial state (b), and optimised geometries (top) and distribution of design variables (bottom) for the unconstrained (c) and constrained (d) test case D2E20	149
8.8	Test case D2E20, unconstrained. Deformed geometries (top) and value of the design variables (bottom) for the initial guess, two intermediate optimisation stages ($k=10$ and $k=20$), and the final design.	150
8.9	Test case D2E20, constrained. Deformed geometries (top) and value of the design variables (bottom) for the initial guess, two intermediate optimisation stages ($k=5$ and $k=10$), and the final design.	151
8.10	Optimised geometries (top) and distribution of design variables (bottom) for a constrained D2E20 test case with an initial guess based on the unconstrained result.	152
8.11	Optimisation history (a), initial state (b), and optimised geometries (top) and distribution of design variables (bottom) for the unconstrained (c) and constrained (d) test case D2E40	153

8.12 Optimised geometries (top) and distribution of design variables (bottom) for a constrained D2E40 test case with an initial guess based on the unconstrained result.	153
8.13 Design variables (a) and target geometry (b) with actuation mechanism.	154
8.14 optimisation history for the actuated case.	155
8.15 Actuated test case. Initial geometry (a), optimised geometry (b), and value of the design variables for the optimised geometry (c), for the cases with domains D16 (upper) and D2 (lower).	156

Nomenclature

State variables:

$\mathbf{u}$	Structural displacements
$\mathbf{w}$	Flow conservative variables
$\mathbf{z}$	Fluid mesh nodes locations

Residual equations:

$\mathcal{S}(\cdot)$	Structural equations
$\mathcal{F}(\cdot)$	Flow equations
$\mathcal{M}(\cdot)$	Flow mesh movement equations
$\mathcal{G}(\cdot)$	Generic, non-linear equations

Fixed-point equations:

$\mathbf{S}(\cdot)$	Structural equations
$\mathbf{F}(\cdot)$	Flow equations
$\mathbf{M}(\cdot)$	Flow mesh movement equations

Coupled problem:

Γ	FSI interface
λ	Flow tractions
ω	Aitken's relaxation parameter
$\mathbf{K}$	Tangent (Jacobian) matrix

Optimisation problem:

J	Objective function
α	Design variables

β_k	Step size
$\mathcal{D}_k$	Search direction
k	Optimisation step
$\mathcal{L}$	Lagrangian

Adjoint problem:

$\tilde{\cdot}$	Conventional adjoint variables
$\bar{\mathbf{u}}$	AD-based structural adjoint variables
$\bar{\mathbf{w}}$	AD-based fluid adjoint variables
$\bar{\mathbf{z}}$	AD-based mesh adjoint variables
$\bar{\mathbf{S}}$	Reverse structural fixed-point solver
$\bar{\mathbf{F}}$	Reverse fluid fixed-point solver
$\bar{\mathbf{M}}$	Reverse mesh fixed-point solver
$\mathbf{q}$	Structural extended nodal variable

Fluid problem:

ρ_f	Fluid density
$\mathbf{v}$	Flow velocities
e	Total energy of the flow per unit mass
$\boldsymbol{\tau}$	Viscous stress tensor
μ	Fluid viscosity
C_p	Specific heat
T	Temperature
$\mathbf{F}^c(\cdot)$	Convective flux
$\mathbf{F}^v(\cdot)$	Viscous flux

		D	Constitutive matrix
Mesh problem:		M	Mass matrix
$\tilde{\mathbf{K}}$	Fictitious stiffness matrix	$\mathbf{R}_T$	Time-dependent residual component
$\tilde{\mathbf{f}}$	Fictitious force	ϵ_0	Prestretch
		E	Young's modulus
Structural problem:		ν	Poisson's ratio
$\mathcal{B}_0$	Body in reference configuration		
$\mathcal{B}$	Body in current configuration	Electric problem:	
$\mathbf{R}$	Vector of reference nodal coordinates	$\mathbf{E}_F$	Electric field in the current configuration
$\mathbf{r}$	Vector of current nodal coordinates	$\tilde{\mathbf{E}}_F$	Electric field in the reference configuration
Φ	Transformation function	V	Actuation voltage
$\mathbf{F}_\Phi$	Deformation gradient	t	Dielectric elastomer thickness
$\mathbf{E}$	Green strain tensor		
$\mathbf{e}$	Almansi strain tensor		
$\mathbf{C}$	Right Cauchy-Green deformation tensor		
$\mathbf{b}$	Left Cauchy-Green deformation tensor		
$\boldsymbol{\sigma}$	Cauchy stress tensor		
ρ_s	Structural density		
$\mathbf{s}$	Imbalance force		
δW	Virtual work		
$\mathbf{S}_{PK}$	Second Piola-Kirchhoff tensor		
J_Φ	Determinant of $\mathbf{F}_\Phi$		
Ψ	Strain energy function		
$\mathcal{C}$	Material elasticity fourth-order tensor		
$\mathbf{T}$	Equivalent internal forces		
$\mathbf{F}_b$	Body forces		
$\mathbf{F}_\Gamma$	Surface forces		
$\mathbf{K}_c$	Constitutive term of the tangent matrix		
$\mathbf{K}_\sigma$	Stress term of the tangent matrix		
$\mathbf{K}_{ext}$	External load term of the tangent matrix		
N	Shape function		
$\mathbf{K}$	Matrix of derivatives of the shape functions		

Chapter 1

Introduction

The interaction between flexible structures and fluid flows is a highly challenging problem with many applications in engineering contexts. From the design of largely-deformable aircraft wings to the investigation of the blood flowing through our veins, the study of the coupled behaviour of solids and fluids interacting with each other is a demanding task, due to the complexity of the constitutive equations and the strongly non-linear coupling effects.

Computational Fluid-Structure Interaction (FSI) analysis has reached a significant level of maturity since the earlier works of Belytschko [1, 2], Bathe and Hahn [3] or Donea *et al.* [4] in the late 1970s and early 80s. The advances in the capacity of computers and the development of new numerical techniques have made it feasible to effectively address complex industrial problems by means of a purely computational FSI analysis [5].

As a result, computational FSI is an area of active research that covers a wide range of problems involving complex flow and/or structural behaviour. Applications in aeronautical engineering are numerous. A number of established CFD solvers developed by reputed institutes have been used for computational aeroelasticity, such as NASA's FUN3D [6–8], DLR's TAU [9–11] or ONERA's elsA [12–14]. In most cases, they have been coupled to external structural solvers. The computational study of the coupling effects between flexible wings and surrounding fluid flows is currently a central topic in many pieces of research. Saeedi *et al.* [15] use FSI analysis to investigate the aerodynamic performance of a double membrane sailwing with a rigid leading-edge mast. The role of the flexibility of the wings in natural flyers is studied by Nguyen *et al.* [16] and Tobing *et al.* [17] in the computational modelling and analysis of the flapping flight of fireflies and bumblebees, respectively. Bio-inspired, flexible structures have applications in energy harvesting as explained by Liu *et al.* [18], who run a computational investigation on the

effect of the passive flexibility of fish-like cambered airfoils. Such structures can also be actively controlled via active morphing of a spine-like structure, as done by Woods *et al.* [19] using coupled FSI methods. Active control of bio-inspired wings is also studied via computational means by Buoso and Palacios [20, 21], who investigate the effect of electromechanical actuation in compliant, flexible bat-like membrane wings.

Computational FSI has several other applications in engineering beyond wing-flow interaction analysis. Computational aeroelastic analysis can be extended to other kind of structures, such as bridge decks [22, 23] or solar collectors [24]. These techniques can also be applied to problems undergoing complex mechanical behaviour, such as fatigue-damage prediction in wind turbines [25], the behaviour of flexible structures subject to gust-induced wind loads [26] and blast impacts [27], or even the collision between two ships [28]. Another area in which computational FSI is widely used is in bioengineering. From the simulation of the red blood and tumour cells in thin blood vessels [29] to the simulation of a full respiratory system [30], several works tackle the applicability of FSI methods to medical problems. Other recent examples found in this area range from the study of the flow in cerebral arteries by Tricerri *et al.* [31], to the evaluation of the behaviour of aortic valve prosthesis by Luraghi *et al.* [32] or the characterisation of a cerebral artery with aneurysm by Eken and Sahin [33], just to name a few.

The previous examples demonstrate the vast number of problems that are currently being tackled using computational FSI. The maturity of the field has led to a progressive incorporation of high-fidelity FSI analysis in the analysis stages of several industrial applications. However, some challenges remain before these techniques can be effectively incorporated into the design chain.

Firstly, the large cost per evaluation of the coupled solution methods in FSI problems limits the applicability of gradient-free optimisation methods, which generally require a large number of evaluations of the solver in order to advance and converge the optimisation problem. This fact leads in some cases to the use of surrogate models in coupled FSI optimisation frameworks (see, e.g., [34, 35]). However, the accuracy of these methods depend greatly on the choice of the surrogate, which may be problem dependent [36, 37]. Alternatively, lower fidelity [38] and multifidelity [39] methods may be developed in order to decrease the cost of evaluating the objective function, but these methods are normally developed specifically for a particular problem and may lack generality.

Gradient-based optimisation methods can reduce the computational cost of the optimisation

process based on high-fidelity models by greatly reducing the number of evaluations of the objective function, as shown for aerodynamic design problems by Zingg *et al.* [40]. However, the efficiency of these methods depends largely on the cost and accuracy of the calculation of the gradients. Very accurate gradients may be obtained using techniques such as the complex-step method [41] or direct methods based on the chain rule [42–44]. Even simpler methods such as standard finite-differences can provide a good accuracy for practical applications. However, the cost of any of these methods depends directly on the number of design variables of the problem, which limits their applicability to problems with a small number of variables.

Adjoint methods are able to provide accurate gradients with a cost that is independent of the number of design variables. They were first used in control theory as early as the 1960s [45, 46]. In the context of coupled, complex multiphysics problems, sensitivity analysis was first tackled by Sobieszczanski-Sobieski in 1990 [47], however it is not until more than a decade later that incipient adjoint method applications were shown in the solution of high-fidelity coupled systems. This was initiated by the earlier works of Maute *et al.* [48, 49] and Martins *et al.* [50, 51] in the field of aero-structural steady-state optimisation, and was then followed by several others such as Barcelos *et al.* [52], Fazzolari *et al.* [53], Perego *et al.* [54], Abu-Zurayk and Brezillon [55] or Degroote *et al.* [56].

1.1 Motivation

Recent applications of the adjoint method to the design of full aircraft configurations in steady-state [57, 58], or even to time-dependent optimisation of helicopter rotors [59] demonstrate the applicability of this method to very complex, industrial-scale FSI problems. However, the analytical approach taken in these cases requires to handle very large Jacobian matrices who must be exactly derived and stored for the solution of the adjoint problem. The high development cost involved by this approach and the constant updates that would be required when the code is altered are pointed out by Zingg *et al.* [40] as some of the key disadvantages of gradient-based optimisation methods based in adjoints.

In this thesis, our main goal is to investigate new methodologies to calculate adjoint-based gradients in fully non-linear coupled FSI problems, which are able to overcome this limitation. This research will explore the applicability of state-of-the-art methods based on the application

of Algorithmic Differentiation (AD) techniques to fixed-point iterators in flow solvers, to structural problems undergoing material and geometrical non-linearities, first, and then to coupled FSI problems. A finite-element-based solver has been developed within the open-source CFD platform SU2, and coupled natively with the fluid solver. Then, a novel coupled adjoint method has been proposed and incorporated to the suite. This dissertation will describe the theoretical principles behind these techniques and their implementation. Finally, the applicability of this new method to sensitivity analysis and optimal design of fully coupled, non-linear FSI problems will be demonstrated.

1.2 Problem overview

In this section, a brief summary of the state of the art in adjoint-based optimal design of coupled Fluid-Structure Interaction problems is provided. The literature review at this stage has been kept at the very high level, with detailed reviews of the specifics provided independently in each chapter. First, the computational solution of the *primal* problem, that is, the physical problem involving the interaction of (potentially) non-linear fluid and structural domains, is discussed. Then, an overview of optimisation methods for coupled FSI problems is provided, and the computation of sensitivities for gradient-based optimisation is covered, with a focus on coupled adjoint methods.

1.2.1 Fluid-Structure Interaction (primal) problem

From a physical point of view, the FSI problem involves the analysis of two different fields (fluid and structure) governed by their own constitutive equations ($\mathcal{F}$ and $\mathcal{S}$), and interacting with each other over a common interface (Γ). We define the fluid and structural state variables, respectively, as $\mathbf{w}$ and $\mathbf{u}$. In order for the problem to be fully coupled, the tractions from the fluid field, λ_Γ , are applied onto the structure at the common interface. This results in a deformation $\mathbf{u}$ of the structural domain, which modifies its external shape and affects the fluid field. Consequently, both the structural and fluid constitutive equations need to be expressed as a function not only of their own state variables, but also of the state of the other field. This is shown in Fig. 1.1.

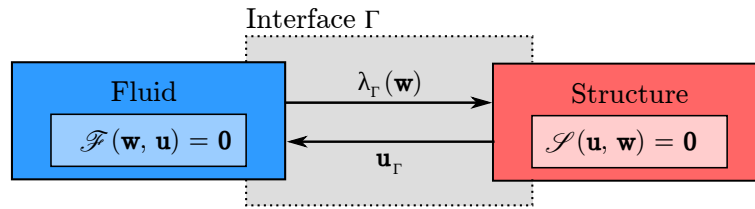


Figure 1.1: Fluid-Structure Interaction problem.

Nobile *et al.* [60] summarise the three major challenges involved in the numerical solution of coupled FSI problems. First, geometrical non-linearity arises from the position of the interface, which is unknown and affects both sub-problems. Second, both the fluid and the structural domains may independently be subject to constitutive non-linearities. Third, the physics of the problem require a careful treatment of the interface continuity and equilibrium conditions.

With regards to the geometrical non-linearities, Ref. [60] states that the interface position can be treated either implicitly, e.g., through Newton or fixed-point iterations [61–68], or explicitly without enforcing coupling conditions at the end of each time period, see for example Refs. [69–71]. Once one of these strategies is adopted, there are two main software architectures that may be implemented to deal with the linearised problem: monolithic and partitioned.

Monolithic approaches solve the fluid and structural equations and impose the interface conditions all at once, allowing therefore an implicit discretisation of the whole system [72]. This approach has been proven to be efficient in some applications in biomechanics [73, 74], mainly because of the so-called added-mass effect [75–77] in haemodynamics and other problems involving incompressible flows and similar fluid and structural densities. It requires, however, the development of highly specialized code as each independent subproblem has its own requirements in terms of the physics involved. This is, according to Piperno [71], a challenging task that in most cases leads to suboptimal solvers. Richter [72] mentions that these strategies are normally regarded as more robust; however, he also mentions that the system matrices yield in most cases very large and ill-conditioned algebraic systems due to the differences in scales between problems. The added-mass effect also affects the conditioning of these systems, which in some cases has led to specific treatments of the problem such as for example pressure segregation [76] or multigrid solvers with partitioned smoothers [72]. Finally, as pointed out by Storti *et al.* [78], the implementation of monolithic schemes can be unmanageable or, at the least, too laborious for large FSI problems on complex geometries, and substantial changes may be required depending on the nature of the problem to be tackled.

Partitioned methods permit the employment of an adequate solver for each subproblem [61]

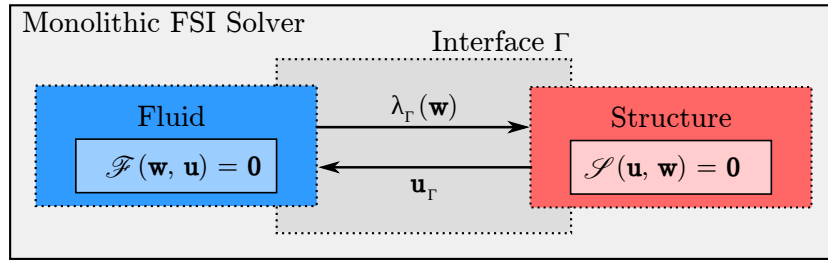


Figure 1.2: Monolithic Fluid-Structure Interaction solver.

which, as pointed out by Joosten *et al.* [79] and Degroote *et al.* [80], allows to solve the coupled problem efficiently with a more specific treatment for each field. Partitioned methods are generally simpler and easier to implement, allowing to reuse pre-existing codes [60, 78] even if they are black boxes [80]. This allows for a high degree of modularity of the solution methods [79] without even requiring a high expertise on the independent solvers.

A partitioned approach based on the 3-field formulation proposed by Farhat *et al.* [81] is described in Fig. 1.3. This method is, according to Maute *et al.* [42] suitable for problems with large structural deformations. The moving boundary, Γ , affects the solution of the fluid field, which no longer may be analysed using a purely Eulerian framework and requires an Arbitrary Lagrangian-Eulerian (ALE) framework [4, 82]. The geometry of the fluid mesh, $\mathcal{M}$, becomes the third field of the problem, which depends on the displacements of the mesh nodes $\mathbf{z}$ due to the deformation at the interface, $\mathbf{u}_\Gamma$. The coupling between the fluid and structural field will be obtained by applying compatibility and continuity over Γ .

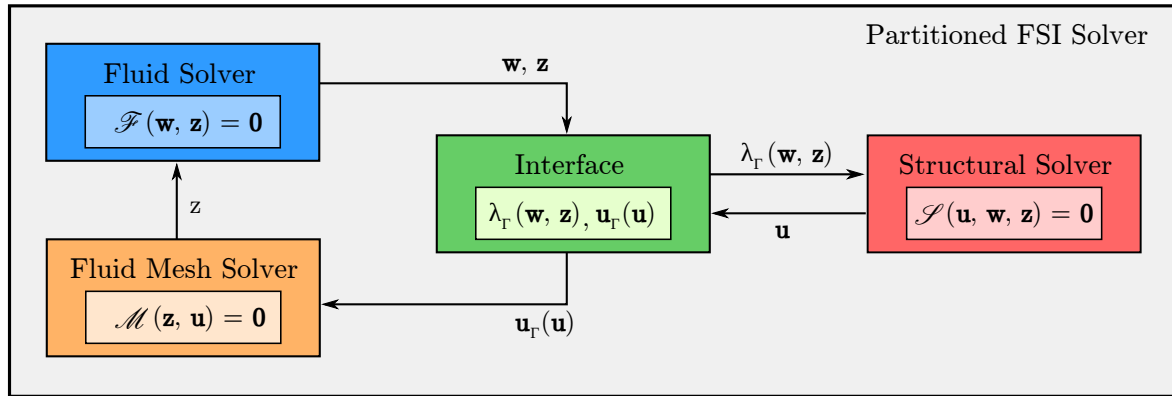


Figure 1.3: Partitioned Fluid-Structure Interaction solver.

Both the fluid and structural subsolvers may involve complex physical behaviour. This results in different computational methods that have arisen specifically oriented to solve each particular subproblem. Meanwhile, the coupled multiphysics responses are also strongly dependent on the kind of problem of study, and the numerical methods need to account for these par-

ticularities. As an example, computational aeroelasticity problems normally deal with heavy, streamlined structures in air flows. The structural vibrations become the dominant parameter of the interaction, which in many cases allows for explicit, loosely-coupled or staggered schemes to be sufficient for the solution of the problem [70, 83].

In the study of biological flows such as blood in vessels [31, 84, 85], incompressible fluids interact with very flexible structures, which may lead to instabilities in the solution process that need to be addressed. Löhner *et al.* [86] recently discussed the complexities in the analysis and design of lightweight, civil structures in atmospheric flow. They assert that this discipline remains behind the aeronautical and naval industries due to the more complex physics involved: the massive separation which may require LES simulations, the unsteadiness of the incoming flow field, the influence of the surrounding bodies and the large deformations of the structural domain are problems that are not normally encountered in the industrial design of ships or aircraft, although the trend in the latest goes towards lighter, more flexible designs [87–89]. These lead to very specialised fluid and structural subsolvers, which justify the adoption of partitioned schemes.

As pointed out by Richter [72], the computational solution of complex FSI problem for realistic applications involves several challenges. The fluid domain alone might be extremely challenging in some cases with separation, complex governing laws or large Reynolds numbers. This may be combined with large deformations on the structural domain, which at the same time displace the fluid discretisation. In most cases, the solution of this kind of problems leads to computationally expensive problems, with large demands on CPU and memory.

1.2.2 Adjoint-based gradient calculation for coupled FSI problems

In this work, we are particularly interested in the optimal design of complex, non-linear physical systems. Let $\mathcal{G}$ be the governing equations of such system,

$$\mathcal{G} = \mathcal{G}(\mathbf{x}(\boldsymbol{\alpha}), \boldsymbol{\alpha}) = 0, \quad (1.1)$$

where $\mathbf{x}$ are the state variables of the system and $\boldsymbol{\alpha}$ is a vector of design variables, which are modified throughout the design process. Note that the state variables may implicitly depend on the design variables, and also that the governing equation (1.1) must be satisfied for any

design $\boldsymbol{\alpha}$. In this context, we can define an objective function, J , as

$$J = J(\mathbf{x}(\boldsymbol{\alpha}), \boldsymbol{\alpha}), \quad (1.2)$$

which is the performance parameter that we aim to improve in the design process. As a result, we define the optimisation problem

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & J(\mathbf{x}(\boldsymbol{\alpha}), \boldsymbol{\alpha}) \\ \text{subject to} \quad & \mathcal{G}(\mathbf{x}(\boldsymbol{\alpha}), \boldsymbol{\alpha}) = 0, \end{aligned} \quad (1.3)$$

where the fulfilment of the governing equation of the physical system becomes a constraint of the optimisation.

A number of optimisation algorithms has been proposed in the literature to solve (1.3) using high-fidelity models. They have been classified by some authors, for example Hernández [90] or Martins *et al.* [91], into two large groups: zero-order (gradient-free) methods and first-order (gradient-based) methods. Zero-order methods generally require a large number of evaluations of the objective function J during the iterative process. For large multiphysics problems, the calculation of the objective function is generally expensive and the number of design parameters may be large. Consequently, the computational cost of these methods rapidly increases, and may become prohibitive.

Surrogate models are able to reduce this cost by building a mathematical approximation to the complex physical system which acts as a low-cost substitute of a costly higher-fidelity model [92]. Some recent applications to coupled FSI systems are, for example, the work of Wu *et al.* in the optimisation of a hydraulic arresting gear [34], or the optimal design in the context of vortex-induced vibration by Farias Filho *et al.* [35]. Buoso [93] developed a reduced-order model for feedback control in electromechanically actuated membrane wings. Liem *et al.* [92] indicate that conventional surrogate models might be insufficient to model highly non-linear problems, such as the transonic drag profile in aerodynamic optimisation of fuel burn efficiency. They develop a mixture of experts approach based on the subdivision of the problem into simple problems (*divide-and-conquer* strategy).

On the other hand, some authors such as Khosravi and Zingg [94] and Zhang *et al.* [95] indicate that fully-coupled high-fidelity models are required in many kinds of problems in order to accurately capture aspects of the non-linear physical behaviour that are overlooked by low

and medium-fidelity models. Gradient-based methods typically require a smaller number of evaluations of the objective function to converge to an optimum. This is an advantage for higher-fidelity models for which the evaluation of J is costly. However, they also demand an accurate computation of the gradient of the objective function with respect to the problem design variables, $dJ/d\boldsymbol{\alpha}$, and a generally smooth design space as they can only guarantee they have converged to a local minimum. Let $\boldsymbol{\alpha}_0$ be an initial guess of the design variables. Gradient methods update the design variables at each stage of the optimisation using an expression of the form [90, 96]

$$\boldsymbol{\alpha}_{k+1} = \boldsymbol{\alpha}_k + \beta_k \mathcal{D}_J \left(\frac{dJ(\mathbf{x}(\boldsymbol{\alpha}_k), \boldsymbol{\alpha}_k)}{d\boldsymbol{\alpha}_k} \right), \quad (1.4)$$

where β_k is the step size for the optimisation step k , and $\mathcal{D}_J$ is the search direction, which is a function of the gradient of J . Consequently, the efficiency of the method will link directly with the accuracy and the cost of computing this gradient.

Martins and Hwang [97] provide a wide review of methods for the calculation of gradients in coupled problems. When optimisation with a large number of design variables is sought, the adjoint method normally becomes the most efficient method, as its cost is independent of the number of parameters of the model. Although well-established methods such as finite differences or the complex step method [41] are able to provide the gradient of the objective function with a (normally) sufficient to a very high accuracy, respectively, the cost of either of them is directly dependent on the number of design variables in the problem. They may require a large number of evaluations of the objective function to compute the gradient, which in turn lessens the benefits of gradient-based optimisation.

Given the advantages of the adjoint method to compute sensitivities with respect to even thousands of design variables, this method has been recently applied in the gradient-based optimal design of very large, industry-relevant, coupled aeroelastic problems. Kenway *et al.* [57] carry out a multipoint aerostructural optimisation of a full aircraft configuration. Abu-Zurayk [58] explores single-point and multi-point adjoint-based optimisation of wing-body configurations. Mishra *et al.* [59] perform an aeroelastic, time-dependent shape optimisation of a helicopter rotor. Zhang *et al.* [95] develop an integrated method for geometrical parametrisation and mesh movement in aerostructural optimisation of aircraft wings, for an improved trade-off between weight and drag. One of the main objectives in the aeronautical industry targets the reduction of carbon emissions from commercial aircraft. Khosravi and Zingg [94] use adjoint methods and gradient-based optimisation to explore aerodynamic and fully-coupled

optimisation of winglets, in order to reduce induced drag of aircraft in cruise.

Adjoint methods are also of application into a number of other problems. Garg *et al.* [98] perform coupled hydrostructural optimisation of a 3D hydrofoil. The goal is to optimise the propellers to increase the energy efficiency in maritime transport. They incorporate stress constraints to the analysis and show that coupled design is necessary, as hydrodynamic optimisation alone yields a different optimisation result that violates such constraints. Adjoint methods can also be applied in topology optimisations in coupled problems. Kreissl *et al.* [99] perform topology optimisation on micro-fluidic devices. The optimal distribution of material in saturated, porous media using coupled analysis is studied by Andreasen and Sigmund [100], using homogenisation theory. Picelli *et al.* [101] perform wet topology optimisation of a wall in a channel and a pressure chamber. Jenkins and Maute [102] use an immersed boundary method for wet and dry topology optimisation of coupled problems. They apply their methodology to problems such as the optimisation of beam supports or a bio-prosthetic aortic health valve. Adjoint methods have also been successfully used for the solution of inverse problems in parameter identification for medical applications by Perego *et al.* [54] and Degroote and co-workers [56]. Other applications of the method range from coupled electrostatic FSI problems [103], to goal-oriented error estimation [104, 105] or steering applications [106], just to name a few.

An efficient calculation of gradients via the adjoint method can also be of great use in the development of surrogate models, particularly in those who seek to incorporate gradient information to enhance the quality of the surrogate. That is the case for gradient-enhanced kriging (or cokriging) methods, which have been successfully used to solve computationally expensive problems such as aerodynamic shape optimisation [107–109]. Kriging models were originally proposed by Krige [110], from whom they take their name. They use a two component model to provide a statistically interpolated response surface [107, 109]. Gradient information can be incorporated to them in order to construct a more accurate and efficient model. The formulation of these methods can be consulted in Refs. [109, 111]. They were successfully used by Chung and Alonso [107] to perform aerodynamic design of a low boom supersonic business jet. Ollar *et al.* [112] show their applicability in the structural optimisation of an aircraft wing, comparing them to other gradient-based and kriging. A comparison between gradient-based methods, kriging and gradient-enhanced kriging is also provided by Laurenceau *et al.* [108]. They show that incorporating gradient information to optimisation methods may be needed

in costly problems such as aerodynamic shape optimisation. In their study, gradient-based methods outperform kriging and cokriging in high-dimensional problems, as too many global searches are required in the latter methods [112]. However, they show a noticeable improvement of gradient-enhanced kriging over regular kriging.

The discussion above suggests that, both for gradient-based optimisation and gradient-enhanced surrogate models, an accurate and efficient computation of the gradients using the adjoint method is required to reduce the overall computational cost of optimising complex, non-linear coupled problems with many design variables. However, solving the adjoint problem is highly challenging in itself. The main complexity of this technique lies behind the fact that it requires to solve a large linear system of equations, whose matrix is the transposed Jacobian of the physical problem. For large, non-linear problems, this Jacobian may not be explicitly available. Sometimes only an approximation to the Jacobian is available, which is normally enough to solve a non-linear problem using Newton iterations, but on the other hand it is not valid for the solution of the adjoint problem. Even when the exact Jacobian is available, the complexity involved in the computation of the transpose of the Jacobian of the flux in a parallel setting [43] may limit the applicability of the adjoint method. The calculation of the Jacobian becomes particularly challenging in multi-disciplinary problems, due to the cross-discipline terms. Because of these issues, the development cost of adjoint methods is usually high. Moreover, new implementations in the primal, physical problem would require a constant update of the adjoint solver, which may be unaffordable. These last two issues were listed by Zingg *et al.* [40] as some of the key disadvantages of gradient-based optimisation methods based on adjoints.

Due to this technical issue, many simplifications of the problem can be found in the literature, that have limited the spectrum of problems tackled. In the fluid side, some applications in aerostructural optimisation neglect turbulent effects [95]. Viscosity is also neglected outright in many cases, see for example [42, 48, 57, 113]. Although there have been some attempts to include turbulent and viscous effects into the analysis, this is a challenging task and its applicability remains limited. From the structural point of view, in most cases only linear elastic approaches have been adopted. This assumption limits the structural deflections to the small deformation regime in several cases [42, 54, 57, 58, 91, 94, 95, 101]. In some cases, even the structural deformations due to the flow effects are neglected, resulting on a one-way coupling [99]. Finally, it has been a common approach to limit the adjoint sensitivity analysis to steady-state problems. There have been some works in adjoint-based parameter

identification for unsteady problems, as for example the work by Degroote *et al.* [56]. However, this implementation was limited to a one-dimensional problem with a limited number of degrees of freedom. It also used a linear elastic structural model. This is due in most cases to the high memory requirements of the solution of the unsteady adjoint, which when the objective function is also dependent of time requires the forward solution of the full time frame before the adjoint solution can be tackled [56]. Perego *et al.* [54] incorporate time-dependent effects, and more recently Mishra *et al.* [59] incorporate time-dependency with a nonlinear bend-twist structural model. In this last case, a full Jacobian for the solution of adjoint is built, where the adjoint computed for 8 sets of variables and involving 8 single-discipline Jacobian blocks in the diagonal and up to 9 crossed-dependent terms. It becomes apparent that the solution of the adjoint in a general framework is challenging and remains a technological issue which limits the progress in optimal design of coupled problems.

In order to overcome some of these issues, Algorithmic Differentiation (AD) techniques [114, 115] can be used. The basic concept behind AD is that any computational code is nothing but a sequence of elementary operations, each of which depends only on one or two variables at a time [114]. Therefore, the problem sensitivities can be evaluated by applying recursively the chain rule to these operations, providing a computational alternative to the analytic evaluation of the Jacobian. AD has been applied to compute discrete adjoints in both fluid [116–118] and structural [119] problems independently. Albring *et al.* recently presented a methodology [120, 121] that applies AD to the solution of aerodynamic problems based on fixed-point iterators. The method brings minimum requirements to the characteristics of the problem or its solution methods. It employs Expression Templates [122] to avoid the large memory requirements and computational cost that results from the application of the most common AD techniques. This technique does not interfere in the future development and improvements of the direct solvers, and has been extended by Zhou *et al.* [123, 124] for aeroacoustic optimisation. The application of AD to FSI adjoint problems has been limited so far to its selective use in some routines that compute the flow residual [117] or some cross-dependencies in the FSI problem [57]. This selective employment complicates the generalisation of the methodology.

1.3 Research Approach

Computational FSI methods have reached a significant level of maturity over the last decades. Important advances have been made in terms of the accuracy, stability and scalability of the simulations, which has led to a progressive incorporation of these techniques at the analysis stage in industrial applications. However, some issues still prevent computational FSI from a widespread use in design stages. The large cost per evaluation of the objective function in coupled problems limits the applicability of gradient-free optimisation methods. On the other hand, gradient-based optimisation methods depend largely on the cost and accuracy of the calculation of the gradients. Very accurate gradients may be obtained using techniques such as the complex-step method, and simpler methods such as standard finite-differences can provide a good accuracy for practical applications. However, the cost of any of these methods depends directly on the number of design variables of the problem, which limits their applicability.

Adjoint methods provide superior performance in the computation of gradients for problems involving a costly evaluation of the objective function and a large number of design variables. The sensitivities with respect to any number of design variables can be directly obtained from the solution of the adjoint equation, whose cost is normally comparable to the solution of the primal problem. The last 15 years have seen successful extensions of the adjoint method to coupled FSI problems, from aeroelasticity to medical applications among other disciplines. Recent applications are capable of dealing with very large, complex multiphysics problems. However, the major drawback of this method is that its implementation depends largely on the analytic derivation of the full Jacobian of the problem, including all of the coupling terms. This task is complex, particularly for strongly non-linear problems where the linearisation of the governing equations can become extremely convoluted. As a result, in most cases approximations or simplifications are made in the definition of the problem, such as neglecting viscous or turbulent effects or limiting the structural deformations to the linear regime.

The current trend is moving towards more and more complex computational models to address industrial problems with physically sound properties. At the same time, the importance of an efficient calculation of the gradients is growing, in parallel with the interest in optimal design. We have observed that the application of the adjoint method to give answer to this issue is still limited, mostly due to the complexity of its implementation in coupled problems. Adjoint methods in coupled solvers have a high associated cost of development of their computational

framework. There is also an important cost of maintenance associated to these techniques, as any addition to the primal solver requires modifications to the adjoint solver. With this in mind, in this work we will be seeking to give answer to the following **research questions**:

1. What are the options to develop an adjoint method that does not require the exact analytical derivation of the Jacobian of the coupled FSI problem? What is the associated development cost?
2. What would there be the main features of a computational framework that could support and integrate such a methodology?
3. Is there any limitation in the derivation of the adjoint imposed by the complexity of the sub-problems?
4. How would this method integrate new developments, such as complex material behaviour, with a contained maintenance cost?
5. What are the advantages of coupled adjoint methods in the resulting system performance?

In this context, we found worth exploring the development and employment of new tools able to give adequate response to these requirements from a general point of view. We considered the open-source software suite SU2 [125–127] as an appropriate starting point for the development of a new, integrated, coupled FSI solver able to provide the solution of the primal and the adjoint problem. SU2 consists of a set of tools that have been developed from the ground up to solve complex multiphysics Partial Differential Equation-based (PDE) problems using unstructured meshes. The code is highly encapsulated and composed of a set of C++ routines organised in a modular way, thus allowing for the integration of new implementations in a structured and efficient manner.

Although it was originally developed as a Computational Fluid Dynamics (CFD) tool aiming to address aerodynamic shape optimisation problems in compressible flows [128–131], SU2 is now being used for analysis and design purposes in several other disciplines. This is thanks to the efforts led by an international consortium¹ composed of an active community of developers that seeks collaborative work in the solution of complex, multidisciplinary problems. Some recent developments within the framework range from the implementation of modelling techniques for non-ideal compressible flows [132–134], to Delayed Detached-Eddy Simulation

¹<http://su2.stanford.edu/>

(DDES) [135] or harmonic balance [136] methods. Recent applications to aerodynamic design optimisation have been carried out by Kline *et al.* [137], Variyar *et al.* [138] or Munguía *et al.* [139], among others.

The suite is natively linked to the open-source library CoDiPack², which makes AD techniques available for any solver in the suite. Algorithmic Differentiation has been employed by Albring *et al.* [120,121] to implement a discrete, AD-based adjoint approach in SU2 for the calculation of gradients in aerodynamic problems. This approach is based on the application of the reverse mode of AD to fixed-point iterators, and will be further described in chapter 3. One of its main advantages is that it minimises the impact of the characteristics of the solution methods over the solution of the adjoint problem. This technique has been extended by Zhou *et al.* [123,124] for aeroacoustic optimisation. As a natural extension of this method, in this work we will introduce a novel, coupled AD-based Discrete Adjoint (ADDA) iterative method to compute accurate gradients for FSI problems involving both viscous flows and non-linear structural behaviour.

1.4 Thesis outline

This first chapter has introduced the problem that will be tackled in this thesis. The current status of the techniques available in the literature for analysis and design of complex FSI problems using computational methods has been discussed, and the motivations and goals for this work have been outlined. The literature review has been kept at a high level, and the specifics of each part of the problem will be discussed in the dedicated chapters and sections. Chapter 2 covers the background equations behind the computational solution of the primal FSI problem and provides a detailed review of the specifics of the FSI problem. The flow equations are presented first, and the strategy adopted in SU2 for their solution is described. The structural equations are then introduced. A native, non-linear structural solver was implemented in SU2 during this project, in agreement with the code structure, and is described in this chapter. This solver was then also natively coupled with the original fluid solver, for the solution of static and dynamic FSI problems. The coupling conditions and available strategies for coupled analysis are reviewed and discussed, and their implementation in SU2 is finally presented [140].

²<http://www.scicomp.uni-kl.de/software/codi/>

In chapter 3 the main strategies available in the literature for gradient calculation are covered. First, the methods based on the definition of derivative and the chain rule are described. The adjoint method is introduced as a reformulation of the optimisation problem using a Lagrangian and the KKT conditions. Then, the AD technique is presented, both in its forward and reverse mode. The applicability of the adjoint method to fluid problems is introduced, and its major advantages and disadvantages are highlighted. An AD-based fluid adjoint previously implemented in SU2, which provides some major advantages over conventional fluid adjoints, is described. This method is first extended within the works in this project to structural problems. Then, the adjoint problem is consistently extended to coupled FSI analysis, in what constitutes the major contribution of this thesis. The implementation initially available in SU2 for flow problems, and its extensions for structural and FSI problems are also described in this chapter [141]. The ability of this method to integrate new developments is tested in chapter 4, by studying the complex material behaviour of the electromechanical actuation of very flexible structures using dielectric elastomers (DEs) [142]. The assumptions made in this work are presented, and the complexity of the calculation of the adjoint of a DE actuator using conventional methods is described. Finally, the direct applicability of the AD-based adjoint method to actuated problems is summarised.

Chapter 5 covers the validation and verification of the primal solver implementations carried out during this project. It first presents a verification of the structural solver under linear conditions, and then under geometrical and material non-linearities, comparing the results against the commercial code ABAQUS. Then, a steady-state FSI problem of a membrane wing in a reattached flow condition is studied [142]. Finally, the time-domain FSI solver is validated with a benchmark test case widely studied in the literature [140, 143].

The sensitivities obtained using the AD-based adjoint method are tested in chapter 6. First, the structural problem is studied and its gradients are tested through their calculation and verification against different methods for a static [141], actuated [142], and dynamic problem. The accuracy of the gradients using our coupled adjoint method is shown then for a fully-coupled FSI problem involving viscous flow and geometrical and material non-linearities in the structural domain [141]. Finally, the method is also demonstrated for a coupled problem involving electric actuation [142].

Chapters 7 and 8 present some examples of gradient-based optimisation in structural and coupled problems using the ADDA method. The gradients obtained using the AD-based ad-

joint method are used in chapter 7 for the optimal design of structural-only problems involving flexible cantilevers. First, a problem involving a parameter identification of the stiffness distribution in the structural domain is studied. Then, the optimal actuation of a flexible cantilever using DEs is analysed. Finally, the optimal distribution of mass within a beam under dynamic conditions is studied, to minimise the time in which the vibrations in the structure are damped out. These benchmark problems demonstrate the applicability of the gradients computed using the adjoint method for the optimisation of non-linear structures.

In chapter 8, the applicability of the coupled primal and adjoint FSI methods described in chapters 2 and 3 is evaluated for the optimal design and electromechanical actuation of a vertical cantilever wall under the action of a viscous flow field. The wall undergoes very large deformations, for which the non-linear structural solver, able to deal with material and geometrical nonlinearities, will be used. The optimisation framework is used in order to obtain the optimal distribution of the stiffness within the wall that is able to achieve pre-defined geometries. Finally, chapter 9 provides a summary of this work, highlighting its major contributions, and concludes outlining some possible future works that may derive from this thesis.

1.5 Publications

The following publications have been written as a result of the works presented in this thesis:

Journal papers

[141] Sanchez, R., Albring, T., Palacios, R., Gauger N.R., Economon, T.D. and Alonso, J.J. (2017) Coupled Adjoint-Based Sensitivities in Large-Displacement Fluid-Structure Interaction using Algorithmic Differentiation, *International Journal for Numerical Methods in Engineering*, <https://doi.org/10.1002/nme.5700>

Conference papers

[140] Sanchez, R., Palacios, R., Economon, T.D., Kline, H.L., Alonso, J.J. and Palacios, F. (2016), Towards a Fluid-Structure Interaction solver for problems with large deformations within the open-source SU2 suite, *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA SciTech*, San Diego, CA, USA, 4-8 January, 2016.

[143] Sanchez, R., Kline, H., Thomas, D., Variyar, A., Righi, M., Economon, T.D., Alonso, J.J., Palacios, R., Dimitriadis, G. and Terrapon, V. (2016), Assessment of the Fluid-Structure

Interaction Capabilities for Aeronautical Applications of the Open-Source Solver SU2, *VII European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2016)*, Crete Island, Greece, 5-10 June, 2016.

[142] Sanchez, R., Palacios, R., Economon, T.D., Alonso, J.J., Albring, T. and Gauger N.R. (2017), Optimal Actuation of Dielectric Membrane Wings using High-Fidelity Fluid-Structure Modelling, *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA SciTech*, Grapevine, TX, USA, 9-13 January, 2017.

[144] Sanchez, R. and Palacios, R. (2017), Computing Derivatives in Nonlinear Aeroelasticity using Algorithm Differentiation, *International Forum on Aeroelasticity and Structural Dynamics (IFASD)*, Como, Italy, 25-28 June, 2017.

Chapter 2

High-fidelity Fluid-Structure Interaction Model

This chapter describes the efforts undergone in this work to establish an appropriate computational framework for FSI modelling with large displacements within SU2. First, the fluid equations are described in section 2.1, along with the solution strategy adopted in SU2. A brief description of the structure of the code is provided, as it determines the properties that must be met by any further development. At the time of starting this project, SU2 did not incorporate means to solve structural problems. Consequently, a finite deformation framework for non-linear continuum mechanics was implemented. The structural equations for geometrical and material non-linear models are presented in section 2.2, along with a description the solution strategy adopted based on standard Finite Elements (FE). The structure of the solver is described in section 2.2.6, and the parallelism with the fluid solver is outlined. Finally, section 2.3 presents the approach adopted for the solution of the FSI problem and its native implementation in SU2, with emphasis in the computational issues derived from the multiphysics nature of the problem.

2.1 Fluid Dynamics

Assuming a viscous, compressible Newtonian flow, we can describe the conservation of mass, momentum and energy using the Navier-Stokes equations,

$$\begin{aligned} \frac{\partial \rho_f}{\partial t} + \nabla \cdot (\rho_f \mathbf{v}) &= 0, \\ \frac{\partial (\rho_f \mathbf{v})}{\partial t} + \nabla \cdot (\rho_f \mathbf{v} \otimes \mathbf{v}) + \nabla p &= \nabla \cdot \boldsymbol{\tau}, \\ \frac{\partial (\rho_f e)}{\partial t} + \nabla \cdot (\rho_f e \mathbf{v} + p \mathbf{v}) &= \nabla \cdot (\boldsymbol{\tau} \cdot \mathbf{v} + \mu C_p \nabla T), \end{aligned} \quad (2.1)$$

where ρ_f is the fluid density, $\mathbf{v}$ the vector of flow velocities in a Cartesian coordinate system, e the total energy of the flow per unit mass, $\boldsymbol{\tau}$ the viscous stress tensor, μ the fluid viscosity, C_p the specific heat and T the temperature. Defining now a conservative variable $\mathbf{w} = (\rho_f, \rho_f \mathbf{v}, \rho_f e)$, it is possible to rewrite the PDE system (2.1) in the fluid domain Ω_f as [125]

$$\partial_t \mathbf{w} + \nabla \cdot \mathbf{F}^c(\mathbf{w}) - \nabla \cdot \mathbf{F}^v(\mathbf{w}) = \mathbf{0} \quad \text{in } \Omega_f, \quad (2.2)$$

where $\mathbf{F}^c(\mathbf{w})$ is the vector of convective fluxes and $\mathbf{F}^v(\mathbf{w})$ corresponds to the viscous fluxes,

$$\mathbf{F}^c(\mathbf{w}) = \begin{pmatrix} \rho_f \mathbf{v} \\ \rho_f \mathbf{v} \otimes \mathbf{v} + \mathbf{I} p \\ \rho_f e \mathbf{v} + p \mathbf{v} \end{pmatrix}, \quad \mathbf{F}^v(\mathbf{w}) = \begin{pmatrix} 0 \\ \boldsymbol{\tau} \\ \boldsymbol{\tau} \cdot \mathbf{v} + \mu_f C_p \nabla T \end{pmatrix}. \quad (2.3)$$

For incompressible flows, the flow equations may be solved using an artificial compressibility formulation [125, 145], where the vector of state variables becomes $\mathbf{w} = (P_f, \rho \mathbf{v})$, P_f is the flow pressure, and the additional artificial compressibility parameter β^2 is introduced into the convective flux, which becomes $\mathbf{F}^c = [\beta^2 \mathbf{v}, \rho_f \mathbf{v} \otimes \mathbf{v} + \mathbf{I} p]^T$.

2.1.1 Solution Strategy

In this work, the solution of (2.2) is tackled using the Finite-Volume-based Euler/N-S/RANS solver in SU2. Let $\mathbf{Z}$ be the coordinates of the nodes of a primal grid that discretises the domain Ω_f , as shown in Fig. 2.1a. SU2 builds a dual-grid, edge-based discretization (Fig. 2.1b) that uses vertex-based control volumes to integrate the semi-discrete form of the fluid problem [125, 126],

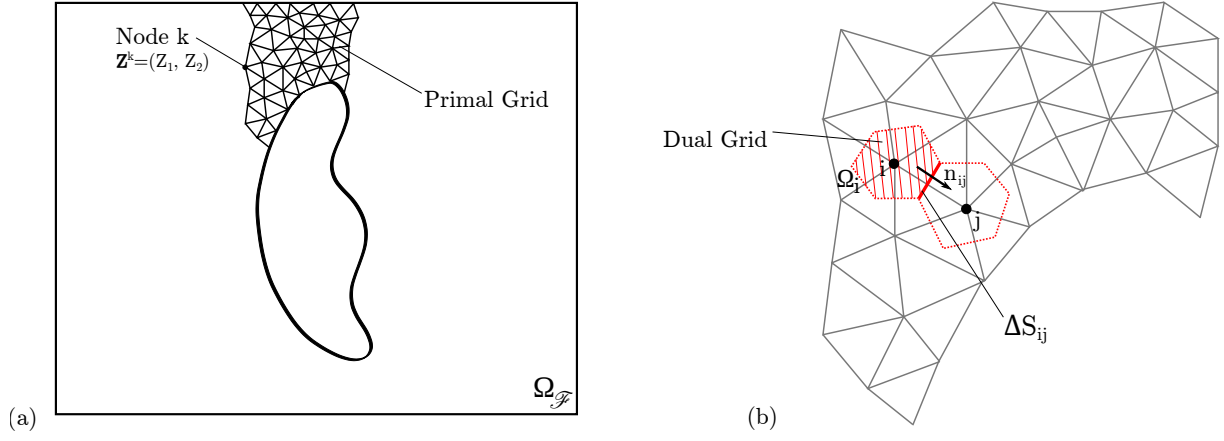


Figure 2.1: Dual grid discretization

$$\int_{\Omega_i^e} \partial_t \mathbf{w} \, d\Omega + \sum_{j \in N(i)} (\tilde{F}_{ij}^c + \tilde{F}_{ij}^v) \Delta S_{ij} = 0, \quad (2.4)$$

where the numerical convective $\tilde{F}_{ij}^c$ and viscous $\tilde{F}_{ij}^v$ fluxes are projected over the edge ij , ΔS_{ij} is the area of the face associated to that edge, $N(i)$ is the set of nodes who are neighbours to i , and Ω_i is the control volume associated to node i . The summation term in the equation corresponds to the spatial residual, which is computed and stored at the nodes. Several upwind and centered methods to compute the convective and viscous terms have been implemented in SU2. The reader may refer to the work by Palacios *et al.* [125, 126] and Economou *et al.* [127] for further details.

For problems involving moving boundaries, as it is the general case for Fluid-Structure Interaction, it is also necessary to account for the deformation of the grid. Let Γ be a boundary of the fluid domain Ω_f . In the discrete domain, the change in position of Γ results not only on a set of displacements at the boundary nodes, $\mathbf{Z}_\Gamma$, but also at the nodes of the internal fluid domain, $\mathbf{Z}_{\Omega_f}$. In order to track the deformation of the mesh, we define the vector of nodal mesh displacements $\mathbf{z}$. The mesh movement problem can be treated as a pseudo-elastic linear problem

$$\tilde{\mathbf{K}}_{\mathcal{M}} \mathbf{z} = \tilde{\mathbf{f}}, \quad (2.5)$$

where $\tilde{\mathbf{K}}_{\mathcal{M}}$ is a fictitious stiffness matrix and $\tilde{\mathbf{f}}$ are the fictitious forces that enforce the displacements of the boundary, $\mathbf{z}_\Gamma$. In order to account for the mesh deformation, the fluid problem (2.2) can be rewritten for moving boundaries as

$$\partial_t \mathbf{w} + \nabla \cdot \mathbf{F}^c(\mathbf{w}, \mathbf{z}) - \nabla \cdot \mathbf{F}^v(\mathbf{w}, \mathbf{z}) = \mathbf{0} \quad \text{in } \Omega_f. \quad (2.6)$$

Regarding time integration, SU2 incorporates both an explicit Runge-Kutta method and an implicit Euler scheme. Focusing on the latter, the semi-discrete equation in (2.4) can be discretized in time for node i using

$$\frac{|\Omega_i^n|}{\Delta t_i^n} \Delta \mathbf{w}_i^n = -\mathbf{R}_i(\mathbf{w}^{n+1}), \quad (2.7)$$

as explained in Ref. [125]. The residual $\mathbf{R}_i(\mathbf{w}^{n+1})$ is unknown, and therefore it must be linearised about t^n . After some algebra, this leads to the expression

$$\left(\frac{|\Omega_i^n|}{\Delta t_i^n} + \frac{\partial \mathbf{R}_i(\mathbf{w}^n)}{\partial \mathbf{w}^n} \right) \Delta \mathbf{w}_i^n = -\mathbf{R}_i(\mathbf{w}^n). \quad (2.8)$$

A value of Δt_i^n is used to relax the problem in (2.8), which is computed in SU2 using a local-time-stepping scheme for each cell in the mesh. The local time step is determined as a function of the Courant-Friedrichs-Lewy (CFL) number and the convective and viscous spectral radii [125].

For unsteady simulations, a dual time-stepping strategy [146] is adopted. The fluid problem is reformulated introducing a fictitious time variable τ that is used to converge a pseudo-steady state problem, resulting in

$$\frac{\partial \mathbf{w}}{\partial \tau} + \mathbf{R}^*(\mathbf{w}) = 0, \quad (2.9)$$

where the modified residual $\mathbf{R}^*(\mathbf{w})$ for a 2^{nd} -order in time scheme is

$$\mathbf{R}^*(\mathbf{w}^n) = \frac{3}{2\Delta t} \mathbf{w}^n + \frac{1}{|\Omega|^n} \left(\mathbf{R}(\mathbf{w}^n) - \frac{2}{\Delta t} \mathbf{w}^{n-1} |\Omega|^{n-1} + \frac{1}{2\Delta t} \mathbf{w}^{n-2} |\Omega|^{n-2} \right), \quad (2.10)$$

Δt is the physical time step and $\mathbf{R}(\mathbf{w})$ the residual of the original equation. In time-domain problems involving moving boundaries, (2.6) is reformulated in an Arbitrary Lagrangian-Eulerian (ALE) framework (see [4, 82, 125, 126]),

$$\partial_t \mathbf{w} + \nabla \cdot \mathbf{F}_{ALE}^c(\mathbf{w}, \mathbf{z}, \dot{\mathbf{z}}) - \nabla \cdot \mathbf{F}^v(\mathbf{w}, \mathbf{z}) = \mathbf{0} \quad \text{in } \Omega_f \times [0, t], \quad (2.11)$$

where the convective term of the equation is rewritten in order to account for the velocities of the nodes of the grid, $\dot{\mathbf{z}}$, as

$$\mathbf{F}_{ALE}^c(\mathbf{w}, \mathbf{z}, \dot{\mathbf{z}}) = \begin{pmatrix} \rho(\mathbf{v} - \dot{\mathbf{z}}) \\ \rho \mathbf{v} \otimes (\mathbf{v} - \dot{\mathbf{z}}) + \mathbf{I}p \\ \rho e(\mathbf{v} - \dot{\mathbf{z}}) + p\mathbf{v} \end{pmatrix}. \quad (2.12)$$

Finally, the node velocities are computed in SU2 using backward differences.

2.1.2 Solver description

SU2 is an object-oriented, C++ code, developed to ease its extension and facilitate the integration of solvers for the solution of multi-physics problems. In Fig. 2.2, an schematic of the fluid solver in SU2 is presented. This structure has been previously described by Palacios *et al.* [125], and full credit is given to the Aerospace Design Lab at Stanford University for its original implementation, and to the SU2 community for the growth of the fluid solver. A short explanation of the role of each class is provided here to understand the particularities and features of the solver, and their implications for the development of the structural code and the coupled Fluid-Structure Interaction solver.

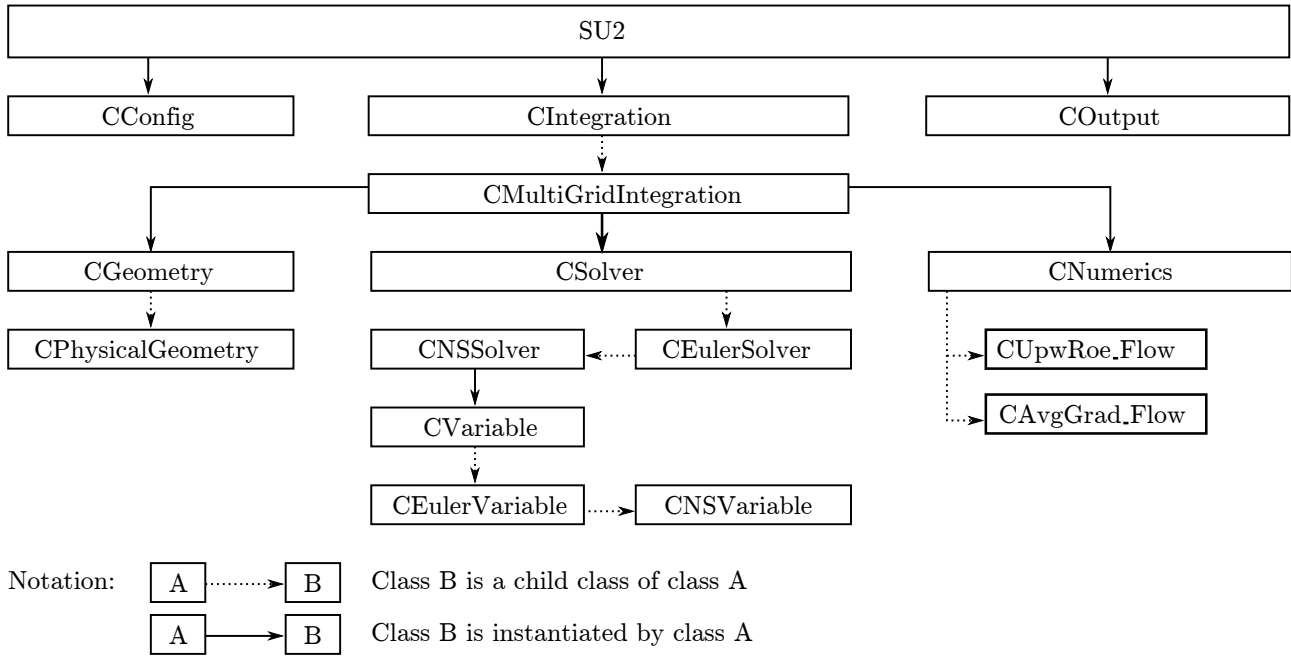


Figure 2.2: Class hierarchy for solving a Navier-Stokes problems.

CConfig Stores the properties of the problem, options and settings from the config file, that will drive the simulations. It is available throughout the whole solution process.

COutput Class responsible for writing the output files.

CIntegration Parent class of *CMultigridIntegration*. Hierarchically solves the Navier-Stokes equations in a Finite Volume context. *CMultigridIntegration* links, for fluid computations, the classes *CGeometry*, *CSolution* and *CNumerics*.

CGeometry Reads the mesh and stores the primal and dual grids, and multigrids if requested.

The child class *CPhysicalGeometry* is responsible for the construction of the dual mesh for Finite Volume formulation from the primal mesh information read from an input file.

CSolver Defines the solution procedure, generating a child class for each different solver in the code. In the fluid problem, *CNSSolution* is a child of *CEulerSolution* because it adds the viscous terms to the solution of the Euler equations.

CNumerics Responsible for the discretisation of each term of the governing equations using the methods required in the config file. For example, *CUpwRoeFlow* includes the numerics of the Roe method for convective fluxes while the computation of the viscous fluxes is included within *CAvgGradCorrectedFlow*.

CVariable Stores the computed information in every node of the grid. Analogous to the solution class, *CNSVariable* stores the viscous terms that are not needed in *CEulerVariable*.

2.2 Structural Mechanics

The description of the continuum mechanics in this work follows the finite deformation framework of Bonet and Wood [147]. Let $\mathcal{B}_0(\mathbf{R})$ be a body described in its reference configuration, $\mathbf{R}$. Let $\Phi = \Phi(\mathbf{R})$ be a function that maps $\mathcal{B}_0$ into a deformed configuration $\mathcal{B}(\mathbf{r})$, where $\mathbf{r}$ is the current position of the body particle. The displacement vector, $\mathbf{u}$, is defined as the difference between the current and the reference configuration, $\mathbf{u} = \mathbf{r} - \mathbf{R}$. This is shown in Fig. 2.3.

The deformation gradient, $\mathbf{F}_\Phi = \partial\Phi/\partial\mathbf{R} = \nabla\Phi$, can be defined as the change in the relative position of two neighbouring particles with respect to their undeformed configuration [147]. For the developments that follow, it is also necessary to define some measure of strain. In the material configuration, we can characterise the strain state by means of the Green strain tensor,

$$\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{I}), \quad (2.13)$$

where $\mathbf{C} = \mathbf{F}_\Phi^T \mathbf{F}_\Phi$ is the right Cauchy-Green deformation tensor. This tensor has its counterpart

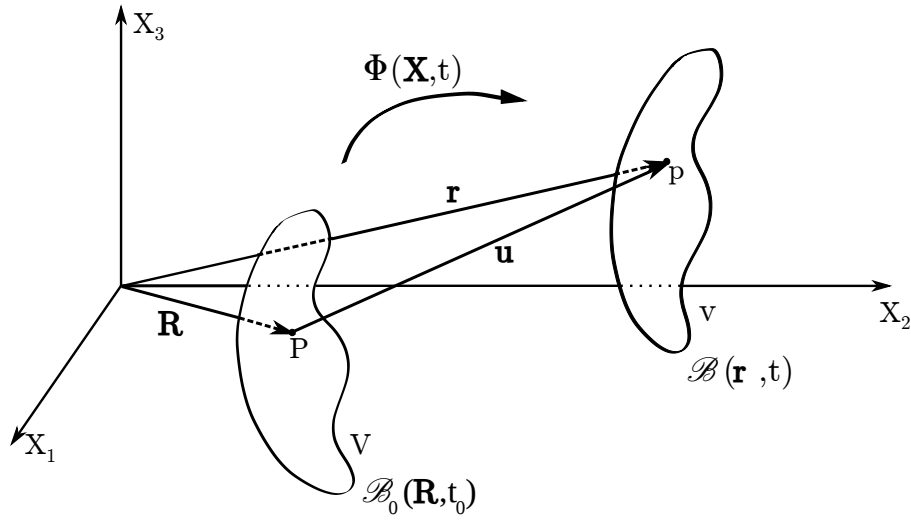


Figure 2.3: Reference and current configuration.

in the current configuration with the Almansi strain tensor,

$$\mathbf{e} = \frac{1}{2}(\mathbf{I} - \mathbf{b}^{-1}), \quad (2.14)$$

where $\mathbf{b} = \mathbf{F}_\Phi \mathbf{F}_\Phi^T$ is the left Cauchy-Green deformation tensor. Both measures of strain are related by a push forward operation,

$$\mathbf{e} = \Phi_*[\mathbf{E}] = \mathbf{F}_\Phi^{-T} \mathbf{E} \mathbf{F}_\Phi^{-1}. \quad (2.15)$$

2.2.1 Nonlinear equilibrium equations

Let $\boldsymbol{\sigma}$ be the Cauchy stress tensor, which defines the stress state of a body in its deformed configuration, and let $\mathbf{f}$ be a set of actuating body forces per unit volume. It is possible to define the pointwise equilibrium condition in the body as

$$\mathbf{s} = \rho_s \ddot{\mathbf{r}} + \text{div } \boldsymbol{\sigma} + \mathbf{f} = 0, \quad (2.16)$$

where $\mathbf{s}$ is an imbalance force which cancels out when equilibrium is met. A finite element solution in continuum mechanics commonly requires a weak form of the equilibrium equation, which can be obtained by the principle of virtual work.

Let $\delta \mathbf{v}$ be an arbitrary, virtual displacement from the current configuration of the body. The virtual work per unit volume caused by the imbalance force $\mathbf{r}$ during the virtual displacement

$\delta \mathbf{v}$ is $\delta w = \mathbf{s} \cdot \delta \mathbf{v}$. Imposing the equilibrium condition in the body $\mathcal{B}$ with volume v , yields

$$\delta W = \int_v \mathbf{s} \cdot \delta \mathbf{v} \, dv = 0. \quad (2.17)$$

Combining now (2.16) and (2.17), and ignoring inertial effects at this stage for simplicity, it is possible to express the equilibrium condition in the current configuration after some algebra [147]

$$\delta W^{curr} = \int_v \boldsymbol{\sigma} : \nabla \delta \mathbf{v} \, dv - \left(\int_v \mathbf{f} \cdot \delta \mathbf{v} \, dv + \int_{\partial v} \mathbf{t} \cdot \delta \mathbf{v} \, da \right) = 0. \quad (2.18)$$

In order to write equation (2.18) in terms of the reference configuration, it is necessary to define the second Piola-Kirchhoff tensor, as a pull back from the Cauchy stress tensor scaled with the determinant of the deformation gradient $J_\Phi = \det(\mathbf{F}_\Phi)$,

$$\mathbf{S}_{PK} = J_\Phi \mathbf{F}_\Phi^{-1} \boldsymbol{\sigma} \mathbf{F}_\Phi^{-T}, \quad (2.19)$$

where J_Φ is the measure of the change in volume, $dv = J_\Phi dV$. Finally, we can define the equilibrium condition in the reference configuration as

$$\delta W^{ref} = \int_V \mathbf{S}_{PK} : \nabla_0 \delta \mathbf{v} \, dV - \left(\int_V \mathbf{f}_0 \cdot \delta \mathbf{v} \, dV + \int_{\partial V} \mathbf{t}_0 \cdot \delta \mathbf{v} \, dA \right) = 0. \quad (2.20)$$

Therefore, the virtual work δW is divided into its internal and external components, as

$$\delta W = \delta W_{int} - \delta W_{ext} = 0. \quad (2.21)$$

2.2.2 Constitutive equations: Hyperelastic materials

Materials are said to be hyperelastic when their behaviour is path-independent and only function of the initial state $\mathbf{R}$ and of the current state of deformation $\mathbf{r}$ [147]. For simplicity, the constitutive model adopted for non-linear materials in this work is considered to be hyperelastic and isotropic.

The behaviour of hyperelastic materials is governed by a strain energy function, Ψ , which determines the work done by the stresses during the deformation from the reference to the current configuration [147]. It is generally expressed in the reference configuration as a function

of the right Cauchy deformation tensor $\mathbf{C}$. The second Piola-Kirchhoff stress tensor can be evaluated from the strain energy function as

$$\mathbf{S}_{PK}(\mathbf{C}(\mathbf{X}), \mathbf{X}) = 2 \frac{\partial \Psi}{\partial \mathbf{C}} = \frac{\partial \Psi}{\partial \mathbf{E}}. \quad (2.22)$$

Given the non-linearity of the problem, it will be necessary to further linearise equation (2.20) to solve the problem incrementally, as it will be done in section 2.2.3. In order to do so, the second Piola-Kirchhoff stress tensor can be linearised in the direction of an increment of the load $\Delta \mathbf{u}$ as

$$D\mathbf{S}_{PK}[\Delta \mathbf{u}] = \mathcal{C} : D\mathbf{E}[\Delta \mathbf{u}], \quad (2.23)$$

where $\mathcal{C}$ is the material elasticity fourth-order tensor, defined as

$$\mathcal{C} = \frac{\partial \mathbf{S}_{PK}}{\partial \mathbf{E}} = 4 \frac{\partial^2 \Psi}{\partial \mathbf{C} \partial \mathbf{C}}. \quad (2.24)$$

One of the simplest cases of hyperelastic materials is the neo-Hookean material model, which at the same time clearly exemplifies the properties previously defined. It is possible to define its strain energy function, second Piola-Kirchhoff stress tensor and material elasticity fourth-order tensors as

$$\Psi_{NH} = \frac{\mu}{2}(\text{tr}(\mathbf{C}) - 3) - \mu \ln J_{\Phi} + \frac{\lambda}{2}(\ln J_{\Phi})^2, \quad (2.25a)$$

$$\mathbf{S}_{PK,NH} = \mu(\mathbf{I} - \mathbf{C}^{-1}) + \lambda(\ln J_{\Phi})\mathbf{C}^{-1}, \quad (2.25b)$$

$$\mathcal{C}_{NH} = \lambda \mathbf{C}^{-1} \otimes \mathbf{C}^{-1} - 2(\mu - \lambda \ln J_{\Phi}) \frac{\partial \mathbf{C}^{-1}}{\partial \mathbf{C}}, \quad (2.25c)$$

where λ and μ are defined as

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \quad , \quad \mu = \frac{E}{2(1+\nu)}, \quad (2.26)$$

in which E is the Young's Modulus of the material in the linear regime and ν the Poisson's ratio.

2.2.3 Linearisation

It is a common approach to adopt a Newton-Raphson method to numerically converge to the equilibrium solution. Considering a trial solution ϕ_k , we can linearise the principle of virtual work as defined in (2.18) or (2.20) in the direction of an increment $\Delta \mathbf{u}$ as [147]

$$\delta W(\phi_k, \delta \mathbf{v}) + D\delta W(\phi_k, \delta \mathbf{v})[\Delta \mathbf{u}] = 0. \quad (2.27)$$

In the reference configuration, the linearisation of the internal virtual work, δW_{int} yields

$$\begin{aligned} D\delta W_{int}^{ref}(\phi_k, \delta \mathbf{v})[\Delta \mathbf{u}] &= \int_V \nabla_0 \delta \mathbf{v} : D\mathbf{S}_{PK}[\Delta \mathbf{u}] dV + \int_V \mathbf{S}_{PK} : D(\nabla_0 \delta \mathbf{v})[\Delta \mathbf{u}] dV, \\ &= \int_V D\mathbf{E}[\delta \mathbf{v}] : \mathcal{C} : D\mathbf{E}[\Delta \mathbf{u}] dV + \int_V \mathbf{S}_{PK} : [(\nabla_0 \Delta \mathbf{u})^T (\nabla_0 \delta \mathbf{v})] dV. \end{aligned} \quad (2.28)$$

Defining now the spatial elasticity tensor in the current configuration, $\mathbf{c}$, as

$$c_{ijkl} = J^{-1} \sum_{I,J,K,L} \mathbf{F}_{\Phi,iI} \mathbf{F}_{\Phi,jJ} \mathbf{F}_{\Phi,kK} \mathbf{F}_{\Phi,lL} \mathcal{C}_{IJKL}, \quad (2.29)$$

we obtain the linearised internal virtual work in the current configuration,

$$\begin{aligned} D\delta W_{int}^{curr}(\phi_k, \delta \mathbf{v})[\Delta \mathbf{u}] &= \int_v \nabla \delta \mathbf{v} : \mathbf{c} : \boldsymbol{\epsilon} dv + \int_v \boldsymbol{\sigma} : [(\nabla(\Delta \mathbf{u}))^T (\nabla \delta \mathbf{v})] dv, \\ &= D\delta W_{int,c}^{curr}(\phi_k, \delta \mathbf{v})[\Delta \mathbf{u}] + D\delta W_{int,\sigma}^{curr}(\phi_k, \delta \mathbf{v})[\Delta \mathbf{u}], \end{aligned} \quad (2.30)$$

where $D\delta W_{int,c}[\Delta \mathbf{u}]$ is the constitutive component of the linearized virtual work and $D\delta W_{int,\sigma}[\Delta \mathbf{u}]$ the initial stress component.

With respect to the external virtual work, the first component of δW_{ext} in (2.18), which corresponds to the body forces, is independent of the increment $\Delta \mathbf{u}$, thus $D\delta W_{ext}^{body}[\Delta \mathbf{u}] = 0$. The second component, however, relates to the external forces applied in the boundary of the solid, and yields a non-zero linearised term. Defining the traction over the boundary as $\mathbf{t} = \boldsymbol{\sigma} \mathbf{n}$, where $\mathbf{n}$ is the normal to the surface, and parametrizing the boundary ∂v into a parent domain $A_\xi = A_\xi(\xi, \eta)$, it is possible to rewrite the normal and the boundary da as [147]

$$\mathbf{n} = \frac{\frac{\partial \mathbf{r}}{\partial \xi} \times \frac{\partial \mathbf{r}}{\partial \eta}}{\left\| \frac{\partial \mathbf{r}}{\partial \xi} \times \frac{\partial \mathbf{r}}{\partial \eta} \right\|}, \quad da = \left\| \frac{\partial \mathbf{r}}{\partial \xi} \times \frac{\partial \mathbf{r}}{\partial \eta} \right\| d\xi d\eta, \quad (2.31)$$

which yields, after some algebra,

$$D\delta W_{ext}(\phi_k, \delta \mathbf{v})[\Delta \mathbf{u}] = \int_{A_\xi} \boldsymbol{\sigma} \left[\frac{\partial \mathbf{r}}{\partial \eta} \cdot \left(\delta \mathbf{v} \times \frac{\partial \Delta \mathbf{u}}{\partial \xi} \right) - \frac{\partial \mathbf{r}}{\partial \xi} \cdot \left(\delta \mathbf{v} \times \frac{\partial \Delta \mathbf{u}}{\partial \eta} \right) \right] d\xi d\eta. \quad (2.32)$$

This term is non-symmetric in terms of the magnitudes $\delta \mathbf{v}$ and $\Delta \mathbf{u}$, which will break the symmetry of the tangent matrix.

2.2.4 Discretisation

In this work, a finite element isoparametric discretisation is adopted. The position of any particle in the reference $\mathbf{R}$ or in the current $\mathbf{r}$ configuration may be expressed as a function of the position of the nodes a as

$$\mathbf{R} = \sum_{a=1}^{n_{nodes}} N_a(\xi, \eta, \zeta) \mathbf{R}_a, \quad \mathbf{r} = \sum_{a=1}^{n_{nodes}} N_a(\xi, \eta, \zeta) \mathbf{r}_a, \quad (2.33)$$

where $N_a(\xi, \eta, \zeta)$ are the shape functions in the parent coordinates. Recalling the expression for internal virtual work (2.18), we can define the discrete problem in the current configuration for node a , $\mathcal{S}_a(\mathbf{u}) = 0$, $\mathbf{u} = \mathbf{r} - \mathbf{R}$, as

$$\mathcal{S}_a(\mathbf{u}) = \int_v \boldsymbol{\sigma} : \nabla \delta \mathbf{v} \, dv - \left(\int_v \mathbf{f} \cdot \delta \mathbf{v} \, dv + \int_{\partial v} \mathbf{t} \cdot \delta \mathbf{v} \, da \right) = 0, \quad (2.34a)$$

$$= \sum_{\forall e|a \in e} \int_{v^e} \mathbf{B}_a^T \underline{\boldsymbol{\sigma}} \, dv - \sum_{\forall e|a \in e} \left(\int_v N_a \mathbf{f} \, dv + \int_{\partial v} N_a \mathbf{t} \, da \right) = 0, \quad (2.34b)$$

$$= \mathbf{T}_a(\mathbf{u}) - \mathbf{F}_{b,a} - \mathbf{F}_{\Gamma,a}(\mathbf{u}) = 0, \quad (2.34c)$$

where the matrix $\mathbf{B}_a$ and the stress vector $\underline{\boldsymbol{\sigma}}$ are defined as in [147],

$$\mathbf{B}_a = \begin{bmatrix} \frac{\partial N_a}{\partial r_1} & 0 & 0 \\ 0 & \frac{\partial N_a}{\partial r_2} & 0 \\ 0 & 0 & \frac{\partial N_a}{\partial r_3} \\ \frac{\partial N_a}{\partial r_2} & \frac{\partial N_a}{\partial r_1} & 0 \\ \frac{\partial N_a}{\partial r_3} & 0 & \frac{\partial N_a}{\partial r_1} \\ 0 & \frac{\partial N_a}{\partial r_3} & \frac{\partial N_a}{\partial r_2} \end{bmatrix}, \quad \underline{\boldsymbol{\sigma}} = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{bmatrix}. \quad (2.35)$$

The full discrete structural non-linear problem may be written as

$$\mathcal{S}(\mathbf{u}) = \mathbf{T}(\mathbf{u}) - \mathbf{F}_b - \mathbf{F}_\Gamma(\mathbf{u}) = \mathbf{0}, \quad (2.36)$$

which is linearised as

$$\frac{\partial \mathcal{S}(\mathbf{u})}{\partial \mathbf{u}} \Delta \mathbf{u} = -\mathcal{S}(\mathbf{u}), \quad (2.37)$$

and then solved iteratively using a Newton method,

$$\mathbf{K}^i \Delta \mathbf{u}^{i+1} = -\mathcal{S}(\mathbf{u}^i), \quad (2.38a)$$

$$\mathbf{u}^{i+1} = \mathbf{u}^i + \Delta \mathbf{u}^{i+1}, \quad (2.38b)$$

until $\mathcal{S}(\mathbf{u}) \approx \mathbf{0}$. The tangent matrix $\mathbf{K} = \partial \mathcal{S}(\mathbf{u}) / \partial \mathbf{u}$ may be assembled as [147]

$$\mathbf{K} = \mathbf{K}_c + \mathbf{K}_\sigma - \mathbf{K}_{ext}, \quad (2.39)$$

where $\mathbf{K}_c$, $\mathbf{K}_\sigma$ and $\mathbf{K}_{ext}$ are, respectively, the constitutive, initial stress and external load components of the tangent matrix. For nodes a and b within an element e , the internal work components may be written as [147, 148]

$$\mathbf{K}_{c,ab}^e = \int_v \nabla \delta \mathbf{v} : \mathbf{c} : \boldsymbol{\epsilon} \, dv = \int_{v^e} \mathbf{B}_a^{eT} \mathbf{D}^e \mathbf{B}_b^e \, dv^e, \quad (2.40a)$$

$$\mathbf{K}_{\sigma,ab}^e = \int_v \boldsymbol{\sigma} : [(\nabla(\Delta \mathbf{u}))^T (\nabla \delta \mathbf{v})] \, dv = \int_{v^e} (\nabla \mathbf{N}_a^e \cdot \boldsymbol{\sigma}^e \nabla \mathbf{N}_b^e) \mathbf{I} \, dv^e. \quad (2.40b)$$

The indices of the matrix $\mathbf{D}_{ij}$ correspond with those of the spatial constitutive tensor $\mathbf{c}_{ijkl}$ using the mapping $\{1, 2, 3, 4, 5, 6\} \leftarrow \{11, 22, 33, 12, 13, 23\}$. Finally, the component corresponding to the linearisation of the external work may be expressed as

$$\mathbf{K}_{ext,ab}^e = \int_{A_\xi} \boldsymbol{\sigma} \left[\frac{\partial \Delta \mathbf{u}}{\partial \eta} \cdot \left(\delta \mathbf{v} \times \frac{\partial \Delta \mathbf{u}}{\partial \xi} \right) - \frac{\partial \Delta \mathbf{u}}{\partial \xi} \cdot \left(\delta \mathbf{v} \times \frac{\partial \Delta \mathbf{u}}{\partial \eta} \right) \right] d\xi d\eta, \quad (2.41a)$$

$$= \mathcal{E} \int_{A_\xi} \boldsymbol{\sigma} \left[\frac{\partial \Delta \mathbf{u}}{\partial \eta} \cdot \left(N_a \frac{\partial N_b}{\partial \xi} \right) - \frac{\partial \Delta \mathbf{u}}{\partial \xi} \cdot \left(N_a \frac{\partial N_b}{\partial \eta} \right) \right] d\xi d\eta, \quad (2.41b)$$

where $\mathcal{E}$ is the three-dimensional Levi-Civita permutation symbol.

2.2.5 Time integration

Assuming no structural damping, inertial effects may be added from Eq. (2.16) into Eq. (2.38) by including the product of the mass matrix $\mathbf{M}$ premultiplying the acceleration undergone by the structural particles, as in (2.42)

$$\begin{aligned}\mathbf{M}\ddot{\mathbf{u}}^t + \mathbf{K}^t \Delta \mathbf{u}_k^t &= \mathbf{F}_b + \mathbf{F}_\Gamma(\mathbf{u}_{k-1}^t) - \mathbf{T}(\mathbf{u}_{k-1}^t), \\ \mathbf{u}_k^t &= \mathbf{u}_{k-1}^t + \Delta \mathbf{u}_k^t,\end{aligned}\tag{2.42}$$

where the tangent matrix $\mathbf{K}$ has been defined in (2.39) and the mass matrix, may be written as [147, 148]

$$\mathbf{M}_{ab}^e = \int_{\Omega_s^e} \rho_s^e \mathbf{N}_a^{eT} \mathbf{N}_b^e d\Omega_s^e.\tag{2.43}$$

The Newmark [149] integration method has been used for time integration. It relies on the approximations

$$\begin{aligned}\mathbf{u}^t &= \mathbf{u}^{t-1} + \Delta t \dot{\mathbf{u}}^{t-1} + \left[\left(\frac{1}{2} - \beta\right) \ddot{\mathbf{u}}^{t-1} + \beta \ddot{\mathbf{u}}^t\right] \Delta t^2, \\ \dot{\mathbf{u}}^t &= \dot{\mathbf{u}}^{t-1} + [(1 - \gamma) \ddot{\mathbf{u}}^{t-1} + \gamma \ddot{\mathbf{u}}^t] \Delta t.\end{aligned}\tag{2.44}$$

Combining the expressions in (2.44), the acceleration may be approximated as

$$\ddot{\mathbf{u}}^t = \frac{1}{\beta \Delta t^2} (\mathbf{u}^t - \mathbf{u}^{t-1}) - \frac{1}{\beta \Delta t} \dot{\mathbf{u}}^{t-1} - \frac{1 - 2\beta}{2\beta} \ddot{\mathbf{u}}^{t-1}.\tag{2.45}$$

The Newmark method imposes equilibrium at the end of the time step according to Eq. (2.42). Substituting (2.44) and (2.45) into (2.42) we obtain that, for each iteration of the non-linear solver, we need to solve the system

$$\mathbf{K}_T^t \Delta \mathbf{u}_k^t = \mathbf{F}_b + \mathbf{F}_\Gamma(\mathbf{u}^t) - \mathbf{T}(\mathbf{u}^t) + \mathbf{R}_T(\mathbf{u}_{k-1}^t, \mathbf{u}^{t-1}, \dot{\mathbf{u}}^{t-1}, \ddot{\mathbf{u}}^{t-1}),\tag{2.46}$$

where the effective stiffness matrix is defined as

$$\mathbf{K}_T^t = \mathbf{K}^t + \frac{1}{\beta \Delta t^2} \mathbf{M},\tag{2.47}$$

and the time-dependent residual component $\mathbf{R}_T$ is computed as

$$\mathbf{R}_T = \mathbf{M} \left[\frac{1}{\beta \Delta t^2} (\mathbf{u}^{t-1} - \mathbf{u}_{k-1}^t) + \frac{1}{\beta \Delta t} \dot{\mathbf{u}}^{t-1} + \left(\frac{1}{2\beta} - 1 \right) \ddot{\mathbf{u}}^{t-1} \right]. \quad (2.48)$$

According to G eradin and Rixen [150], a value of numerical damping $\delta > 0$ may be introduced in this scheme to damp out undesirable high frequency oscillations. This is obtained by setting the values of β and γ according to

$$\gamma = \frac{1}{2} + \delta, \quad \beta = \frac{1}{4} \left(\frac{1}{2} + \gamma \right)^2. \quad (2.49)$$

2.2.6 Solver implementation

In this work, a Finite-Element Method based, non-linear structural solver has been implemented in SU2, following the same philosophy of modularity as the original fluid solver. This flexible structure of the code allows for sustainability, as new features may be added to address more complex and specialized problems just by developing and expanding the capabilities. The main modifications in the code affect the *CSolver* and *CNumerics* classes, as shown in Fig. 2.4.

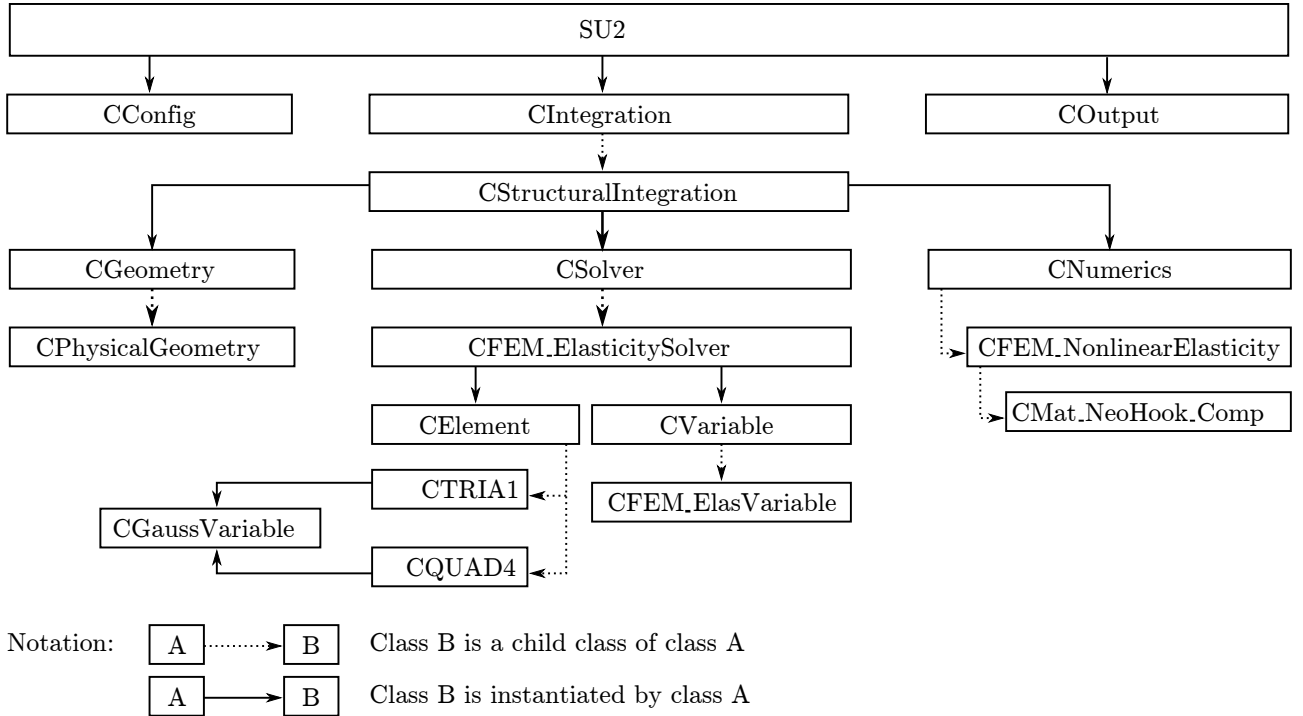


Figure 2.4: Class hierarchy for solving a two-dimensional, non-linear structural problem.

CStructuralIntegration Hierarchically solves the Finite Element problem.

CSolver Instantiates the variable *CFEM_ElasVariable*, and the containers for the Finite Element formulation, *CElement*. Carries out the assembly of the tangent matrix and the imposition of boundary conditions.

CNumerics The numerics class has two levels of inheritance:

- Geometrical part (Fig. 2.6): The child classes, namely *CFEM_LinearElasticity* and *CFEM_NonlinearElasticity*, compute the different terms of the tangent matrix, the mass matrix and the stress terms on the right hand side of Eq. (2.38) at the element level.
- Material part (Fig. 2.7): Contains the equations that define the material model to be employed. The interface with the geometrical part of the problem is the computation of the constitutive matrix $\mathbf{D}$ and the Cauchy stress tensor $\boldsymbol{\sigma}$.

CVariable Stores the displacements $\mathbf{u}$, obtained from the solution of Eq. (2.38), as well as the velocities $\dot{\mathbf{u}}$ and accelerations $\ddot{\mathbf{u}}$ in the case of a dynamic problem as defined in Eq. (2.42)

CElement Hosts the common properties of the FEM structure. One child class of *CElement* is defined for each relevant kind of element. The current implementation is limited to linear, first order elements. The list can be easily extended to include improved elements for different applications not yet envisioned. The full list available at the time of writing is shown in Fig. 2.5.

CGaussVariable In order to perform Gaussian integration, several relevant quantities, such as the deformation gradient $\mathbf{F}$, the Cauchy stress tensor $\boldsymbol{\sigma}$ or the constitutive matrix $\mathbf{D}$, are computed at the Gaussian Points. Also, the element shape functions $\mathbf{N}$ and their gradients $\nabla \mathbf{N}$ need to be computed at these locations. The value of these quantities is stored in this auxiliary container, common for all types of elements.

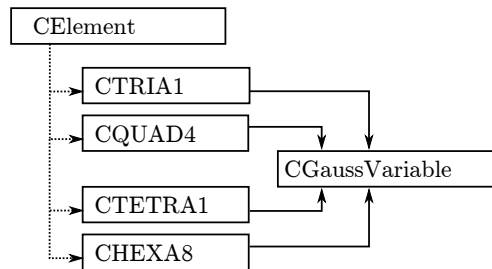


Figure 2.5: List of child classes for the *CElement* class.

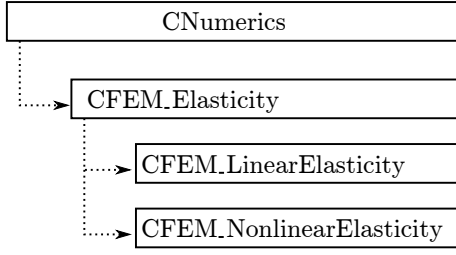


Figure 2.6: Child class structure for the definition of geometrical non-linearities.

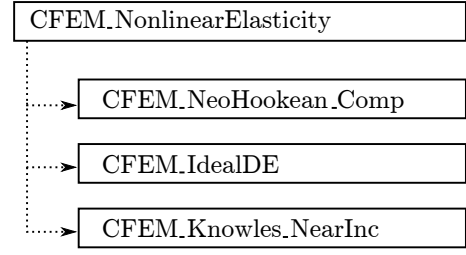


Figure 2.7: Child class structure for the definition of material non-linearities.

2.3 Coupled Fluid-Structure Interaction Problem

Adopting a partitioned approach for Fluid-Structure Interaction requires imposing compatibility and continuity conditions at the FSI interface $\Gamma = \Omega_f \cap \Omega_s$. Continuity of displacements is defined as

$$\mathbf{u}_\Gamma = \mathbf{z}_\Gamma, \quad (2.50)$$

while the equilibrium of tractions over the interface requires that

$$\boldsymbol{\lambda}_f + \boldsymbol{\lambda}_s = 0, \quad (2.51)$$

where $\boldsymbol{\lambda}_f$ and $\boldsymbol{\lambda}_s$ are, respectively, the tractions over the fluid and structural sides of Γ . They may be defined as

$$\begin{cases} \boldsymbol{\lambda}_f = -p\mathbf{n}_f + \boldsymbol{\tau}_f\mathbf{n}_f & \text{on } \Gamma_f, \\ \boldsymbol{\lambda}_s = \sigma_s\mathbf{n}_s & \text{on } \Gamma_s, \end{cases} \quad (2.52)$$

where $\mathbf{n}_f$ and $\mathbf{n}_s$ are the dimensional, outward normals to the fluid and structural sides of Γ (including area information).

We define the structural, fluid mesh and fluid problems respectively as $\mathcal{S}$, $\mathcal{M}$ and $\mathcal{F}$. Using (2.50) and (2.52), we can write the governing equations of the coupled FSI problem as a function of the problem state variables, $\mathbf{u}$, $\mathbf{w}$ and $\mathbf{z}$, and their time derivatives,

$$\mathcal{G}(\mathbf{u}, \mathbf{w}, \mathbf{z}) = \begin{cases} \mathcal{S}(\mathbf{u}, \mathbf{w}, \mathbf{z}) = \mathbf{T}(\mathbf{u}) - \mathbf{F}_b - \mathbf{F}_\Gamma(\mathbf{u}, \mathbf{w}, \mathbf{z}) + \rho_s \ddot{\mathbf{u}} & = \mathbf{0}, \\ \mathcal{F}(\mathbf{w}, \mathbf{z}) = \dot{\mathbf{w}} + \nabla \cdot \mathbf{F}^c(\mathbf{w}, \mathbf{z}, \dot{\mathbf{z}}) - \nabla \cdot \mathbf{F}^v(\mathbf{w}, \mathbf{z}) & = \mathbf{0}, \\ \mathcal{M}(\mathbf{u}, \mathbf{z}) = \tilde{\mathbf{K}}_{\mathcal{M}}\mathbf{z} - \tilde{\mathbf{f}}(\mathbf{u}) & = \mathbf{0}. \end{cases} \quad (2.53)$$

2.3.1 Coupling Strategy

The advantages and disadvantages of monolithic and partitioned approaches were discussed in section 1.2.1. Given that in this project we are planning to deal with arbitrarily complex physics for both subproblems and for the coupled FSI framework, a partitioned approach has been considered to be more appropriate in order to maintain modularity in the sub-solvers. In this context, two different strategies may be used for the integration in time of the coupled problem. On the one hand, it is possible to adopt a loosely-coupled, or staggered, procedure. In general, this approach consists of sequentially solving the fluid and structure problems, advancing the solvers in time without enforcing the coupling conditions at the end of each time step. On the other hand, strongly-coupled strategies are also solved sequentially, but they require both solvers to meet the coupling conditions defined in the previous section at every single time step, generally by means of subiterations on the coupled solver. In this section, we will review the literature available for both strategies and discuss their implications. A strongly-coupled, non-linear Block Gauss-Seidel algorithm with predictor and relaxation has been implemented in SU2. The structure of the code is then presented in section 2.3.2.

Loosely-coupled methods

From the point of view of the coupled problem, we can define as loosely-coupled or *staggered* algorithms those who couple the fluid and the structural solver in an explicit manner. This strategy is the most elementary technique in order to solve the coupled problem, and due to its simple and modular implementation, it is widely used in Fluid-Structure Interaction problems, especially in the field of computational aeroelasticity [69–71, 151].

The simplest partitioned algorithm is the Conventional Serial Staggered (CSS) [69] procedure (algorithm 1), that consists of one single solution of each subsolver per time step. This is a collocation method, that is, each subfield has to be evaluated at the end of the same time window [61, 71].

This procedure has several drawbacks: First of all, the kinematic continuity condition is not enforced, as the fluid is computed for $t + \Delta t$ with the position of the interface at time t . This results in a first order energy-accurate scheme, according to Piperno and Farhat [71], no matter the order of each subsolver independently.

As a result of this lack of energy equilibrium, spurious energy is generated during the sim-

Algorithm 1: Conventional Serial Staggered (CSS) method

```

while  $t < T$  do
    Mesh update:  $\mathcal{M}(\mathbf{u}^t, \mathbf{z}^{t+1}) = 0$ 
    Fluid iteration:  $\mathcal{F}(\mathbf{w}^{t+1}, \mathbf{z}^{t+1}) = 0$ 
    Structural iteration:  $\mathcal{S}(\mathbf{u}^{t+1}, \mathbf{w}^{t+1}, \mathbf{z}^{t+1}) = 0$ 
     $t \leftarrow t + 1$ 

```

ulation, which may eventually result in large discrepancies in the prediction of pressures and viscous stresses on the interface [152]. Piperno and Farhat [71] carry out an assessment of the effects of this artificially generated energy derived from the staggering process, by comparing the work done by the fluid pressures and viscous stresses with the work of the structural forces. One of the conclusions that may be extracted from their study is that it is possible to improve the accuracy of a staggered scheme by using a predictor for the structural displacements at the beginning of the simulation. Ref. [71] also asserts that second order energy accurate schemes may be obtained using a first order time accurate predictor, and even third order energy accurate methods can be achieved if conservation of momentum is enforced together with the use of a second order predictor. The resulting algorithm is the so-called Generalized Serial Staggered (GSS) [71], as referred in algorithm 2 [61], where the displacement predictors $\mathcal{P}_1(\mathbf{u}_f^t)$ (order 1) and $\mathcal{P}_2(\mathbf{u}_f^t)$ (order 2) may be defined as

$$\begin{aligned}
 \mathcal{P}_1(\mathbf{u}^t) &= \mathbf{u}^t + \Delta t \dot{\mathbf{u}}^t, \\
 \mathcal{P}_2(\mathbf{u}^t) &= \mathbf{u}^t + \frac{1}{2} \Delta t (3\dot{\mathbf{u}}^t - \dot{\mathbf{u}}^{t-1}).
 \end{aligned}
 \tag{2.54}$$

Algorithm 2: Generalized Serial Staggered (GSS)

```

while  $t < T$  do
    Predictor:  $\tilde{\mathbf{u}}^{t+1} = \mathcal{P}_i(\mathbf{u}^t)$  ;
    Mesh update:  $\mathcal{M}(\tilde{\mathbf{u}}^{t+1}, \mathbf{z}^{t+1}) = 0$  ;
    Fluid iteration:  $\mathcal{F}(\mathbf{w}^{t+1}, \mathbf{z}^{t+1}) = 0$  ;
    Structural iteration:  $\mathcal{S}(\mathbf{u}^{t+1}, \mathbf{w}^{t+1}, \mathbf{z}^{t+1}) = 0$  ;
     $t \leftarrow t + 1$ ;

```

Some other attempts have been made to develop a higher order staggered scheme, for example

the Combined Interface Boundary Conditions (CIBC) method developed by Jaiman et. al [152], or the works of Dettmer and Peric [77] involving a traction predictor and a corrector step. Despite these efforts, some instabilities and inaccuracies are inherent to loosely-coupled schemes and have been reported throughout the literature.

In particular, and due to the issues previously described, the stability of these schemes is highly dependent on the density ratio ρ_s/ρ_f and the compressibility of the flow [61, 62, 77]. Incompressible flows combined with low density ratios generally require strongly coupled solvers due to the added-mass effect [75, 77]. Moreover, thin-walled structures which undergo large deformations are also very sensitive to the quality of the solution scheme, as explained in Ref. [62], where the authors state that “small parameter changes in data or even slight initial numerical deviations may lead to a completely different structural response”, thus making it necessary to use iterative solution schemes in order to achieve accurate results.

Strongly-coupled methods

In order to prevent the instabilities derived from the adoption of staggered schemes, it is common to adopt a strongly-coupled method [61–67]. These methods intend, in various ways, to eliminate the energy errors due to the lack of enforcement of the coupling conditions, by imposing (2.50) and (2.51) at the end of each time step.

Due to the non-linear nature of the coupled problem, it becomes natural to think of Newton-Raphson methods to obtain the coupled solution (see, e.g., refs. [64, 68]). The problem defined in Eq. (2.53) may be linearised within a time step writing it in the form of a **Block-Newton** iteration

$$\begin{bmatrix} \frac{\partial \mathcal{S}}{\partial \mathbf{u}} & \frac{\partial \mathcal{S}}{\partial \mathbf{w}} & \frac{\partial \mathcal{S}}{\partial \mathbf{z}} \\ \mathbf{0} & \frac{\partial \mathcal{F}}{\partial \mathbf{w}} & \frac{\partial \mathcal{F}}{\partial \mathbf{z}} \\ \frac{\partial \mathcal{M}}{\partial \mathbf{u}} & \mathbf{0} & \frac{\partial \mathcal{M}}{\partial \mathbf{z}} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u} \\ \Delta \mathbf{w} \\ \Delta \mathbf{z} \end{bmatrix} = - \begin{bmatrix} \mathcal{S}(\mathbf{u}, \mathbf{w}, \mathbf{z}) \\ \mathcal{F}(\mathbf{w}, \mathbf{z}) \\ \mathcal{M}(\mathbf{u}, \mathbf{z}) \end{bmatrix}. \quad (2.55)$$

It becomes evident from (2.55) the need to compute or estimate the partial derivatives of the structural, fluid mesh and fluid problems with respect to each domain’s own variables, and also with respect to the crossed variables. This procedure is complex and computationally expensive, and even though convergence should be quadratic if the exact Jacobian is computed and thus converge in few iterations, the cost per iteration is high.

The **Interface Newton-Krylov** method [62, 66] reduces this cost, by defining the fixed-

point iterator $\mathbf{R}(\mathbf{u}_\Gamma) = \mathcal{S}(\mathbf{u}_\Gamma, \mathcal{F}(\mathbf{u}_\Gamma), \mathcal{M}(\mathbf{u}_\Gamma)) - \mathbf{u}_\Gamma = \mathbf{0}$. The problem state variables are reduced to the interface displacements $\mathbf{u}_\Gamma$, and the non-linear problem

$$\frac{\partial \mathbf{R}^n(\mathbf{u}_\Gamma)}{\partial \mathbf{u}_\Gamma} \Delta \mathbf{u}_\Gamma^n = -\mathbf{R}_\Gamma^n, \quad \mathbf{u}_\Gamma^{n+1} = \mathbf{u}_\Gamma^n + \Delta \mathbf{u}_\Gamma^n, \quad (2.56)$$

is solved. In order to avoid complex and the time consuming computation of the exact Jacobian $\partial \mathbf{R}(\mathbf{u}_\Gamma)/\partial \mathbf{u}_\Gamma$, some alternatives have been proposed to compute the matrix-vector product of the Jacobian with an arbitrary vector. One possibility is to carry out this computation through finite-difference methods (see, e.g., [62, 153]). Another option is to approximate this computation by means of a reduced-order model (ref. [66, 154]), using residual and displacements information from previous iterations.

One of the most widely used strongly-coupled methods is the **Block Gauss-Seidel** (BGS) method, which implies the iterative solution of fluid, mesh and structural subsolvers within a time step until a tolerance criterion ϵ is met in the residual $\mathcal{R}_\mathbf{u} = \|\mathbf{u}_{\Gamma,k+1} - \mathbf{u}_{\Gamma,k}\| < \epsilon$. Following the non-linear BGS (NLBGS) procedure presented by Barcelos and Maute [43], the upper-diagonal terms that couple the problem in (2.55) may be neglected, and the independent solvers can be run sequentially.

Some authors have reported slow convergence or even divergence when the Fluid-Structure Interaction is strong, due for example to low density ratios, incompressible flows or large structural deformations (see, e.g., [62, 66]). As a result, and in order to improve the convergence of the scheme, higher order displacement predictors analogous to the described in equation 2.54 must be used, combined with the employment of relaxation techniques (see [61, 62, 65, 67]). The resultant procedure is described in algorithm 3.

In order to obtain the fastest possible convergence, it is necessary to appropriately define the relaxation parameter ω_n . Küttler and Wall [65] carry out an interesting review on the possible values of ω_n . As the simplest option, a fixed relaxation parameter can prevent the computation from diverging, as long as it is small enough. Nevertheless, this choice is in most cases inefficient, as it results in a large number of iterations due to the inability to use the most of the previous solution. Also, it is mentioned in ref. [65] that the optimal value for a fixed ω_n is problem-dependent.

A simple and efficient option, as reported in several papers [62, 65, 67] is to use the dynamic

Algorithm 3: Non-linear Block Gauss-Seidel algorithm with predictor and relaxation

```

while  $t < T$  do
   $n \leftarrow 0$ 
  Predictor:  $\tilde{\mathbf{u}}_0^{t+1} = \mathcal{P}_i(\mathbf{u}^t)$ 
  while  $n < n_{max}$  do
    Mesh update:  $\mathcal{M}(\tilde{\mathbf{u}}_n, \mathbf{z}_{n+1}) = 0$ 
    Fluid iteration:  $\mathcal{F}(\mathbf{w}_{n+1}, \mathbf{z}_{n+1}) = 0$ 
    Structural iteration:  $\mathcal{S}(\mathbf{u}_{n+1}, \mathbf{w}_{n+1}, \mathbf{z}_{n+1}) = 0$ 
    Relaxation:  $\tilde{\mathbf{u}}_{n+1} = (1 - \omega_n)\tilde{\mathbf{u}}_{n+1} + \omega_n \mathbf{u}_{n+1}$ 
    if  $\|\tilde{\mathbf{u}}_{n+1} - \tilde{\mathbf{u}}_n\| < \epsilon$  then
      Mesh update:  $\mathcal{M}(\mathbf{u}_{n+1}, \mathbf{z}_{n+1}) = 0$ 
       $\mathbf{u}^{t+1} \leftarrow \mathbf{u}_{n+1}$ 
       $\mathbf{w}^{t+1} \leftarrow \mathbf{w}_{n+1}$ 
       $\mathbf{z}^{t+1} \leftarrow \mathbf{z}_{n+1}$ 
      break
    else
       $n \leftarrow n + 1$ 
   $t \leftarrow t + 1$ 

```

Aitken's Δ^2 relaxation parameter [155]. If we define

$$\begin{aligned}\Delta \mathbf{u}_n &= \tilde{\mathbf{u}}_n - \mathbf{u}_{n-1} \quad \text{on } \Gamma, \\ \Delta \mathbf{u}_{n+1} &= \tilde{\mathbf{u}}_{n+1} - \mathbf{u}_n \quad \text{on } \Gamma,\end{aligned}\tag{2.57}$$

the Aitken's dynamic parameter is obtained as [65]

$$\omega_n = -\omega_{n-1} \frac{(\Delta \mathbf{u}_n)^T (\Delta \mathbf{u}_{n+1} - \Delta \mathbf{u}_n)}{|\Delta \mathbf{u}_{n+1} - \Delta \mathbf{u}_n|^2}.\tag{2.58}$$

As the information from iterations n and $n - 1$ is needed, it is suggested in [65, 155] to use the last relaxation parameter from the previous time step for the first iteration in a new time step, upper constrained with a ω_{max} , which is problem-dependent, to avoid using a value too large,

$$\omega_0^{t+1} = \max(\omega_n^t, \omega_{max}),\tag{2.59}$$

where ω_0^{t+1} is the initial relaxation parameter for the new time step and ω_n^t is the value of the relaxation parameter in the last iteration of the previous time step. This procedure is simple to implement and provides good results in terms of accuracy and robustness [62, 67]. Some other attempts have been made in order to obtain an improved solution for this relaxation parameter, as for example the *steepest descent* relaxation parameter (see [65]). It is based on obtaining the optimal step length in $\Delta \mathbf{u}_\Gamma^{n+1}$ direction. Nevertheless, this approach requires the computation of the interface Jacobian or its approximation as a matrix-vector product. The small improvements that may be achieved on convergence generally do not justify the extra computational cost of this procedure [65].

There have also been some efforts to improve convergence of the BGS method by means of improved predictors as, for example, the *coarse grid predictor* method developed by von Scheven and Ramm [62]. It is based on the idea of obtaining the solution of the refined problem having previously converged the coupled problem on a coarser grid, and using this solution as a predictor for the fine grid computations. This predictor has reported good results for problems with strong interactions.

Both Block Gauss-Seidel and Newton-Krylov methods have been used for Fluid-Structure Interaction problems with largely deformable structures. Even though the Newton-Krylov method is expected to have a lesser number of iterations per time step due to its higher order of convergence, the higher cost of each iteration results in comparable overall computational

times with the Block Gauss-Seidel method with Aitken's relaxation (see, e.g., [62, 65]). This fact, together with the ease of implementation of the latter, has led to the implementation of the Block Gauss-Seidel with Aitken's relaxation in this work.

2.3.2 Solver Implementation

The coupled FSI solver available within the source code of SU2 has been implemented as a part of this work. The solution of a multiphysics problem involving different solvers required the introduction of two new classes in SU2, namely *CDriver* and *CIteration*, as shown in in Fig. 2.8. The class *CIteration* accounts for the different stages of the solution process within one time step, and for a particular solver. In particular, it provides support for the pre- and post-processing of the solver, the iteration itself and the solution update once convergence has been reached.

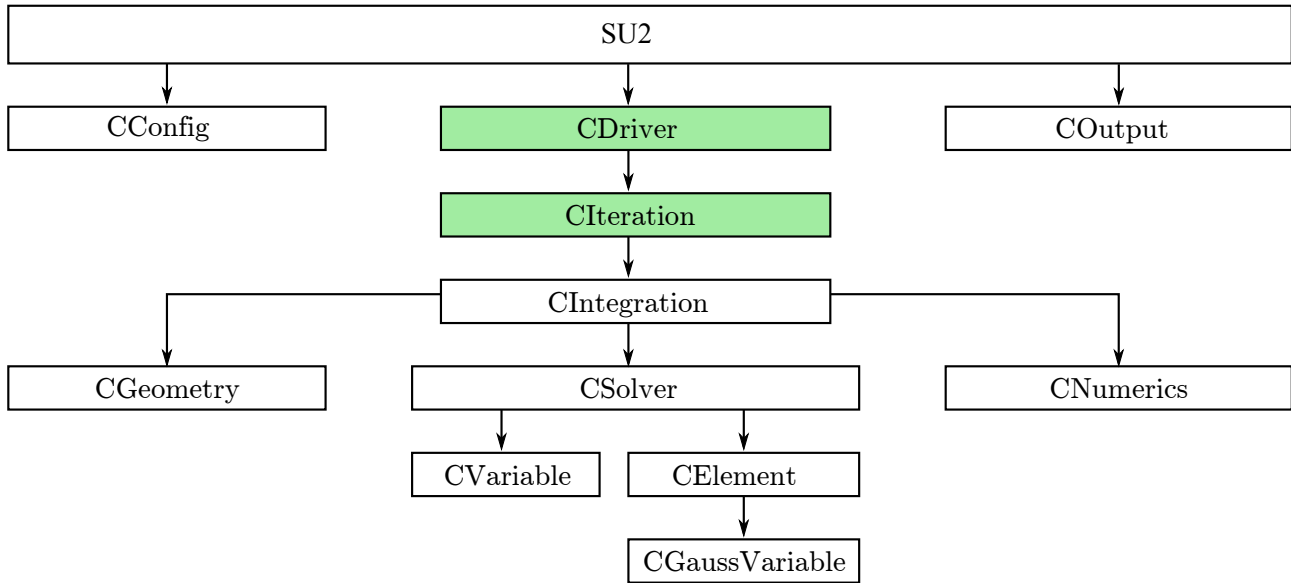


Figure 2.8: Introduction of the *CIteration* and *CDriver* classes.

The modularity of this approach permits to combine different solvers using the upper-level class *CDriver*. For a single zone fluid problem, for example, the child class *CFluidDriver* is in charge of performing the operations of *CIteration* in the appropriate order. For problems involving multiple physical domains, the *CDriver* allows to use the different parts of the code as if they were black-boxes, simply by calling the methods of *CIteration*. However, when the different solvers need to interact with each other, it is necessary to introduce mechanisms to communicate information at their interface. This is a complex task, especially when the solution

process is run in multiple processors.

This situation is illustrated in Fig. 2.9 for an example in which a solid body is immersed in a flow domain, run in two processors. A multi-core, multi-physics problem run on a distributed memory computer requires the MPI communication of information to be done not only within each independent solver, but also across the different solvers. Moreover, an optimal partition of the discretisation of each zone of the physical domain, which reduces to a minimum the number of edges cut by the partitions, may result in non-compliant regions across zones. This fact forces the transfer mechanisms to be able to efficiently distribute the information from processor i in the fluid domain to processor j in the structural domain. This issue has been resolved by developing a third class, called *CTransfer*, with two levels of inheritance which abstract the physics of the information to be transferred from the MPI transfer routines. This structure allows for different and specialized teams to develop and improve each particular part of the problem independently. More specifically, and for Fluid-Structure Interaction applications, the resulting code structure is shown in Fig. 2.10.

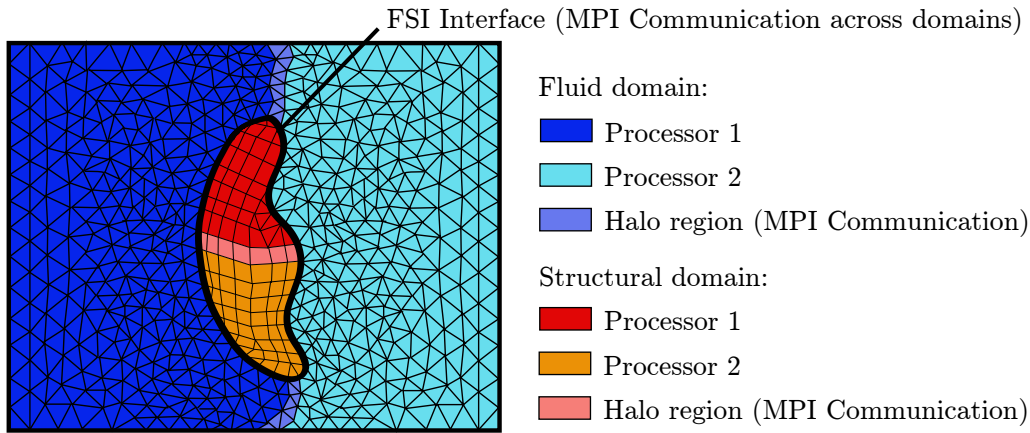


Figure 2.9: Sample FSI problem run in multiple cores.

The self-contained FSI solver implemented within SU2 includes a Block Gauss-Seidel (BGS), partitioned, strongly-coupled, implicit solution method in order to advance the fluid and structural solvers in time, as described in section 2.3.1 and Refs. [43, 61, 62, 65, 67], among others. This solver takes advantage of the modular implementation of SU2, in particular of the *CDriver* and *CIteration* methods which allow to use each solver as a black-box, together with the transfer methods previously described. The code subiterates within each time step as explained in Algorithm 3, until the convergence criteria that has been established for the interface displacements is met.

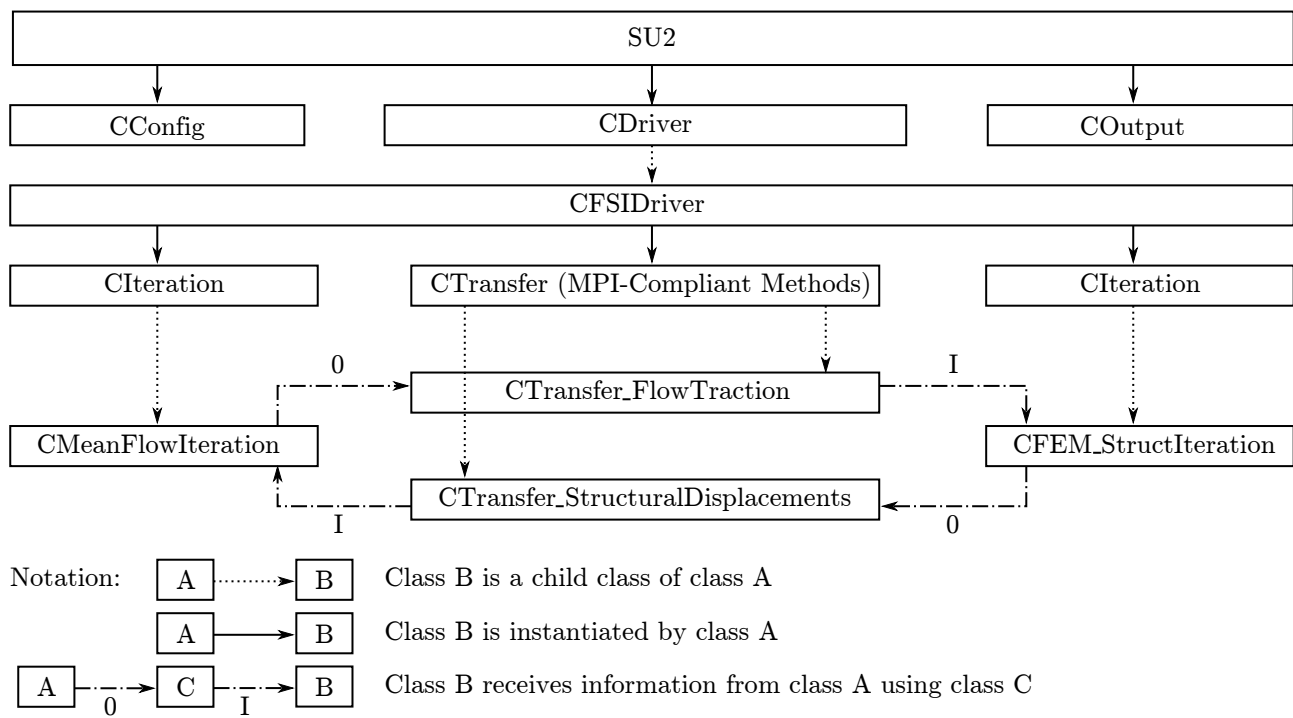


Figure 2.10: Transfer structure for Fluid-Structure Interaction problems.

Chapter 3

Gradient-based Optimisation Using Coupled Adjoints and AD

There are several methods available in the literature to compute the gradient of a generic objective function J with respect to a set of design variables $\boldsymbol{\alpha}$, as it was discussed in section 1.2.2. In section 3.1, the most common methods are covered and the advantages of the adjoint method for problems involving a large number of design variables are highlighted. Then, the principles of Algorithmic Differentiation (AD), and their applicability in the computation of gradients are presented in section 3.2.

Section 3.3 discusses the applicability and the main challenges of the adjoint method in costly CFD optimal shape design problems. An alternative approach to computing the adjoints based on fix-point iterators and AD is described, which simplifies the requirements of the adjoint method and specifically relaxes the need for an analytic derivation of the Jacobian of the fluid problem. This method was originally proposed by Korivi *et al.* [156], and has been implemented in SU2 by the group of Professor Nicolas Gauger at TU Kaiserslautern [120,121,123,124]. Their approach has been summarised in sections 3.3.1 and 3.3.2, and their implementation in section 3.3.3.

It is shown in section 3.4 that this AD-based adjoint method is also amenable to structural problems reformulated as fix-point iterators, both in static and dynamic conditions. This approach provides some advantageous features particularly for very non-linear problems, where the exact analytic linearisation of the structural equations becomes complex. These features were incorporated to the structural solver in SU2 as a part of this project, which is described in section 2.2.6.

Finally, section 3.5 discusses the extension of the adjoint method to coupled Fluid-Structure Interaction problems. For the case of a 3-field partitioned solver as described in section 2.3, conventional strategies require not only the linearisation of the fluid, fluid mesh and structural problems, but also the analytic linearisation across disciplines, a task which complicates the applicability of this techniques in most realistic problems. Alternatively, a novel approach for coupled FSI problems is presented in this thesis (section 3.5.1), which consistently applies the principles of the adjoint method based on fix-point iterators and AD described in sections 3.3 and 3.4 to the coupled, partitioned FSI solver. At this stage, the implementation has been limited to steady-state coupled problems, due to time constraints. However, the methodology is also outlined for time-dependent problems, as no major limitation prevents its implementation.

3.1 Conventional Methods to Compute Gradients

In this section, the most common strategies to compute gradients based on the analytical definition of derivative are briefly reviewed. Their advantages and disadvantages for their use in gradient-based optimisation are discussed based on the number of design variables involved in the optimisation process. For further reference, the reader may refer to Arora's book [157].

3.1.1 Finite differences

Assume J is a continuous and differentiable function. We can define its derivative with respect to any single design variable α_k using the well-known definition

$$\frac{dJ}{d\alpha_k} = \lim_{h \rightarrow 0} \frac{J(\alpha_k + h) - J(\alpha_k)}{h}, \quad (3.1)$$

where for convenience the implicit dependency of J in the state variables of the problem, $\mathbf{x}$, has been omitted and the notation simplified, $J(\alpha_k + h) = J(\mathbf{x}(\alpha_k + h), \alpha_k + h)$.

Based on this definition, it is possible to compute the gradient of the objective function with respect to any particular design variables using a first order finite-difference forward scheme,

$$\frac{dJ}{d\alpha_k} = \frac{J(\alpha_k + h_k) - J(\alpha_k)}{h_k} + \mathcal{O}(h), \quad \forall k \in n_\alpha, \quad (3.2)$$

where h_k is a finite increment and n_α the number of design variables. Alternatively, it is possible to use a more accurate second-order central difference,

$$\frac{dJ}{d\alpha_k} = \frac{J(\alpha_k + h_k) - J(\alpha_k - h_k)}{2h_k} + \mathcal{O}(h^2), \quad \forall k \in n_\alpha, \quad (3.3)$$

among many other finite-difference schemes available.

Due to their simplicity, these methods remain extremely popular in engineering applications for the calculation of gradients. This is particularly true for those cases in which the solution method is obtained using a black-box solver, as they don't require any change or insight into the source code. Their implementation is relatively straightforward, as they consist in the evaluation of the objective functions with a positive and/or a negative increment applied and a simple algebraic operation.

The gradient computed using finite differences will normally be reasonably accurate and acceptable for optimisation purposes as long as the increment h_k is adequately chosen, because these methods suffer from both cancellation and truncation errors. The former prevents from choosing a value of the increment too small, due to the subtraction in the numerator in (3.2) or (3.3). On the other hand, truncation errors limit the values of the increment in the upper side. As a result, the accuracy of these methods are extremely dependent on the value of the increment, which normally requires a parametric study to prevent the introduction of numerical noise into the gradient-based optimisation.

Another major drawback of finite difference methods relates to the efficiency of the computation. It may be observed that the expressions (3.2) and (3.3) are written for a scalar value of α_k . This is because the finite-difference derivative must be calculated independently *per design variable*. As a result, first order schemes require $n_\alpha + 1$ evaluations of the objective function, while second order require $2n_\alpha + 1$. Moreover, a valid increment h_k for a certain design variable α_k may suffer from cancellation or truncation errors for any other design variable α_j , $j, k \in n_\alpha$. Therefore, a different parametric study may be needed for each design variable.

3.1.2 Complex step method

The complex step method [41] is a highly accurate alternative to finite difference methods. Its main innovation is that it translates the calculation of the incremented function into the

complex domain, resulting

$$\frac{dJ}{d\alpha_k} = \frac{\text{Im}[J(\alpha_k + ih_k)]}{h} + \mathcal{O}(h^2), \quad \forall k \in n_\alpha. \quad (3.4)$$

This definition effectively removes the cancellation errors associated to finite differences and, therefore, the increment h may be as small as required. As a result, the accuracy of the method is high. On the other hand, the computational cost directly depends on the number of design variables, as the objective function has to be evaluated at least $n_\alpha + 1$ times. Moreover, the applicability of this method is generally limited to open-source codes or in-house platforms. This is because the problem variables need to be redefined in the complex domain, which requires access to the source code.

3.1.3 Direct method

The direct method is based on the application of the chain rule to the function $J(\mathbf{x}(\boldsymbol{\alpha}), \boldsymbol{\alpha})$. The gradient is written as a row vector,

$$\frac{dJ}{d\boldsymbol{\alpha}} = \frac{\partial J}{\partial \boldsymbol{\alpha}} + \frac{\partial J}{\partial \mathbf{x}} \frac{d\mathbf{x}}{d\boldsymbol{\alpha}}. \quad (3.5)$$

The conditions imposed to (1.3) require the fulfillment of the governing equations (1.1) for any given value of the design variable $\boldsymbol{\alpha}$. This implies that their total derivative with respect to the design variables is equal to zero, that is,

$$\frac{d\mathcal{G}}{d\boldsymbol{\alpha}} = \frac{\partial \mathcal{G}}{\partial \boldsymbol{\alpha}} + \frac{\partial \mathcal{G}}{\partial \mathbf{x}} \frac{d\mathbf{x}}{d\boldsymbol{\alpha}} = \mathbf{0} \quad (3.6)$$

within the design space. Let $\mathbf{K}_{\mathcal{G}}$ be the Jacobian of the governing equations, which may be written as $\mathbf{K}_{\mathcal{G}} = \partial \mathcal{G} / \partial \mathbf{x}$. We can rewrite (3.6) as

$$\mathbf{K}_{\mathcal{G}} \frac{d\mathbf{x}}{d\boldsymbol{\alpha}} = -\frac{\partial \mathcal{G}}{\partial \boldsymbol{\alpha}}, \quad (3.7)$$

which yields an expression that allows to compute $d\mathbf{x}/d\boldsymbol{\alpha}$ by solving one linear system of equations per design variable, that is, n_α linear systems. The cost of solving each linear system is proportional to the size of the physical problem, n_x , as the matrix of the system coincides

with the Jacobian of the problem $\mathbf{K}_{\mathcal{G}}$. Once obtained the value of $d\mathbf{x}/d\boldsymbol{\alpha}$, it is substituted into (3.6) for the calculation of the gradient. This is normally called the *direct* method [42, 158].

This method requires the exact analytic Jacobian of the problem to be available, as it may be noted from Eq. (3.7). For the solution of complex physical problems where the linearization of the equations is convoluted, sometimes only an approximated Jacobian is available, as it is generally enough for the convergence of implicit quasi-Newton schemes. This is a major drawback in problems involving complex physics, and their implications are discussed later in this chapter.

3.1.4 Adjoint method

Let $\tilde{\mathbf{x}}$ be defined as the adjoint, or dual, variables of the problem (1.1). We can use $\tilde{\mathbf{x}}$ to rewrite problem (1.3) in the form of an unconstrained optimization problem, by defining the problem Lagrangian,

$$\mathcal{L} = J(\mathbf{x}, \boldsymbol{\alpha}) + \tilde{\mathbf{x}}^T \mathcal{G}(\mathbf{x}, \boldsymbol{\alpha}). \quad (3.8)$$

From the imposition of the Karush-Kuhn-Tucker (KKT) conditions, see e.g. [157, 159], one can recover the state equations of the system,

$$\frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{x}}^T} = \mathcal{G}(\mathbf{x}, \boldsymbol{\alpha}) = \mathbf{0}, \quad (3.9)$$

the optimality condition, which enforces that at a minimum

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\alpha}} = \frac{\partial J(\mathbf{x}, \boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} + \tilde{\mathbf{x}}^T \frac{\partial \mathcal{G}(\mathbf{x}, \boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} = \mathbf{0}, \quad (3.10)$$

and, finally, the adjoint equation

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = \frac{\partial J(\mathbf{x}, \boldsymbol{\alpha})}{\partial \mathbf{x}} + \tilde{\mathbf{x}}^T \frac{\partial \mathcal{G}(\mathbf{x}, \boldsymbol{\alpha})}{\partial \mathbf{x}} = \mathbf{0}. \quad (3.11)$$

Using the definition of the problem Jacobian, the adjoint equation may be rewritten as

$$\mathbf{K}_{\mathcal{G}}^T \tilde{\mathbf{x}} = -\frac{\partial J^T}{\partial \mathbf{x}}. \quad (3.12)$$

Finally, substituting the value of $(\partial J / \partial \mathbf{x})^T$ into (3.5), and using (3.7), we can write

$$\frac{dJ}{d\boldsymbol{\alpha}} = \frac{\partial J}{\partial \boldsymbol{\alpha}} + \tilde{\mathbf{x}}^T \frac{\partial \mathcal{G}}{\partial \boldsymbol{\alpha}}. \quad (3.13)$$

From (3.12), it becomes apparent that one single linear system of the size of the physical problem of interest, n_x , needs to be solved to compute the adjoint variables. Once obtained $\tilde{\mathbf{x}}$, its value may be directly substituted into (3.13) to compute the gradient. Consequently, the cost of computing the gradients is no longer dependent on the number of design variables, except for the almost negligible computational cost of the matrix-vector product in (3.13). This approach is extremely powerful for gradient-based optimisation problems in which the number of design variables is large, as discussed later in this work. However, the adjoint method as defined in this section also requires the exact analytic Jacobian to be available for the solution of (3.12), which normally complicates its implementation in problems with complex physics.

3.2 Gradient Computation using Algorithmic Differentiation

As an alternative to the methods described in section 3.1, it is also possible to employ Algorithmic Differentiation (AD) techniques [114, 115] in the computation of gradients. The main idea behind AD links to the fact that any computational code is built as a sequence of l elementary functions φ_i , $i < l$. These functions may be simple operations, as for example addition, multiplication, exponential functions, etc. [114].

This method is best explained with an example. Following the description of Albring *et al.* [120], let $y = J(\alpha_1)$ be the value of the objective function with one design variables, and let v_i be a set of intermediate values in the computational sequence. The evaluation of J may be written as in table 3.1.

v_0	$= \alpha_1$	
v_i	$= \varphi_i(v_j)_{j < i}$	$i = 1, \dots, l$
y_1	$= v_l$	

Table 3.1: Computational evaluation of the function $y = J(\alpha_1)$.

There are two main possibilities to exploit this property. On the one hand, it is possible to use

the *forward* mode of AD, which propagates the derivative of a chosen input in the background throughout the whole computational sequence, allowing to obtain the exact gradient (accurate to machine level). The main limitation is that the l operations in Tab. 3.1 need to be run once per variable with respect to which we aim to differentiate. Alternatively, the *reverse* mode allows to compute gradients with respect to any number of inputs, by running the computational sequence backwards. Both methods are further described in the following sections.

3.2.1 Forward mode

The application of the chain rule to the elementary functions φ_i in table 3.1 leads to the definition of the forward mode of AD. This technique propagates the tangents of the variable α_1 in the computational code. The evaluation of each intermediate function φ_i also involves the propagation of its the derivative [115, 120], as shown in table 3.2.

Table 3.2: Forward mode of AD.

Function Evaluation		Derivative Evaluation	
v_0	$= \alpha_1$	$\dot{v}_0$	$= \dot{\alpha}_1$
v_i	$= \varphi_i(v_j)_{j < i}$	$\dot{v}_i$	$= \sum_{j < i} \frac{\partial \varphi_i(v_j)}{\partial v_j} \dot{v}_j \quad i = 1, \dots, l$
y	$= v_l$	$\dot{y}$	$= \dot{v}_l$

The value of the design variable derivative, $\dot{\alpha}_1$ in Table 3.2, is initialized to 1.0, as corresponds to $d\alpha_1/d\alpha_1$. Finally, using the chain rule, $dJ/d\alpha_1 = \dot{v}_l$. For this case, therefore, the objective function needs to be evaluated once per design variable, that is, n_α times.

This method is conceptually linked to the complex-step method as noted by Martins *et al* [41], although the magnitude that is being propagated through the code for the forward mode is the value of the derivative itself. The forward mode requires the overloading of the operators in order to handle the derivative storage and evaluation. An overloaded *double* structure needs to be able to store two, instead of one, floats. The operators need to handle the derivative operations apart from their own evaluation. An alternative is to use tools that transform the source code of the solver. This is normally more efficient in terms of memory requirements than operator overloading [115]. An example of such tool is Tapenade [160]. In both cases, access to the source code of the solver is necessary.

3.2.2 Reverse mode

In the context of AD, the sensitivity problem may also be redefined for an increased efficiency in the case of a large number of design variables. This is achieved using the so-called *reverse* mode. The chain rule can be applied backwards in the sequence of elementary functions that constitute the numerical solution. Following the description of Griewank *et al.* [115], let $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$ be the adjoint variables corresponding to the input $\mathbf{x}$ and output $\mathbf{y}$ of the primal evaluation of the function $\mathbf{y} = \mathbf{f}(\mathbf{x})$. An equivalent relation to the adjoint equation in (3.12) may be written as

$$\bar{\mathbf{x}} = \left(\frac{d\mathbf{f}}{d\mathbf{x}} \right)^T \bar{\mathbf{y}}, \quad (3.14)$$

which implies the relation $\bar{\mathbf{y}}^T \dot{\mathbf{y}} = \bar{\mathbf{x}}^T \dot{\mathbf{x}}$ between the primal and dual variables. The transpose in Eq. (3.14) implies a reverse sequence of operations starting with $\bar{\mathbf{y}}$ [97], which results in the numerical procedure of Tab. 3.3.

Table 3.3: Reverse mode of AD with one input and one output.

v_0	$= x_1$	
v_i	$= \varphi_i(v_j)_{j < i}$	$i = 1, \dots, l$
y_1	$= v_l$	
$\bar{v}_l$	$= \bar{y}_1$	
$\bar{v}_j$	$= \bar{v}_j + \bar{v}_i \frac{\partial \varphi_i(v_j)}{\partial v_j}, i > j$	$i = l, \dots, 1$
$\bar{x}_1$	$= \bar{v}_0$	

Given that the elementary functions $\varphi_i(v_j)$ are evaluated backwards or in *reverse*, the intermediate variables v_j must have been previously computed. Only one additional computation, run backwards through the adjoint path, allows for the calculation of as many rows of $\frac{d\mathbf{f}}{d\mathbf{x}}$ as needed, according to Griewank [115]. This corresponds to the second block in Table 3.3, and means that the gradients with respect to any number of input variables can be obtained performing a single run, which has a cost between one to four times that required for the primal problem [115]. As a drawback, the reverse mode needs to be rerun as many times as the number of outputs we are interested in.

The main challenge for an efficient implementation of the reverse mode of AD is the program reversal. One option is to leverage in the advanced capabilities of some programming languages, such as operator overloading in C++. Alternatively, source-code transformation is also possible

and normally preferred, as explained by Mader *et al.* [117]. Still, the main drawbacks relate to the memory footprint, which is largely increased in both cases due to the need to store intermediate results. Moreover, the use of source code transformation requires a very deep knowledge of the solution process and can limit its applicability to realistic problems. This is because they often require large, sophisticated software packages that are difficult to transform. Source code transformation also limits the incorporation of new methods to the solver, as constant updates of the code would be required to account for their differentiation.

Expression templates

As an alternative to source code transformation and regular methods for operator overloading, SU2 implements the reverse mode using Expression Templates (ET), a methodology first proposed by Veldhuizen [161]. ET may be defined as a meta-programming technique that creates a compile-time parse tree of the overloaded expressions in the code. This method has been successfully applied before in the efficient solution of complex PDE systems using optimised code [162, 163].

The main advantage of ET for the application of the reverse mode is that it allows for the computational graph of any expression to be efficiently recorded and run backwards using an AD tool. Hogan [122] recently described an implementation of this methodology for reverse adjoint applications in C++ libraries. A similar philosophy has been followed in order to link the AD tool CoDiPack with SU2, as discussed by Albring *et al.* [120, 121]. An example for the computational path of a simple expression, $y = x_2 \cos(x_1)$, is presented in Fig. 3.1. Instead of returning the relatively expensive result of the expression, the operators of a code implementing ET return a small, temporary object [120, 122]. These objects have been highlighted in Fig. 3.1 for each expression over the outgoing arrow. The object *su2double* in the figure corresponds to the special type that has been defined in SU2 for doubles.

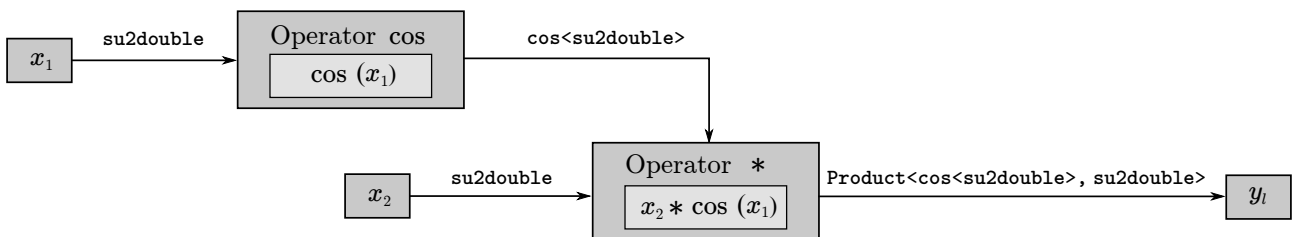


Figure 3.1: Computational graph for the expression $y = v_0 \cos(v_{-1})$. [141], adapted from [120, 122]

The process of the backwards run of the computational graph in Fig. 3.1 is presented in

Fig. 3.2. The efficiency of the reverse mode is improved by a dramatic increase in the speed of the backwards run, as shown by Hogan [122]. At the same time, the memory usage is reduced with respect to regular operator-overloading libraries. These advantages make of ET a very competitive alternative for the application of the reverse mode of AD in complex problems. Moreover, the extension for new developments is almost automatic for any solver implemented in a framework that includes ET, a feature that has been tested and exploited in this thesis.

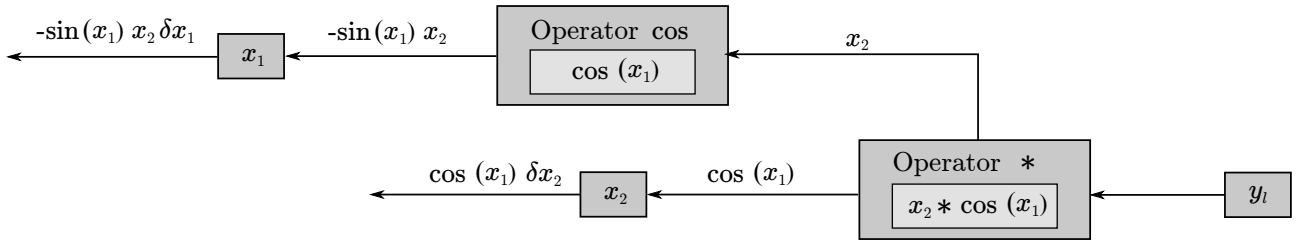


Figure 3.2: Propagation of gradients in reverse mode for the expression $y = v_0 \cos(v_{-1})$. [141], adapted from [120, 122]

3.3 The Adjoint Fluid Solver in SU2

The earlier works of Pirennau in the 1970s [164] are normally regarded to be the first use of adjoint methods in fluid dynamics. However, this approach really gained momentum in aerodynamic shape design when Jameson published his paper in 1988 [165]. He applied the adjoint method and optimal control theory to the flow equations, with the objective of reducing the cost of computing gradients in costly CFD simulations for a large number of design variables. Ever since then, the adjoint method has been widely used in gradient-based optimal design applied to fluid dynamics problems, for example in aerodynamic shape design [166–169] or turbomachinery applications [170, 171], just to name a few.

We will now illustrate the advantages of the adjoint method for optimal shape design. Then, in section 3.3.1, we will describe the efficient methodology adopted in SU2 [120, 121] for fluid discrete adjoints, which is the starting point that will lead to our coupled FSI adjoint. The applicability of this method to time-domain aeroacoustics problems [123, 124] is briefly outlined in section 3.3.2. Finally, the implementation of this methodology which was available in SU2 is covered in section 3.3.3

Let us now present a simple, conceptual example, in which the objective function J is the drag coefficient, $J = C_D$, for an airfoil such as the one shown in Fig. 3.3. This airfoil surface

has been discretised using $n_p = 100$ points in its surface, for which each point is associated to a set of 2D coordinates, $\mathbf{Z}^i = (Z_1^i, Z_2^i)$. Let now the optimisation problem be defined as

$$\begin{aligned} \min_{\boldsymbol{\alpha}} J(\mathbf{w}(\boldsymbol{\alpha}), \boldsymbol{\alpha}) \\ \text{subject to } \mathcal{F}(\mathbf{w}(\boldsymbol{\alpha}), \boldsymbol{\alpha}) = 0, \end{aligned} \quad (3.15)$$

where $\mathbf{w}$ are the flow conservative variables, $\mathcal{F}$ the flow equations and $\boldsymbol{\alpha}$ a set of design variables, which corresponds to the coordinates of the surface discretised in n_p points, $\boldsymbol{\alpha} = [\mathbf{Z}^1, \dots, \mathbf{Z}^{n_p}]$.

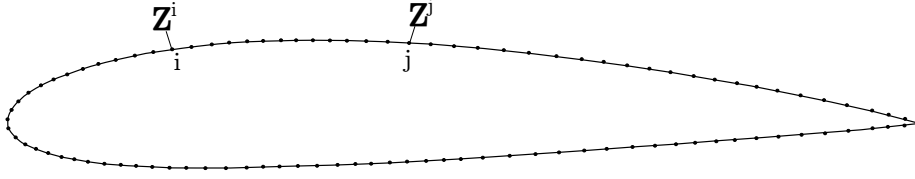


Figure 3.3: Sample airfoil NACA 2412 discretised with 100 points in the surface.

From (3.12), we can define the adjoint of the flow conservative variables, $\tilde{\mathbf{w}}$,

$$\mathbf{K}_{\mathcal{F}}^T \tilde{\mathbf{w}} = - \left[\frac{\partial J}{\partial \mathbf{w}} \right]^T. \quad (3.16)$$

Once solved (3.16), we can compute the gradient using

$$\frac{dJ}{d\boldsymbol{\alpha}} = \frac{\partial J}{\partial \boldsymbol{\alpha}} + \tilde{\mathbf{w}}^T \frac{\partial \mathcal{F}}{\partial \boldsymbol{\alpha}}. \quad (3.17)$$

In a gradient-based optimisation setting, one iteration of the optimiser would normally require both the evaluation of the objective function J , that is, the solution of the primal problem, and also the evaluation of the gradient with respect to all the design variables, $dJ/d\boldsymbol{\alpha}$. For this particular example, we compare in Tab. 3.4 the number of evaluations of the solution process required for one iteration of the gradient-based optimiser. For comparison purposes, we assume the cost of evaluating the linear system in (3.5) is similar or equivalent to the cost of the primal solution, given that the size of the matrix of the problem is in both cases $n_{\mathbf{w}} \times n_{\mathbf{w}}$. A similar assumption may be adopted regarding the cost of computing the adjoint solution. The complex step has a higher computational cost per evaluation due to the overload of the solution process in order to accomodate complex numbers.

From this simple example in Tab. 3.4, it becomes clear that using an adjoint method is highly advantageous for gradient-based optimisation in problems with a large number of design variables. Given that the computational cost of the adjoint remains independent of the number of

Table 3.4: Number of evaluations per iteration of a problem with $n_\alpha = 200$ in an optimization framework. *The complex step method requires operator overloading, which increases the cost per evaluation. **The direct method does not evaluate the primal solver, but a linear system with the same number of variables.

	Evaluation of J	Evaluation of $dJ/d\alpha$		Total Evaluations
	Primal CFD	Primal CFD	Adjoint CFD	
Forward Differences (3.2)	1	200	-	201
Central Differences (3.3)	1	400	-	401
Complex Step (3.4)	1*	200*	-	201*
Direct Method (3.5)	1	200**	-	201
Adjoint Method (3.16)	1	-	1	2

design parameters, the number of costly evaluations of the fluid primal problem is dramatically reduced.

However, one major disadvantage implicit to the formulation of the method requires further comment. The computation of the adjoint variables requires the solution of the linear system in Eqn. (3.16). This equation requires the Jacobian of the fluid problem, $\mathbf{K}_{\mathcal{F}}$, to be explicitly available. In large problems with fine levels of discretisation and/or involving complex phenomena, this is normally not feasible, as the solver would incur in a massive memory overhead in order to pre-compute and store this matrix [172]. Moreover, this Jacobian is very sparse and normally the matrix-vector product to compute the residual occurs at a local level. According to Giles and Pierce, the process to rework this local product and transpose it for adjoint applications is “relatively easy for the inviscid fluxes of the Euler equations [...] (but) far more arduous for the viscous fluxes in the Navier-Stokes equations” [172]. The complexity of transposing the analytic Jacobian, even if it is available, is also noted by Barcelos and Maute [43], particularly for parallel architectures. Albring *et al.* [121] point out that the several approximations are often adopted in the computation of the Jacobian in CFD codes, which are not valid for the calculation of the adjoint as it requires the use of the exact matrix $\mathbf{K}_{\mathcal{F}}$. Turbulence model equations, for example, are normally assumed constant for adjoint applications, due to the complexity of hand-differentiating the turbulent terms [173]. All these limitations and assumptions lead normally to approximate adjoint methods, where the effect of the approximations requires further assessment [174].

In order to overcome these complications, SU2 implements an alternative approach for computing the flow adjoint variables. It is based on the application of the reverse mode of Algorithmic Differentiation (AD) to the fluid problem, which is rewritten as a fixed-point iterator. This allows for the computational path of the solver to be recorded using ET, and then run

backwards to iteratively compute the adjoint. It removes the need for the exact analytical differentiation of the flow equations, allowing the use of any flow solver which implements these techniques to compute the adjoint. The only condition is that the primal solver converges to the solution. Full credit for this methodology and its implementation in SU2 for fluid and aeroacoustics problems is given to Professor Gauger's group at the Chair for Scientific Computing of TU Kaiserslautern. A brief summary of the works of Albring *et al.* [120, 121] and Zhou *et al.* [123, 124] describing this method is provided in the sections 3.3.1 and 3.3.2, and their implementation in SU2 is described in section 3.3.3. This method provides the base for the structural and coupled FSI adjoint methods that are later presented in sections 3.4 and 3.5, and that have been developed and implemented in SU2 during this work.

3.3.1 Adjoint Methods Based on Fixed-Point Iterators and AD

This section presents a brief description of a methodology to the calculation of the adjoint problem based on fixed-point iterators. This method has been developed by Albring *et al.* [120, 121] based on previous works by Korivi *et al.* [156], and it is available in SU2 for optimal shape design problems. From Eqn. 2.6, let the fluid residual equation be written as

$$\mathcal{F}(\mathbf{w}, \mathbf{z}) = \partial_t \mathbf{w} + \nabla \cdot \mathbf{F}^c(\mathbf{w}, \mathbf{z}) - \nabla \cdot \mathbf{F}^v(\mathbf{w}, \mathbf{z}) = \mathbf{0} \quad \text{in } \Omega_f. \quad (3.18)$$

where the dependency on the deformation of the computational mesh $\mathbf{z}$ has been included. This equation may be rewritten in the form of a fixed-point iteration in terms of the state variables of the fluid problem, $\mathbf{w}$, by defining the fix-point operator $\mathbf{F}$,

$$\mathbf{w}^{n+1} = \mathbf{F}(\mathbf{w}^n, \mathbf{z}), \quad (3.19)$$

where $\mathbf{F}$ is only feasible at the solution of the fluid problem, $\mathbf{w}^*$, i.e.,

$$\mathcal{F}(\mathbf{w}^*, \mathbf{z}) = \mathbf{0} \Leftrightarrow \mathbf{w}^* = \mathbf{F}(\mathbf{w}^*, \mathbf{z}). \quad (3.20)$$

The problem defined by (3.20) converges if $\mathbf{F}(\mathbf{w}^n, \boldsymbol{\alpha})$ is contractive, according to the Banach

fixed-point theorem. This means that, in a suitable matrix norm [121],

$$\left\| \frac{\partial \mathbf{F}}{\partial \mathbf{w}} \right\| < 1. \quad (3.21)$$

From (2.5), the mesh problem is solved as

$$\mathcal{M}(\mathbf{z}, \boldsymbol{\alpha}) = \tilde{\mathbf{K}}_{\mathcal{M}} \mathbf{z} - \tilde{\mathbf{f}}(\boldsymbol{\alpha}) = \mathbf{0}, \quad (3.22)$$

where the design variables $\boldsymbol{\alpha} = [\mathbf{Z}^1, \dots, \mathbf{Z}^{n_p}]$ are the positions of the discretised surface of the airfoil in Fig. 3.3. The previous problem can also be rewritten in the form of a fix-point operator,

$$\mathbf{z} = [\tilde{\mathbf{K}}^m]^{-1} \tilde{\mathbf{f}}(\boldsymbol{\alpha}) := \mathbf{M}(\boldsymbol{\alpha}), \quad (3.23)$$

where the dependency of $\mathbf{z}$ on $\boldsymbol{\alpha}$ becomes explicit given the linear character of (3.22). Using (3.19) and (3.23), (3.15) may be rewritten as

$$\begin{aligned} & \min_{\boldsymbol{\alpha}} J(\mathbf{w}, \mathbf{z}) \\ & \text{subject to } \mathbf{F}(\mathbf{w}, \mathbf{z}) - \mathbf{w} = \mathbf{0}, \\ & \mathbf{M}(\boldsymbol{\alpha}) - \mathbf{z} = \mathbf{0}, \end{aligned} \quad (3.24)$$

where the explicit dependency of $J = C_D$ in $\boldsymbol{\alpha}$ has been dropped as it is implicit to $\mathbf{w}(\boldsymbol{\alpha})$ and $\mathbf{z}(\boldsymbol{\alpha})$.

Following the procedure presented in section 3.1.4, the problem in (3.15) may be solved minimising the problem Lagrangian instead,

$$\min J(\mathbf{w}, \mathbf{z}) + \bar{\mathbf{w}}^T [\mathbf{F}(\mathbf{w}, \mathbf{z}) - \mathbf{w}] + \bar{\mathbf{z}}^T [\mathbf{M}(\boldsymbol{\alpha}) - \mathbf{z}]. \quad (3.25)$$

Differentiating (3.25) with respect to the flow conservative variables $\mathbf{w}$, we obtain an implicit equation for the flow adjoint,

$$\bar{\mathbf{w}}^T = \frac{\partial J(\mathbf{w}, \mathbf{z})}{\partial \mathbf{w}} + \bar{\mathbf{w}}^T \frac{\partial \mathbf{F}(\mathbf{w}, \mathbf{z}, \boldsymbol{\alpha})}{\partial \mathbf{w}}, \quad (3.26)$$

from which it is possible to define a fixed-point iterator in $\bar{\mathbf{w}}$,

$$\bar{\mathbf{w}}^{n+1} = \left[\frac{\partial J(\mathbf{w}^*, \mathbf{z}^*)}{\partial \mathbf{w}} \right]^T + \left[\frac{\partial \mathbf{F}(\mathbf{w}^*, \mathbf{z}^*)}{\partial \mathbf{w}} \right]^T \bar{\mathbf{w}}^n := \bar{\mathbf{F}}(\bar{\mathbf{w}}^n), \quad (3.27)$$

where the derivatives of J and $\mathbf{F}$ are evaluated around the feasible (converged) solution of the flow problem, $\mathbf{w}^*$, and the corresponding mesh $\mathbf{z}^*$. The linear equation in (3.26) cannot be solved directly as there is not an explicit expression available for $\partial \mathbf{F} / \partial \mathbf{w}$. Instead, Algorithmic Differentiation is used to compute the matrix-vector product in (3.27). The operator $\bar{\mathbf{F}}$ corresponds to the reverse path of the fluid solver, which is recorded using ET and AD and run in reverse using the AD tool CoDiPack from Professor Gauger's group. The recording process will be covered in section 3.3.3. It is shown by Albring *et al.* [120] that $\bar{\mathbf{F}}$ is contractive provided that $\mathbf{F}$ is contractive. This means that the adjoint fix-point iteration directly inherits the convergence properties of the primal solver, although there is no guarantee of the contractiveness of $\mathbf{F}$ even if the solver converges. This needs to be investigated numerically for each problem.

Differentiating now (3.25) with respect to the geometry variables $\mathbf{z}$, we obtain an explicit expression that depends on the flow adjoint, $\bar{\mathbf{w}}$,

$$\bar{\mathbf{z}}^T = \frac{\partial J(\mathbf{w}, \mathbf{z})}{\partial \mathbf{z}} + \bar{\mathbf{w}}^T \frac{\partial \mathbf{F}(\mathbf{w}, \mathbf{z})}{\partial \mathbf{z}}, \quad (3.28)$$

from where the mesh adjoint may be evaluated once a converged value of $\bar{\mathbf{w}}$ has been obtained. Finally, the derivative of the Lagrangian with respect to the design variables $\boldsymbol{\alpha}$ yields

$$\frac{d\mathcal{L}}{d\boldsymbol{\alpha}} = \frac{dJ}{d\boldsymbol{\alpha}} = \bar{\mathbf{z}}^T \frac{\partial \mathbf{M}(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}}, \quad (3.29)$$

which can be evaluated using the value of $\bar{\mathbf{z}}$ obtained from (3.28).

3.3.2 AD-based Adjoint for Time-Domain Simulations

The approach described in the previous section was extended by Zhou *et al.* [123, 124] to time-domain problems. A brief description of their methodology will be given here, as an analogous concept will be later applied to structural mechanics. Let the dual time-stepping equation in

(2.10) be rewritten as

$$\mathbf{R}^*(\mathbf{w}^n) = \frac{3}{2\Delta t}\mathbf{w}^n + \mathbf{R}(\mathbf{w}^n) - \frac{2}{\Delta t}\mathbf{w}^{n-1} + \frac{1}{2\Delta t}\mathbf{w}^{n-2}, \quad (3.30)$$

where the time dependency on the changes in the control volumes has been dropped, assuming that the mesh remains constant throughout the time-domain simulation. Introducing the pseudo-time τ as in (2.9) and using an implicit Euler method to march in time, it results

$$\mathbf{w}_{k+1}^n - \mathbf{w}_{k+1}^n + \Delta\tau \mathbf{R}^*(\mathbf{w}_k^n) = \mathbf{0}, \quad (3.31)$$

which after linearisation and some algebra [123] can be rewritten in the form of a fixed-point iteration

$$\mathbf{w}_{k+1}^n = \mathbf{F}(\mathbf{w}_k^n, \mathbf{w}^{n-1}, \mathbf{w}^{n-2}, \mathbf{z}), \quad (3.32)$$

where a generic dependency on $\mathbf{z}$ has been included for consistency with the previous section. The optimization problem results now

$$\begin{aligned} & \min_{\boldsymbol{\alpha}} \frac{1}{n} \sum_{k=1}^n \hat{J}(\mathbf{w}^k, \mathbf{z}) \\ & \text{subject to } \mathbf{F}(\mathbf{w}^n, \mathbf{w}^{n-1}, \mathbf{w}^{n-2}, \mathbf{z}) - \mathbf{w}^n = \mathbf{0}, \\ & \quad \mathbf{F}(\mathbf{w}^{n-1}, \mathbf{w}^{n-2}, \mathbf{w}^{n-3}, \mathbf{z}) - \mathbf{w}^{n-1} = \mathbf{0}, \\ & \quad \dots \\ & \quad \mathbf{F}(\mathbf{w}^3, \mathbf{w}^2, \mathbf{w}^1, \mathbf{z}) - \mathbf{w}^3 = \mathbf{0}, \\ & \quad \mathbf{F}(\mathbf{w}^2, \mathbf{w}^1, \mathbf{z}) - \mathbf{w}^2 = \mathbf{0}, \\ & \quad \mathbf{F}(\mathbf{w}^1, \mathbf{z}) - \mathbf{w}^1 = \mathbf{0}, \\ & \quad \mathbf{M}(\boldsymbol{\alpha}) - \mathbf{z} = \mathbf{0}, \end{aligned} \quad (3.33)$$

where the hat in $\hat{J}(\mathbf{w}^k, \mathbf{z})$ indicates that its value corresponds to the *instantaneous* value of the objective function for time instant k . The optimisation problem seeks now to minimise the average of J in the time window $[1, n]$. Eqn. 3.33 can be rewritten in the form of the Lagrangian

$$\mathcal{L} = \frac{1}{n} \sum_{k=1}^n \hat{J}(\mathbf{w}^k, \mathbf{z}) + \sum_{k=1}^n [\bar{\mathbf{w}}^k]^T [\mathbf{F}(\mathbf{w}^k, \mathbf{w}^{k-1}, \mathbf{w}^{k-2}, \mathbf{z}) - \mathbf{w}^k] + \bar{\mathbf{z}}^T [\mathbf{M}(\boldsymbol{\alpha}) - \mathbf{z}]. \quad (3.34)$$

As explained in section 3.1.4, differentiating $\mathcal{L}$ with respect to the state variables $\mathbf{w}^i$, $\forall i \in n$ leads to the adjoint equations. The following system of equations is obtained:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mathbf{w}^n} &= \frac{1}{n} \frac{\partial \hat{J}}{\partial \mathbf{w}^n}(\mathbf{w}^n, \mathbf{z}) + [\bar{\mathbf{w}}^n]^T \frac{\partial \mathbf{F}}{\partial \mathbf{w}^n}(\mathbf{w}^n, \mathbf{w}^{n-1}, \mathbf{w}^{n-2}, \mathbf{z}) - [\bar{\mathbf{w}}^n]^T &= \mathbf{0}, \\
\frac{\partial \mathcal{L}}{\partial \mathbf{w}^{n-1}} &= \frac{1}{n} \frac{\partial \hat{J}}{\partial \mathbf{w}^{n-1}}(\mathbf{w}^{n-1}, \mathbf{z}) + [\bar{\mathbf{w}}^n]^T \frac{\partial \mathbf{F}}{\partial \mathbf{w}^{n-1}}(\mathbf{w}^n, \mathbf{w}^{n-1}, \mathbf{w}^{n-2}, \mathbf{z}) \\
&\quad + [\bar{\mathbf{w}}^{n-1}]^T \frac{\partial \mathbf{F}}{\partial \mathbf{w}^{n-1}}(\mathbf{w}^{n-1}, \mathbf{w}^{n-2}, \mathbf{w}^{n-3}, \mathbf{z}) - [\bar{\mathbf{w}}^{n-1}]^T &= \mathbf{0}, \\
\frac{\partial \mathcal{L}}{\partial \mathbf{w}^{n-2}} &= \frac{1}{n} \frac{\partial \hat{J}}{\partial \mathbf{w}^{n-2}}(\mathbf{w}^{n-2}, \mathbf{z}) + [\bar{\mathbf{w}}^n]^T \frac{\partial \mathbf{F}}{\partial \mathbf{w}^{n-2}}(\mathbf{w}^n, \mathbf{w}^{n-1}, \mathbf{w}^{n-2}, \mathbf{z}) \\
&\quad + [\bar{\mathbf{w}}^{n-1}]^T \frac{\partial \mathbf{F}}{\partial \mathbf{w}^{n-2}}(\mathbf{w}^{n-1}, \mathbf{w}^{n-2}, \mathbf{w}^{n-3}, \mathbf{z}) \\
&\quad + [\bar{\mathbf{w}}^{n-2}]^T \frac{\partial \mathbf{F}}{\partial \mathbf{w}^{n-2}}(\mathbf{w}^{n-2}, \mathbf{w}^{n-3}, \mathbf{w}^{n-4}, \mathbf{z}) - [\bar{\mathbf{w}}^{n-2}]^T &= \mathbf{0}, \\
&\dots \\
\frac{\partial \mathcal{L}}{\partial \mathbf{w}^1} &= \frac{1}{n} \frac{\partial \hat{J}}{\partial \mathbf{w}^1}(\mathbf{w}^1, \mathbf{z}) + [\bar{\mathbf{w}}^3]^T \frac{\partial \mathbf{F}}{\partial \mathbf{w}^1}(\mathbf{w}^3, \mathbf{w}^2, \mathbf{w}^1, \mathbf{z}) \\
&\quad + [\bar{\mathbf{w}}^2]^T \frac{\partial \mathbf{F}}{\partial \mathbf{w}^1}(\mathbf{w}^2, \mathbf{w}^1, \mathbf{z}) \\
&\quad + [\bar{\mathbf{w}}^1]^T \frac{\partial \mathbf{F}}{\partial \mathbf{w}^1}(\mathbf{w}^1, \mathbf{z}) - [\bar{\mathbf{w}}^1]^T &= \mathbf{0},
\end{aligned} \tag{3.35}$$

which can be rewritten in a compact form,

$$\left\{ \begin{array}{l} \bar{\mathbf{w}}^n = \bar{\mathbf{F}}_n^n(\bar{\mathbf{w}}^n), \\ \bar{\mathbf{w}}^{n-1} = \bar{\mathbf{F}}_{n-1}^n(\bar{\mathbf{w}}^n) + \bar{\mathbf{F}}_{n-1}^{n-1}(\bar{\mathbf{w}}^{n-1}), \\ \bar{\mathbf{w}}^{n-2} = \bar{\mathbf{F}}_{n-2}^n(\bar{\mathbf{w}}^n) + \bar{\mathbf{F}}_{n-2}^{n-1}(\bar{\mathbf{w}}^{n-1}) + \bar{\mathbf{F}}_{n-2}^{n-2}(\bar{\mathbf{w}}^{n-2}), \\ \dots \\ \bar{\mathbf{w}}^1 = \bar{\mathbf{F}}_1^3(\bar{\mathbf{w}}^3) + \bar{\mathbf{F}}_1^2(\bar{\mathbf{w}}^2) + \bar{\mathbf{F}}_1^1(\bar{\mathbf{w}}^1), \end{array} \right. \tag{3.36}$$

where $\bar{\mathbf{F}}_{n-1}^n$ corresponds to the reverse path of the solver $\mathbf{F}$ evaluated at time n and differentiated with respect to $n-1$. For a generic time position t , the fix-point iteration becomes

$$\bar{\mathbf{w}}^t = \bar{\mathbf{F}}_t^t(\bar{\mathbf{w}}^t) + \bar{\mathbf{F}}_t^{t+1}(\bar{\mathbf{w}}^{t+1}) + \bar{\mathbf{F}}_t^{t+2}(\bar{\mathbf{w}}^{t+2}). \tag{3.37}$$

The evaluation of the different terms via AD will be described in section 3.3.3. From an examination of (3.36), it becomes apparent that it is necessary to obtain the solution for $\bar{\mathbf{w}}^n$ first, and then proceed backwards in time. Therefore, the primal solver needs to be advanced to the full time domain before the adjoint simulation can be run. Finally, once obtained the values of the adjoint for $\bar{\mathbf{w}}^i, \forall i \in n$, the mesh adjoint is evaluated as

$$\bar{\mathbf{z}}^T = \sum_{k=1}^n \left[\frac{1}{n} \frac{\partial \hat{J}(\mathbf{w}, \mathbf{z})}{\partial \mathbf{z}} + (\bar{\mathbf{w}}^n)^T \frac{\partial \mathbf{F}(\mathbf{w}^n, \mathbf{w}^{n-1}, \mathbf{w}^{n-2}, \mathbf{z})}{\partial \mathbf{z}} \right], \quad (3.38)$$

and, finally, the gradient can be obtained using (3.29).

3.3.3 Code Structure of the AD-Based Adjoint Method in SU2

The implementation of the AD-based adjoint flow solver for the iterative solution of Eqn. (3.27) was carried out at the Chair for Scientific Computing of TU Kaiserslautern for aerodynamic shape optimisation and aeroacoustics, and it is described in Refs. [120, 121, 123, 124]. A brief description of the code structure will be given here.

Once obtained the solution of the direct problem $\mathbf{w}^*$, the primal solver may be advanced for one further iteration, obtaining $\mathbf{w}^* + \Delta \mathbf{w}^* \approx \mathbf{F}(\mathbf{w}^*, \mathbf{z}^*)$. This iteration is recorded using CoDiPack. When the solver is recorded, the *input* is defined as the variable with respect to which the Lagrangian is differentiated, in this case $\mathbf{w}$. The *output* of the fixed-point iteration recorded, $\mathbf{w} = \mathbf{F}(\mathbf{w})$, is also $\mathbf{w}$. This is shown schematically in Fig. 3.4.

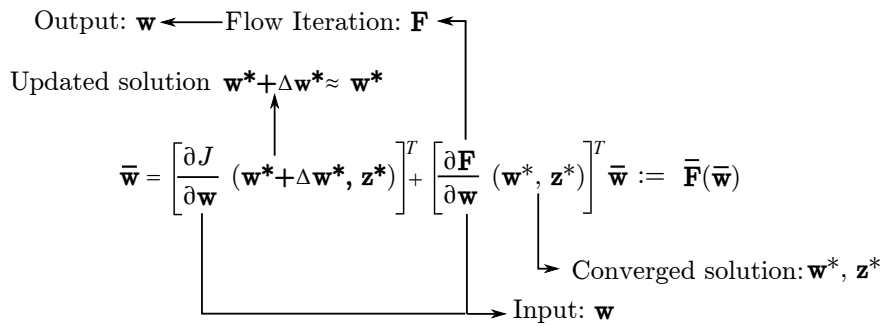


Figure 3.4: Fix-point adjoint iteration.

The recording of the path of the primal solver, $\mathbf{F}$, is outlined in Fig. 3.5,a). The solution $\mathbf{w}^*, \mathbf{z}^*$ is read from a solution file into the code, and the recording process is started. Once declared the *input* variables for the recording tool, the fixed-point solver $\mathbf{F}$ is iterated once, and the *output* variables are also set. Finally, the objective function, in this example the drag

coefficient C_D , is evaluated.

Once recorded, the adjoint problem is iteratively solved using AD (Fig. 3.5,b). The adjoint of the objective function is initialised to a unitary value, and the vector of adjoint variables $\bar{\mathbf{w}}$ is initialised to $\bar{\mathbf{w}}_0 = \mathbf{0}$. The adjoint variables are propagated in reverse through the path $\bar{\mathbf{F}}$, and the process is iterated until $\|\bar{\mathbf{w}}_{k+1} - \bar{\mathbf{w}}_k\| < \varepsilon_{tol}$ for a given tolerance.

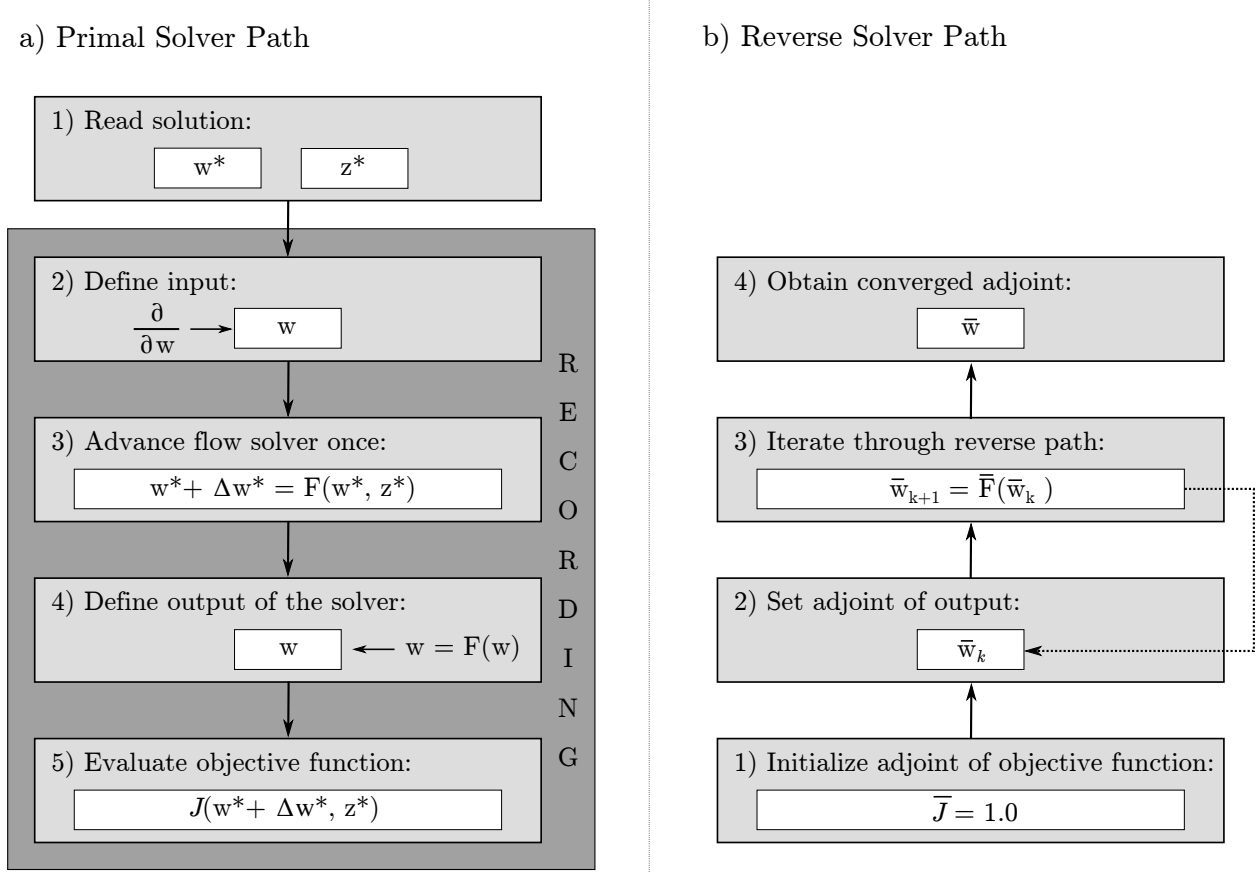
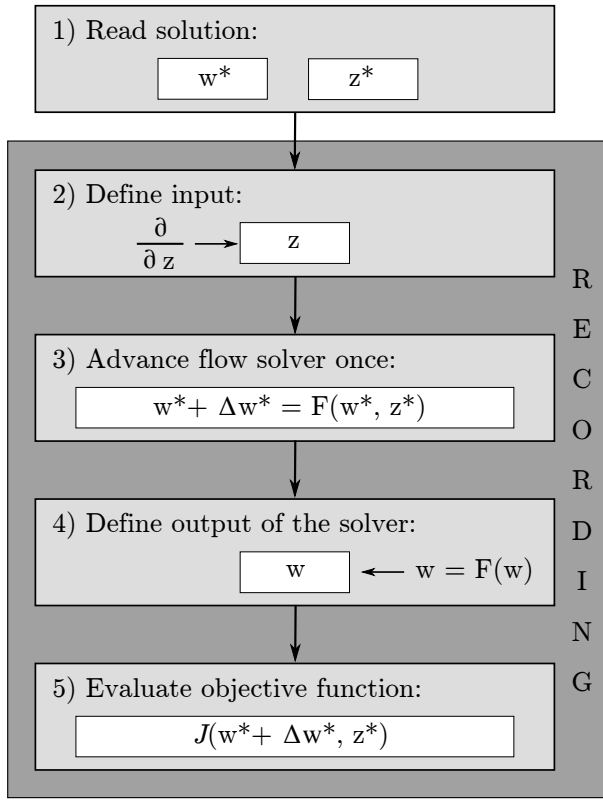


Figure 3.5: Primal (a) and reverse (b) computational paths for the solver $\mathbf{F}$.

The same principle can be applied to compute the adjoint of the mesh variables, $\mathbf{z}$. In this case, the equation (3.28) shows an explicit dependency of $\bar{\mathbf{z}}$ on the r.h.s., which nonetheless depends on $\bar{\mathbf{w}}$. The *input* of the solver is now $\mathbf{z}$, which leads to a different computational path being recorded, $\bar{\mathbf{F}}_{\mathbf{z}}$. However, the *output* of the solver remains $\mathbf{w}$. This requires using the value of $\bar{\mathbf{w}}$ to initialize the adjoint of the output, which has been previously converged. This is shown in Fig. 3.6.

Once obtained $\bar{\mathbf{z}}$, the gradient $dJ/d\boldsymbol{\alpha}$ may be obtained from Eqn. (3.29). The *input* of the computational path are the design variables $\boldsymbol{\alpha}$, while the output is now $\mathbf{z}$, given that we are recording $\mathbf{z} = \mathbf{M}(\boldsymbol{\alpha})$. Similar to the mesh adjoint, (3.29) is an explicit evaluation and therefore no iterations are required in the evaluation of the reverse path. The process is analogous to the

a) Primal Solver Path



b) Reverse Solver Path

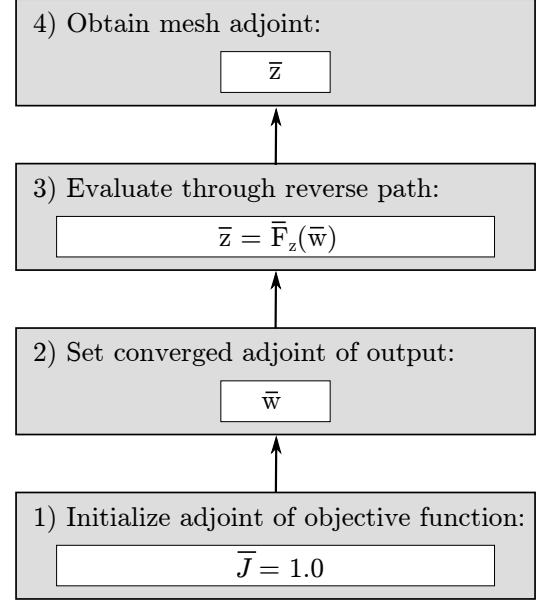
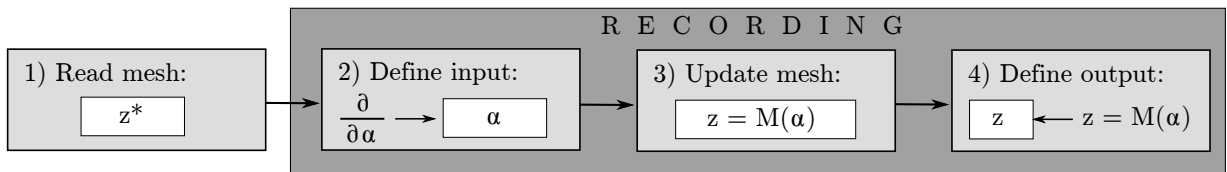


Figure 3.6: Primal (a) and reverse (b) computational paths for the solver $\mathbf{F}_{\mathbf{z}}$, where the subindex indicated the path is derived with respect to the mesh variables $\mathbf{z}$.

two previously described, however in this case the objective function is not evaluated as there is no explicit dependency of J on α . The schematic is shown in Fig. 3.7.

a) Primal Solver Path



b) Reverse Solver Path

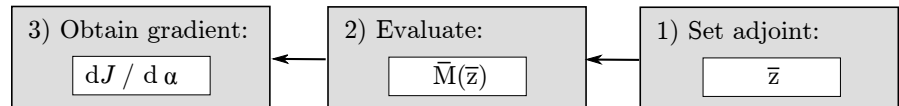


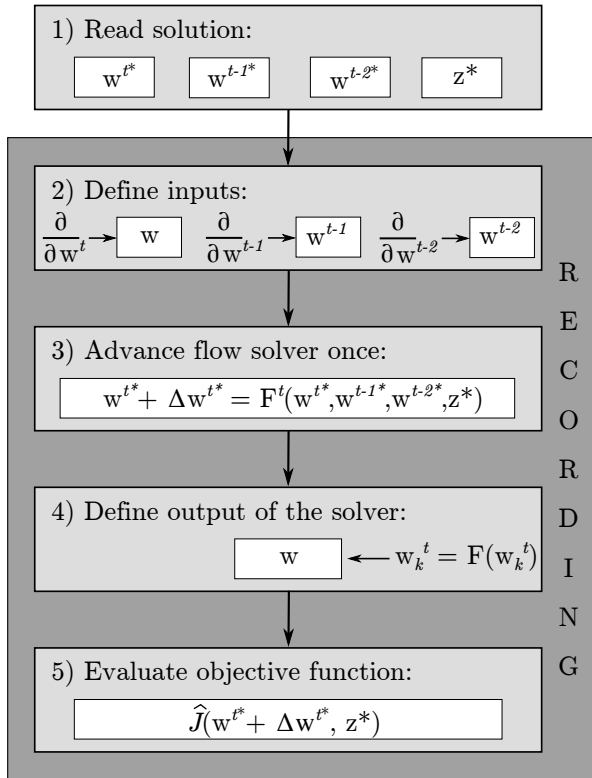
Figure 3.7: Primal (a) and reverse (b) computational paths for the calculation of the gradient.

Only slight modifications of this concept are required for the extension of the solver to time-domain flow simulations (Fig. 3.8). One of the main advantages of the adjoint method is that incorporating additional *inputs* does not increase the computational cost proportionally to their number, although it does have a larger memory footprint. This allows to define, for a fluid

iteration evaluated at time t , $\mathbf{w}^t = \mathbf{F}^t(\mathbf{w}^t, \mathbf{w}^{t-1}, \mathbf{w}^{t-2}, \mathbf{z})$, three *inputs* ($\mathbf{w}^t, \mathbf{w}^{t-1}, \mathbf{w}^{t-2}$) for a single *output* ($\mathbf{w}^t$).

The computational path $\mathbf{F}$ is recorded and run backwards in order to iterate $\bar{\mathbf{w}}^t = \bar{\mathbf{F}}_t^t(\bar{\mathbf{w}}^t)$, where $\bar{\mathbf{F}}_t^t$ corresponds to the path evaluated at time t (superscript) differentiated with respect to the variables $\mathbf{w}^t$ (subscript). The terms $\bar{\mathbf{F}}_t^{t+1}(\bar{\mathbf{w}}^{t+1})$ and $\bar{\mathbf{F}}_t^{t+2}(\bar{\mathbf{w}}^{t+2})$ must have been previously computed and are added as source terms to the fixed-point reverse iteration. Once this reverse iteration $\bar{\mathbf{w}}^t = \bar{\mathbf{F}}_t^t(\bar{\mathbf{w}}^t)$ has converged, not only the value of $\bar{\mathbf{w}}^t$ but also the values of $\bar{\mathbf{F}}_{t-1}^t(\bar{\mathbf{w}}^t)$ and $\bar{\mathbf{F}}_{t-2}^t(\bar{\mathbf{w}}^t)$ are obtained. Using a simple change of variables $T_1 = t - 1$ and $T_2 = t - 2$, it can be observed that these terms correspond to the values $\bar{\mathbf{F}}_{T_1}^{T_1+1}(\bar{\mathbf{w}}^{T_1+1})$ and $\bar{\mathbf{F}}_{T_2}^{T_2+2}(\bar{\mathbf{w}}^{T_2+2})$, that is, the source terms for the time steps T_1 and T_2 due to t .

a) Primal Solver Path



b) Reverse Solver Path

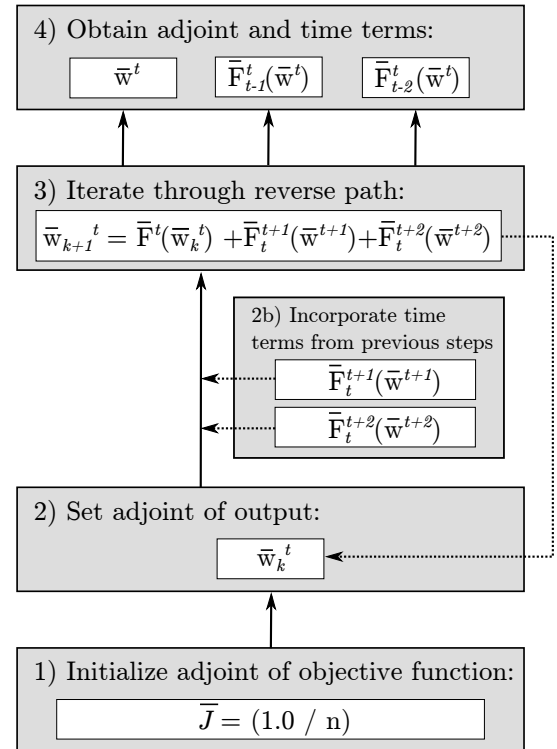


Figure 3.8: Primal (a) and reverse (b) computational paths for the time-domain solver $\mathbf{F}_n$, where the subindex indicated the path is derived with respect to the mesh variables $\mathbf{z}$.

3.4 Adjoint Methods in Structural Mechanics

The application of adjoint methods to structural mechanics is well described by Adelman and Haftka [158] for static and dynamic problems. The adjoint method is particularly efficient in problems with a large number of design variables, which has led to their extensive use in topology optimisation of structures [175, 176], see e.g., the works of Yang and Chuang [177], Buhl *et al.* [178] Jung and Hea [179], or more recently Li *et al.* [180] or Xia *et al.* [181].

Given that the stiffness matrix is symmetric for linear elastic problems, $\mathbf{K}_{\mathcal{S},L} = \mathbf{K}_{\mathcal{S},L}^T$, these problems are normally referred to as *self-adjoint*. That means that the stiffness matrix of the problem can be directly used in the solution of the adjoint problem,

$$\mathbf{K}_{\mathcal{S},L} \tilde{\mathbf{u}} = -\frac{\partial J^T}{\partial \mathbf{u}}. \quad (3.39)$$

In this work, we are interested in achieving predefined geometries by optimally designing or actuating the structures subject of study. With this in mind, we define the objective function

$$J(\mathbf{u}) = (\mathbf{u} - \mathbf{u}_{tgt})^T \mathbf{C}(\mathbf{u} - \mathbf{u}_{tgt}), \quad (3.40)$$

which measures the deformation of the structure $\mathbf{u}$ with respect to a target geometry $\mathbf{u}_{tgt}$. In (3.40), $\mathbf{C}$ is a weighting matrix of the form $\mathbf{C} = k\mathbf{I}$, whose size is the number of degrees of freedom in the structural problem. The r.h.s in Eqn. (3.39) becomes

$$\frac{\partial J}{\partial \mathbf{u}} = 2\mathbf{C}(\mathbf{u} - \mathbf{u}_{tgt}). \quad (3.41)$$

Using (3.41), it is possible to solve (3.39) and obtain the value of $\tilde{\mathbf{u}}$. This is even more efficient if the matrix $\mathbf{K}_{\mathcal{S},L}$ has been explicitly factorised for the solution of the primal problem: in those cases, the adjoint variables can be directly evaluated using the same factorisation.

Finally, substituting $\tilde{\mathbf{u}}$ into

$$\frac{dJ}{d\boldsymbol{\alpha}} = \frac{\partial J}{\partial \boldsymbol{\alpha}} + \tilde{\mathbf{u}}^T \frac{\partial \mathcal{S}}{\partial \boldsymbol{\alpha}}, \quad (3.42)$$

the gradient with respect to the design variables $\boldsymbol{\alpha}$ may be evaluated.

The problem increases in complexity in cases involving geometrical non-linearities, particu-

larly when *follower* forces are involved. Recall the structural problem in (2.36),

$$\mathcal{S}(\mathbf{u}) = \mathbf{T}(\mathbf{u}) - \mathbf{F}_\Gamma(\mathbf{u}) - \mathbf{F}_b = \mathbf{0}. \quad (3.43)$$

We define *follower* forces as those applied in the structural surface whose direction changes with the deformation of the continuum, $\mathbf{F}_\Gamma(\mathbf{u})$. Contrary to the body forces $\mathbf{F}_b$, which remain independent of the deformation $\mathbf{u}$, the surface forces do contribute to the problem Jacobian, introducing a non-symmetric term in the tangent matrix, $\mathbf{K}_{ext}$ in (2.39). Consequently, the problem is no longer *self-adjoint*,

$$\mathbf{K}_{\mathcal{S},NL}^T \tilde{\mathbf{u}} = -\frac{\partial J^T}{\partial \mathbf{u}}. \quad (3.44)$$

Analogously to the fluid problem, an approximate tangent matrix $\hat{\mathbf{K}}_{\mathcal{S},NL}$ is normally enough to converge the Newton iteration in (2.38). Therefore, the term $\mathbf{K}_{ext}$ in (2.39) can be neglected not affecting the final solution, however reducing the order of convergence. On the contrary, the exact linearisation for any given loading condition is required for the adjoint problem as defined in (3.44).

This requirement becomes more difficult to be met in cases where material non-linearities are introduced into the structural problem. The linearisation of simple hyperelastic models can be straight-forward in some cases, as shown for the neo-Hookean model in (2.25). However, for the case of more complicated material models (see e.g. Suchoki [182]) or complex effects such as viscoelasticity (see, e.g., Buoso and Palacios [21]), the linearisation of the material strain energy function and the Cauchy stress tensor may become extremely convoluted and error prone. Sometimes, the use of an approximate tangent matrix for the solution of the primal problem may become the only choice. This matrix, as explained before, cannot be used for the solution of the adjoint problem.

Because of these considerations, in this work we have adopted the methodology proposed by Albring *et al.* [120,121] and Zhou *et al.* [123,124] for fluid dynamics and aeroacoustics, and we have extended it for structural mechanics problems. This method relaxes the requirement for an *exact* analytical derivation of the Jacobian matrix. The description of this method for statics and dynamics will be presented respectively in sections 3.4.1 and 3.4.2, and our implementation in SU2 will be described in section 3.4.3

3.4.1 Structural Adjoint Based on Fixed-Point Iterators and AD

Let the structural problem in (3.43) be rewritten in the form of a fixed-point iteration,

$$\mathbf{u}^{n+1} = \mathbf{u}^n - [\mathbf{K}_{\mathcal{S}}(\mathbf{u}^n)]^{-1} \mathcal{S}(\mathbf{u}^n) =: \mathbf{S}(\mathbf{u}^n), \quad (3.45)$$

where the implicit dependency of the tangent matrix on the displacements $\mathbf{u}$ must be noted. Analogous to the fluid fixed-point iterator $\mathbf{F}$ in (3.19), the operator $\mathbf{S}(\mathbf{u}^n)$ is only feasible at the solution $\mathbf{u}^*$,

$$\mathcal{S}(\mathbf{u}^*) = \mathbf{0} \Leftrightarrow \mathbf{u}^* = \mathbf{S}(\mathbf{u}^*). \quad (3.46)$$

The optimisation problem may now be written as

$$\begin{aligned} & \min_{\boldsymbol{\alpha}} J(\mathbf{u}, \boldsymbol{\alpha}) \\ & \text{subject to } \mathbf{S}(\mathbf{u}, \boldsymbol{\alpha}) - \mathbf{u} = \mathbf{0}, \end{aligned} \quad (3.47)$$

where for the purpose of generality a generic dependency on a vector of design variables $\boldsymbol{\alpha}$ has been included in both the objective function and the structural fixed-point solver. The Lagrangian associated to this problem is

$$L(\boldsymbol{\alpha}, \mathbf{u}, \bar{\mathbf{u}}) = J(\mathbf{u}, \boldsymbol{\alpha}) + \bar{\mathbf{u}}^T [\mathbf{S}(\mathbf{u}, \boldsymbol{\alpha}) - \mathbf{u}]. \quad (3.48)$$

Following the procedure presented in section 3.3.1, differentiating the Lagrangian with respect to the state variables and after some algebra, the following fixed-point adjoint iteration in $\bar{\mathbf{u}}$ can be written,

$$\bar{\mathbf{u}}_{k+1} = \left[\frac{\partial J(\mathbf{u}^*, \boldsymbol{\alpha})}{\partial \mathbf{u}} \right]^T + \left[\frac{\partial \mathbf{S}(\mathbf{u}^*, \boldsymbol{\alpha})}{\partial \mathbf{u}} \right]^T \bar{\mathbf{u}}^n := \bar{\mathbf{S}}(\bar{\mathbf{u}}_k), \quad (3.49)$$

where the derivatives of J and $\mathbf{S}$ are evaluated around a solution of (3.46) that must have been previously obtained. The procedure of iterating over $\bar{\mathbf{S}}(\bar{\mathbf{u}}_k)$ will be later described in section 3.4.3.

Once the adjoint variables $\bar{\mathbf{u}}$ have been obtained, the total derivative of the objective function with respect to the design parameters is

$$\frac{dJ}{d\boldsymbol{\alpha}} = \frac{\partial J}{\partial \boldsymbol{\alpha}} + \bar{\mathbf{u}}^T \frac{\partial \mathbf{S}}{\partial \boldsymbol{\alpha}}. \quad (3.50)$$

Transposing expression (3.50), the gradient can be obtained from

$$\left[\frac{dJ}{d\alpha} \right]^T = \left[\frac{\partial J}{\partial \alpha}(\mathbf{u}^*, \alpha) \right]^T + \left[\frac{\partial \mathbf{S}}{\partial \alpha}(\mathbf{u}^*, \alpha) \right] \bar{\mathbf{u}}^T := \bar{\mathbf{S}}_\alpha(\bar{\mathbf{u}}), \quad (3.51)$$

thus using directly the computational path $\bar{\mathbf{S}}$ differentiated with respect to α .

The main advantage of this method is that it is able to compute the adjoint variables and the gradients even if the primal solver uses an approximate tangent matrix, $\hat{\mathbf{K}}_{\mathcal{J},NL} \neq \mathbf{K}_{\mathcal{J},NL}$. This feature provides significant advantage for complicated material models, or follower forces such as those coming from the action of a fluid flow, as it will be shown in section 3.5.1.

3.4.2 Extension of the AD-based Adjoint to Structural Dynamics

The application of adjoint methods in structural mechanics is also of practical interest in dynamical problems [183–185]. Nakshatrala and Tortorelli [186] describe two approaches to the application of adjoint methods in structural dynamics: discretise the equations and then differentiate them, or alternatively differentiate-then-discretise. The application of the adjoint method to non-linear dynamic problems is particularly complex in time-domain problems (see, e.g., [187]), as pointed out by Stanford *et al.* [188]. Nevertheless, in this work it has been found that an AD-based approach similar to the one presented in section 3.3.2 can be adopted for structural dynamics, and that it simplifies the derivation and implementation of the adjoint.

From Eqns. (2.42) and (2.46), the displacements at time t , $\mathbf{u}^t$, can be obtained from

$$\mathbf{u}_k^t = \mathbf{u}_{k-1}^t + [\mathbf{K}_T^t]^{-1} [\mathbf{T}(\mathbf{u}_{k-1}^t) - \mathbf{F}_b - \mathbf{F}_\Gamma(\mathbf{u}_{k-1}^t) - \mathbf{R}_T(\mathbf{u}_{k-1}^t, \mathbf{u}^{t-1}, \dot{\mathbf{u}}^{t-1}, \ddot{\mathbf{u}}^{t-1})], \quad (3.52)$$

where $\dot{\mathbf{u}}^{t-1}$ and $\ddot{\mathbf{u}}^{t-1}$ are computed using

$$\begin{aligned} \dot{\mathbf{u}}^t &= \dot{\mathbf{u}}^{t-1} + [(1 - \gamma)\ddot{\mathbf{u}}^{t-1} + \gamma\ddot{\mathbf{u}}^t]\Delta t, \\ \ddot{\mathbf{u}}^t &= \frac{1}{\beta\Delta t^2} (\mathbf{u}^t - \mathbf{u}^{t-1}) - \frac{1}{\beta\Delta t} \dot{\mathbf{u}}^{t-1} - \frac{1 - 2\beta}{2\beta} \ddot{\mathbf{u}}^{t-1}. \end{aligned} \quad (3.53)$$

For the development that follows, we define the *extended* nodal variable $\mathbf{q}^t = [\mathbf{u}^t, \dot{\mathbf{u}}^t, \ddot{\mathbf{u}}^t]$, which allows for the structural dynamics problem to be written as the fixed-point iterator,

$$\mathbf{q}^t = \mathbf{S}(\mathbf{q}^t, \mathbf{q}^{t-1}). \quad (3.54)$$

The optimisation problem results

$$\begin{aligned}
& \min_{\boldsymbol{\alpha}} \frac{1}{n} \sum_{k=1}^n \hat{J}(\mathbf{q}^k, \boldsymbol{\alpha}) \\
& \text{subject to } \mathbf{S}(\mathbf{q}^n, \mathbf{q}^{n-1}) - \mathbf{q}^n = \mathbf{0}, \\
& \quad \mathbf{S}(\mathbf{q}^{n-1}, \mathbf{q}^{n-2}) - \mathbf{q}^{n-1} = \mathbf{0}, \\
& \quad \dots \\
& \quad \mathbf{S}(\mathbf{q}^2, \mathbf{q}^1) - \mathbf{q}^2 = \mathbf{0}, \\
& \quad \mathbf{S}(\mathbf{q}^1, \mathbf{q}^0) - \mathbf{q}^1 = \mathbf{0}.
\end{aligned} \tag{3.55}$$

which allows to define the Lagrangian function

$$\mathcal{L} = \frac{1}{n} \sum_{k=1}^n \hat{J}(\mathbf{q}^k, \boldsymbol{\alpha}) + \sum_{k=1}^n [\bar{\mathbf{q}}^k]^T [\mathbf{S}(\mathbf{q}^k, \mathbf{q}^{k-1}) - \mathbf{q}^k]. \tag{3.56}$$

Following the procedure presented in section 3.3.2, the adjoint equations result

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mathbf{q}^n} &= \frac{1}{n} \frac{\partial \hat{J}}{\partial \mathbf{q}^n}(\mathbf{q}^n, \boldsymbol{\alpha}) + [\bar{\mathbf{q}}^n]^T \frac{\partial \mathbf{S}}{\partial \mathbf{q}^n}(\mathbf{q}^n, \mathbf{q}^{n-1}) - [\bar{\mathbf{q}}^n]^T &= \mathbf{0}, \\
\frac{\partial \mathcal{L}}{\partial \mathbf{q}^{n-1}} &= \frac{1}{n} \frac{\partial \hat{J}}{\partial \mathbf{q}^{n-1}}(\mathbf{q}^{n-1}, \boldsymbol{\alpha}) + [\bar{\mathbf{q}}^n]^T \frac{\partial \mathbf{S}}{\partial \mathbf{q}^{n-1}}(\mathbf{q}^n, \mathbf{q}^{n-1}) \\
&\quad + [\bar{\mathbf{q}}^{n-1}]^T \frac{\partial \mathbf{S}}{\partial \mathbf{q}^{n-1}}(\mathbf{q}^{n-1}, \mathbf{q}^{n-2}) - [\bar{\mathbf{q}}^{n-1}]^T &= \mathbf{0}, \\
&\dots \\
\frac{\partial \mathcal{L}}{\partial \mathbf{q}^1} &= \frac{1}{n} \frac{\partial \hat{J}}{\partial \mathbf{q}^1}(\mathbf{q}^1, \boldsymbol{\alpha}) + [\bar{\mathbf{q}}^2]^T \frac{\partial \mathbf{S}}{\partial \mathbf{q}^1}(\mathbf{q}^2, \mathbf{q}^1) \\
&\quad + [\bar{\mathbf{q}}^1]^T \frac{\partial \mathbf{S}}{\partial \mathbf{q}^1}(\mathbf{q}^1, \mathbf{q}^0) - [\bar{\mathbf{q}}^1]^T &= \mathbf{0},
\end{aligned} \tag{3.57}$$

which are rewritten using the same criteria as for (3.36),

$$\left\{ \begin{array}{l} \bar{\mathbf{q}}^n = \bar{\mathbf{S}}_n^n(\bar{\mathbf{q}}^n), \\ \bar{\mathbf{q}}^{n-1} = \bar{\mathbf{S}}_{n-1}^n(\bar{\mathbf{q}}^n) + \bar{\mathbf{S}}_{n-1}^{n-1}(\bar{\mathbf{q}}^{n-1}), \\ \dots \\ \bar{\mathbf{q}}^1 = \bar{\mathbf{S}}_1^2(\bar{\mathbf{q}}^2) + \bar{\mathbf{S}}_1^1(\bar{\mathbf{q}}^1), \end{array} \right. \quad (3.58)$$

and finally generalised for a time position t using

$$\bar{\mathbf{q}}^t = \bar{\mathbf{S}}_t^t(\bar{\mathbf{q}}^t) + \bar{\mathbf{S}}_t^{t+1}(\bar{\mathbf{q}}^{t+1}). \quad (3.59)$$

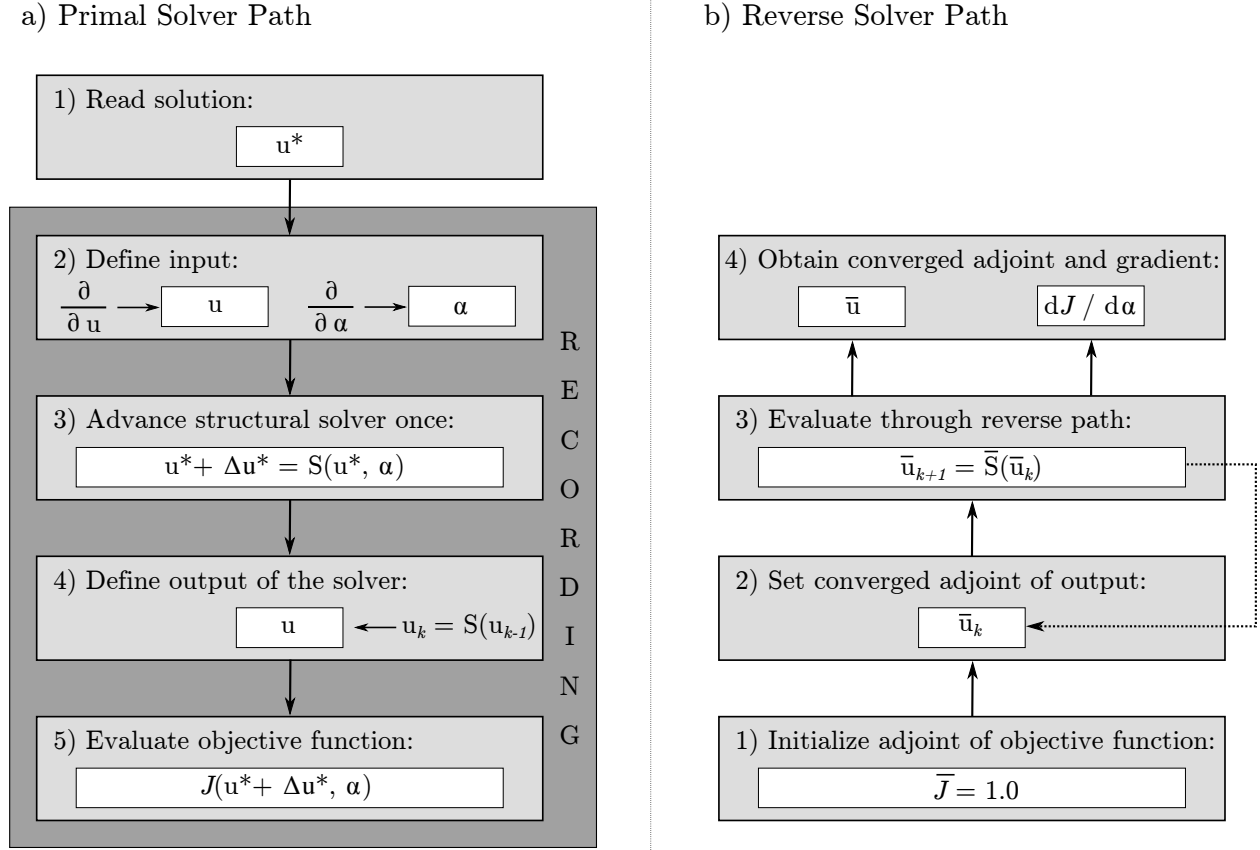
Analogous to the fluid time-domain problem, the adjoint simulation requires to be run backwards in time, starting from the time position n .

3.4.3 Solver Implementation

The implementation of the AD-based adjoint for structural mechanics in SU2 has been carried out by the author as part of this project, based on the description provided in section 3.3.3, and in collaboration with the group of Professor Gauger at TU Kaiserslautern who provided support in the use of the AD tool in SU2.

Let the design variables $\boldsymbol{\alpha}$ be input parameters of the problem, for example, the distribution of stiffness in l different regions of the solid, that is, $[E_1, E_2, \dots, E_l]$. In the definition of the recording, it is possible to add these parameters to the *input* together with the state variables $\mathbf{u}$, thus differentiating the solver $\mathbf{S}$ simultaneously with respect to $\mathbf{u}$ and $\boldsymbol{\alpha}$. Due to the parallelism between expressions (3.49) and (3.51), once the adjoint variables $\bar{\mathbf{u}}$ are converged, the gradient is immediately obtained with no additional effort (Fig.3.9,b). The *output* of the structural solver corresponds to $\mathbf{u} = \mathbf{S}(\mathbf{u})$ (Fig.3.9,a).

For time-domain simulations, the influence of the extended state variables for the previous time step, $\mathbf{q}^{t-1}$, is added to the list of inputs. This allows one to obtain the source term that will be added in the next iteration, corresponding to the previous time step, $T_1 = t - 1$. The design variables $\boldsymbol{\alpha}$ are maintained in the list of inputs, as they allow to compute the instantaneous

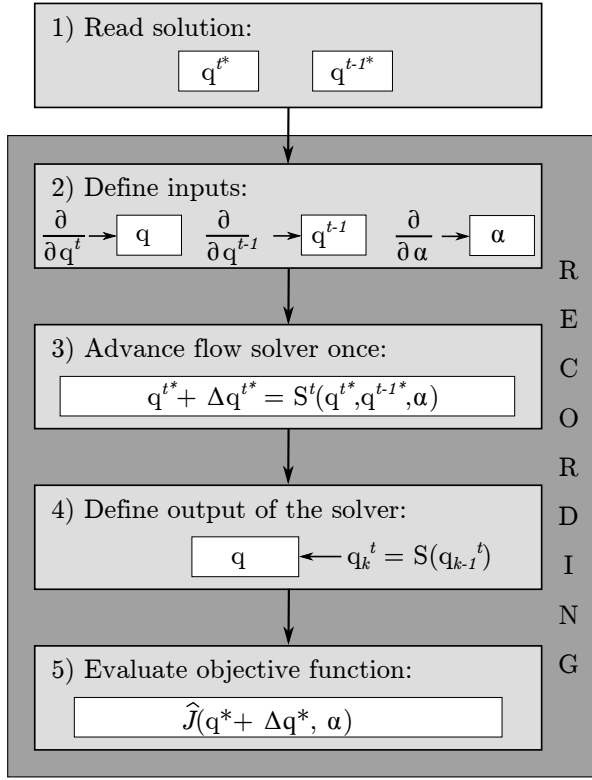
Figure 3.9: Primal (a) and reverse (b) computational paths for the solver S .

gradient term corresponding to time t , $dJ^t/d\alpha$, once the value of $\bar{\mathbf{q}}^t$ has been converged. The simulation is run for the same time window as the primal solver, in reverse, $t = n, n - 1, \dots, 3, 2, 1$. Once all the time steps have been converged, the averaged gradient is obtained from $1/n \sum_{k=1}^n dJ^k/d\alpha$. A schematic of this process is shown in Fig. (3.10). Section 6.1 will investigate the numerical performance of this implementation in the computation of gradients for structural problems.

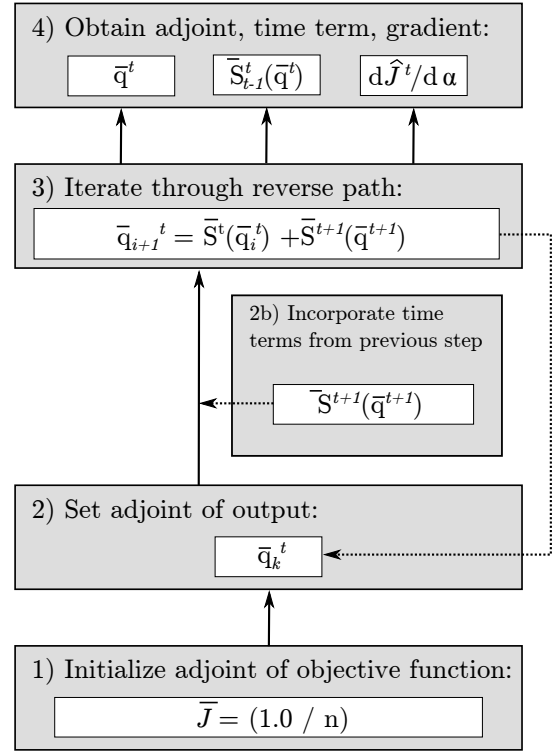
3.5 Coupled Adjoint Methods

Some authors adopt a two-field formulation [50, 51, 57, 91] for the definition of the coupled adjoint problem in FSI, where the deformation of the mesh is accounted for within the fluid problem, $\mathcal{F} = \mathcal{F}(\mathbf{w}, \mathbf{z})$. Although this methodology has been proven suitable to very large aero-structural problems in which the deformations of the mesh remain small, in this work we are interested in non-linear structural behaviour with large displacements. Therefore, a 3-field formulation [81] is adopted here, where the mesh $\mathcal{M} = \mathcal{M}(\mathbf{u}, \mathbf{z})$ becomes the third field of

a) Primal Solver Path



b) Reverse Solver Path

Figure 3.10: Primal (a) and reverse (b) computational paths for the solver $\mathbf{S}$.

the problem. This method has been adopted in the works by Maute *et al.* [42, 48, 49, 189], and according to [42] it is more suitable for problems with large structural deformations, even though it has a higher computational cost. The three-field formulation leads to the governing equations for a steady-state FSI problem,

$$\mathcal{G}(\mathbf{u}, \mathbf{w}, \mathbf{z}) = \mathbf{0} \Leftrightarrow \begin{cases} \mathcal{S}(\mathbf{u}, \mathbf{w}, \mathbf{z}, \boldsymbol{\alpha}) = \mathbf{0}, \\ \mathcal{F}(\mathbf{w}, \mathbf{z}, \boldsymbol{\alpha}) = \mathbf{0}, \\ \mathcal{M}(\mathbf{u}, \mathbf{z}, \boldsymbol{\alpha}) = \mathbf{0}. \end{cases} \quad (3.60)$$

The problem defined in Eq. (3.60) may now be linearised as

$$\begin{bmatrix} \mathcal{S}(\mathbf{u}, \mathbf{w}, \mathbf{z}, \boldsymbol{\alpha}) \\ \mathcal{F}(\mathbf{w}, \mathbf{z}, \boldsymbol{\alpha}) \\ \mathcal{M}(\mathbf{u}, \mathbf{z}, \boldsymbol{\alpha}) \end{bmatrix} + \begin{bmatrix} \frac{\partial \mathcal{S}}{\partial \mathbf{u}} & \frac{\partial \mathcal{S}}{\partial \mathbf{w}} & \frac{\partial \mathcal{S}}{\partial \mathbf{z}} \\ \mathbf{0} & \frac{\partial \mathcal{F}}{\partial \mathbf{w}} & \frac{\partial \mathcal{F}}{\partial \mathbf{z}} \\ \frac{\partial \mathcal{M}}{\partial \mathbf{u}} & \mathbf{0} & \frac{\partial \mathcal{M}}{\partial \mathbf{z}} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{u} \\ \Delta \mathbf{w} \\ \Delta \mathbf{z} \end{bmatrix} = \mathbf{0}, \quad (3.61)$$

where the diagonal terms are the structural, non-linear tangent matrix $\mathbf{K}_{\mathcal{S},NL}$, the Jacobian of the fluid problem $\mathbf{H}$ and the fictitious stiffness matrix for the mesh problem $\tilde{\mathbf{K}}_{\mathcal{M}}$. Considering these redefinitions, the adjoint problem in Fluid-Structure Interaction analysis can be written as

$$\frac{dJ}{d\alpha} = \frac{\partial J}{\partial \alpha} + [\bar{\mathbf{u}}^T \bar{\mathbf{w}}^T \bar{\mathbf{z}}^T] \begin{bmatrix} \frac{\partial \mathcal{S}}{\partial \alpha} \\ \frac{\partial \mathcal{F}}{\partial \alpha} \\ \frac{\partial \mathcal{M}}{\partial \alpha} \end{bmatrix}, \quad (3.62)$$

where the adjoint variables are computed solving the linear system

$$\begin{bmatrix} \mathbf{K}_{\mathcal{S},NL}^T & \mathbf{0} & \frac{\partial \mathcal{M}^T}{\partial \mathbf{u}} \\ \frac{\partial \mathcal{S}^T}{\partial \mathbf{w}} & \mathbf{H}^T & \mathbf{0} \\ \frac{\partial \mathcal{S}^T}{\partial \mathbf{z}} & \frac{\partial \mathcal{F}^T}{\partial \mathbf{z}} & \tilde{\mathbf{K}}_{\mathcal{M}} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{u}} \\ \bar{\mathbf{w}} \\ \bar{\mathbf{z}} \end{bmatrix} = - \begin{bmatrix} \frac{\partial J}{\partial \mathbf{u}} \\ \frac{\partial J}{\partial \mathbf{w}} \\ \frac{\partial J}{\partial \mathbf{z}} \end{bmatrix}. \quad (3.63)$$

Once obtained the expression of the adjoint problem in a FSI context, there are some terms in Eq. (3.63) that require some further understanding. It was recently pointed out by Kenway *et al.* [57] that the effect of the changes in the normals with the applied aerodynamic forces in the FSI boundary $\mathbf{F}_\Gamma$, $\partial \mathbf{F}_\Gamma / \partial \mathbf{u}$, had to be taken into account. This effect corresponds to the directional derivative of the external work in the direction of an infinitesimal perturbation of the structural variables $D\delta W_{ext}(\phi_k, \delta \mathbf{v})[\Delta \mathbf{u}]$, as previously stated in Eq. (2.32). The discretised term, as shown in (2.41b), breaks the symmetry of the non-linear tangent matrix, $\mathbf{K}_{\mathcal{S},NL} \neq \mathbf{K}_{\mathcal{S},NL}^T$. Therefore, the follower forces due to the action of the fluid cause the structural problem to be no longer self-adjoint in the context of Fluid-Structure Interaction.

The terms $(\partial \mathcal{S} / \partial \mathbf{w})^T$ and $(\partial \mathcal{S} / \partial \mathbf{z})^T$ correspond to the derivative of the aerodynamic forces with respect to the fluid variables $(\partial \mathbf{F}_\Gamma / \partial \mathbf{w})^T$ and to the fluid mesh displacements, $(\partial \mathbf{F}_\Gamma / \partial \mathbf{z})^T$. Finally, the term $(\partial \mathcal{F} / \partial \mathbf{z})^T$ accounts for the sensitivity of the fluid problem with respect to variations in the positions of the fluid mesh, both in the FSI interface and its internal nodes.

The computation of the previously defined cross terms is convoluted and requires special attention. In the solution of the direct problem, (3.61), the objective is to iteratively obtain the increments of the state variables $\Delta \mathbf{x}$ that minimize the value of the residual. Therefore, in that case it is in general sufficient to employ an approximate tangent matrix which is good enough to achieve convergence. Because of this, it is possible to omit the computation of the cross terms altogether, which is the basis of the non-linear Block-Gauss-Seidel method as described

in Algorithm 3 and in references [42, 43, 57] among many others.

However, the adjoint problem (3.63) is no longer a linearisation of a set of non-linear equations, but it is a system of linear equations in which the coefficients matrix of the system must be computed exactly. Although it is possible to solve the problem using iterative approaches such as the Lagged Coupled Adjoint (LCA) method proposed by Martins *et al.* in [91], it is necessary to add the contributions of the crossed terms in the matrix in the right hand side of the reduced systems of equations, and then iterate until convergence.

The computation of these terms becomes, as a result, one of the most important (and complex) tasks in the solution of the adjoint FSI problem. If it is possible, they should be computed analytically. However, this is not always an option, and therefore some alternatives arise. There are several options to compute sensitivities, according to Martins *et al.* [91]. One of the most widely used methods is finite differences, due to their ease of implementation. A more accurate method to compute such sensitivities, yet still simple to implement, is the use of the complex-step derivative approximation [41]. Finally, it is possible to use the method of Algorithmic Differentiation (AD) [114, 115] both in its forward and reverse modes. The use given until now for AD in this context has been selective, as it has been applied only to particular subroutines for the computation of some of the crossed-terms (see, e.g., Kenway *et al* [57]). All of these methods, except for the reverse mode of AD, require one computation per variable, which may be a prohibitive cost in realistic applications [41].

3.5.1 A Novel Coupled Adjoint Method for FSI

In this section, a novel methodology that consistently extends the AD-based discrete adjoint described in sections 3.3.1 and 3.4.1 to fully-coupled, steady-state Fluid-Structure Interaction problems is presented. This method has been fully developed and implemented as a part of the works carried out by the author for this PhD thesis. Its objective is to overcome some of the difficulties described in the previous section, and to define a framework in which the adjoint of the coupled FSI problem may be obtained for problems involving complex phenomena in one or both fields involved.

Each of the three sub-problems defined in (2.53) is now rewritten in the form of fixed-point iterations, where the time-domain terms have been dropped at this stage and a generic

dependency on a vector of design variables $\boldsymbol{\alpha}$ has been included for generality purposes:

$$\mathbf{x} = \mathbf{G}(\mathbf{x}, \boldsymbol{\alpha}) \Leftrightarrow \begin{cases} \mathbf{u} &= \mathbf{S}(\mathbf{u}, \mathbf{w}, \mathbf{z}, \boldsymbol{\alpha}), \\ \mathbf{w} &= \mathbf{F}(\mathbf{w}, \mathbf{z}, \boldsymbol{\alpha}), \\ \mathbf{z} &= \mathbf{M}(\mathbf{u}, \boldsymbol{\alpha}). \end{cases} \quad (3.64)$$

From Eqn. (3.64), it becomes apparent that the structural fixed-point solver $\mathbf{S}$ depends on its own state variables $\mathbf{u}$, on the flow state $\mathbf{w}$ and on the positions of the flow nodes $\mathbf{z}$. These dependencies arise from the condition (2.51), which determines the traction state on the interface.

The fluid problem depends on the conservative variables $\mathbf{w}$ and on the position of the nodes in the fluid mesh, $\mathbf{z}$, both on the FSI interface ($\mathbf{z}_\Gamma \in \mathbf{z}$) and the interior of the domain ($\mathbf{z}_\Omega \in \mathbf{z}$). Finally, the mesh problem depends on the position of the interface, which according to the continuity condition (2.50) is determined by the structural domain. From (3.22), the problem is linear, where now the fictitious forces $\tilde{\mathbf{f}}(\mathbf{u}, \boldsymbol{\alpha})$ are a function of the displacements of the structural domain on the interface, $\mathbf{u}_\Gamma \in \mathbf{u}$.

The overall optimization problem takes the form

$$\begin{aligned} \min \quad & J(\mathbf{u}, \mathbf{w}, \mathbf{z}, \boldsymbol{\alpha}) \\ \text{subject to} \quad & \mathbf{S}(\mathbf{u}, \mathbf{w}, \mathbf{z}, \boldsymbol{\alpha}) - \mathbf{u} = \mathbf{0}, \\ & \mathbf{F}(\mathbf{w}, \mathbf{z}, \boldsymbol{\alpha}) - \mathbf{w} = \mathbf{0}, \\ & \mathbf{M}(\mathbf{u}, \boldsymbol{\alpha}) - \mathbf{z} = \mathbf{0}, \end{aligned} \quad (3.65)$$

which leads to the following definition of the Lagrangian of the coupled problem:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\alpha}, \mathbf{u}, \bar{\mathbf{u}}, \mathbf{w}, \bar{\mathbf{w}}, \mathbf{z}, \bar{\mathbf{z}}) = & J(\mathbf{u}, \mathbf{w}, \mathbf{z}, \boldsymbol{\alpha}) + \bar{\mathbf{u}}^T [\mathbf{S}(\mathbf{u}, \mathbf{w}, \mathbf{z}, \boldsymbol{\alpha}) - \mathbf{u}] \\ & + \bar{\mathbf{w}}^T [\mathbf{F}(\mathbf{w}, \mathbf{z}, \boldsymbol{\alpha}) - \mathbf{w}] + \bar{\mathbf{z}}^T [\mathbf{M}(\mathbf{u}, \boldsymbol{\alpha}) - \mathbf{z}]. \end{aligned} \quad (3.66)$$

Differentiating (3.66) with respect to the structural variables $\mathbf{u}$, the flow conservative vari-

ables $\mathbf{w}$, and the mesh variables $\mathbf{z}$, we obtain the adjoint equations

$$\begin{aligned}\bar{\mathbf{u}}^T &= \frac{\partial J}{\partial \mathbf{u}}(\mathbf{u}, \mathbf{w}, \mathbf{z}, \boldsymbol{\alpha}) + \bar{\mathbf{u}}^T \frac{\partial \mathbf{S}}{\partial \mathbf{u}}(\mathbf{u}, \mathbf{w}, \mathbf{z}, \boldsymbol{\alpha}) + \bar{\mathbf{z}}^T \frac{\partial \mathbf{M}}{\partial \mathbf{u}}(\mathbf{u}), \\ \bar{\mathbf{w}}^T &= \frac{\partial J}{\partial \mathbf{w}}(\mathbf{u}, \mathbf{w}, \mathbf{z}, \boldsymbol{\alpha}) + \bar{\mathbf{w}}^T \frac{\partial \mathbf{F}}{\partial \mathbf{w}}(\mathbf{w}, \mathbf{z}, \boldsymbol{\alpha}) + \bar{\mathbf{u}}^T \frac{\partial \mathbf{S}}{\partial \mathbf{w}}(\mathbf{u}, \mathbf{w}, \mathbf{z}, \boldsymbol{\alpha}), \\ \bar{\mathbf{z}}^T &= \frac{\partial J}{\partial \mathbf{z}}(\mathbf{u}, \mathbf{w}, \mathbf{z}, \boldsymbol{\alpha}) + \bar{\mathbf{w}}^T \frac{\partial \mathbf{F}}{\partial \mathbf{z}}(\mathbf{w}, \mathbf{z}, \boldsymbol{\alpha}) + \bar{\mathbf{u}}^T \frac{\partial \mathbf{S}}{\partial \mathbf{z}}(\mathbf{u}, \mathbf{w}, \mathbf{z}, \boldsymbol{\alpha}).\end{aligned}\tag{3.67}$$

Finally, the gradient of the objective function is

$$\frac{dJ}{d\boldsymbol{\alpha}} = \frac{\partial J}{\partial \boldsymbol{\alpha}} + \bar{\mathbf{u}}^T \frac{\partial \mathbf{S}}{\partial \boldsymbol{\alpha}} + \bar{\mathbf{w}}^T \frac{\partial \mathbf{G}}{\partial \boldsymbol{\alpha}} + \bar{\mathbf{z}}^T \frac{\partial \mathbf{M}}{\partial \boldsymbol{\alpha}}.\tag{3.68}$$

We evaluate the derivatives of $\mathcal{L}$ around the feasible solution of the FSI problem, defined by $\mathbf{u}^*$, $\mathbf{w}^*$, and $\mathbf{z}^*$, in a similar fashion as in (3.49). The solution of (3.67) is sought again using an iterative procedure. For that purpose, we approximate it by the fixed-point iteration

$$\bar{\mathbf{x}} = \bar{\mathbf{G}}(\bar{\mathbf{x}}) \Leftrightarrow \begin{cases} \bar{\mathbf{u}}_{k+1} &= \bar{\mathbf{S}}(\bar{\mathbf{u}}_k) + \bar{\mathbf{M}}_{\mathbf{u}}(\bar{\mathbf{z}}), \\ \bar{\mathbf{w}}_{k+1} &= \bar{\mathbf{F}}(\bar{\mathbf{w}}_k) + \bar{\mathbf{S}}_{\mathbf{w}}(\bar{\mathbf{u}}), \\ \bar{\mathbf{z}} &= \bar{\mathbf{F}}_{\mathbf{z}}(\bar{\mathbf{w}}) + \bar{\mathbf{S}}_{\mathbf{z}}(\bar{\mathbf{u}}), \end{cases}\tag{3.69}$$

where the subscripts k and $k+1$ stand for the internal subiterations required both for the structural and fluid adjoints, and $\bar{\mathbf{M}}_{\mathbf{u}}(\bar{\mathbf{z}})$, $\bar{\mathbf{S}}_{\mathbf{w}}(\bar{\mathbf{u}})$, $\bar{\mathbf{F}}_{\mathbf{z}}(\bar{\mathbf{w}})$ and $\bar{\mathbf{S}}_{\mathbf{z}}(\bar{\mathbf{u}})$ correspond to crossed dependencies of the solvers $\mathbf{S}$, $\mathbf{F}$ and $\mathbf{M}$ differentiated with respect to state variables of the rest of the fields.

For the solution of (3.69), in this work a block Gauss-Seidel procedure has been adopted, analogous to the Lagged Coupled Adjoint method proposed by Martins *et al.* [91]. Barcelos and Maute [43] used a very similar method for the computation of FSI sensitivities using a 3-field formulation and the direct approach, which also requires incorporating coupled terms to the formulation. This method is described in Algorithm 4.

3.5.2 Implementation of the Coupled Adjoint Method using AD

This implementation has been carried by the author for this work, and will be briefly described next. In order to solve the system in (3.64), it is necessary to selectively record each independent

Algorithm 4: Linear Block Gauss-Seidel algorithm for coupled adjoint

 $n \leftarrow 0, \bar{\mathbf{u}} \leftarrow \mathbf{0}, \bar{\mathbf{w}} \leftarrow \mathbf{0}, \bar{\mathbf{z}} \leftarrow \mathbf{0}$
while $n < N$ **do**
while $k < k_{max,s}$ **do**
 $\bar{\mathbf{u}}_{k+1}^{n+1} = \bar{\mathbf{S}}(\bar{\mathbf{u}}_k^{n+1}) + \bar{\mathbf{M}}_{\mathbf{u}}(\bar{\mathbf{z}}^n)$

 Evaluate structural crossed terms: $\bar{\mathbf{S}}_{\mathbf{w}}(\bar{\mathbf{u}}^{n+1}), \bar{\mathbf{S}}_{\mathbf{z}}(\bar{\mathbf{u}}^{n+1})$
while $k < k_{max,f}$ **do**
 $\bar{\mathbf{w}}_{k+1}^{n+1} = \bar{\mathbf{F}}(\bar{\mathbf{w}}_k^{n+1}) + \bar{\mathbf{S}}_{\mathbf{w}}(\bar{\mathbf{u}}^{n+1})$

 Evaluate flow crossed term: $\bar{\mathbf{F}}_{\mathbf{z}}(\bar{\mathbf{w}}^{n+1})$

 Compute $\bar{\mathbf{z}}$: $\bar{\mathbf{z}}^{n+1} = \bar{\mathbf{S}}_{\mathbf{z}}(\bar{\mathbf{u}}^{n+1}) + \bar{\mathbf{F}}_{\mathbf{z}}(\bar{\mathbf{w}}^{n+1})$

 Evaluate mesh crossed term: $\bar{\mathbf{M}}_{\mathbf{u}}(\bar{\mathbf{z}}^{n+1})$
if $\|\bar{\mathbf{u}}_{n+1} - \bar{\mathbf{u}}_n\| < \epsilon_s, \|\bar{\mathbf{w}}_{n+1} - \bar{\mathbf{w}}_n\| < \epsilon_f$ **then**
 $\bar{\mathbf{u}} \leftarrow \bar{\mathbf{u}}_{n+1}$
 $\bar{\mathbf{w}} \leftarrow \bar{\mathbf{w}}_{n+1}$
 $\bar{\mathbf{z}} \leftarrow \bar{\mathbf{z}}_{n+1}$
break
else
 $n \leftarrow n + 1$

solver ($\mathbf{S}$, $\mathbf{F}$ and $\mathbf{M}$), adequately choosing the input and output to capture all the crossed-term dependencies. These dependencies mathematically resemble the coupling effects in the solution of the primal problem. Let the cross-dependencies be redefined as

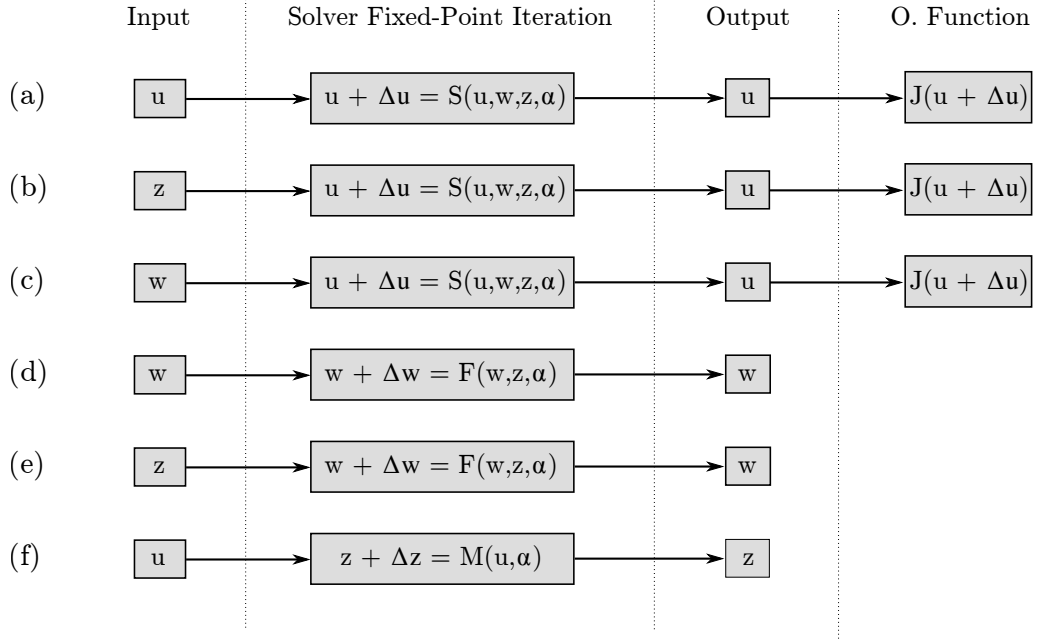
$$\bar{\mathbf{w}}_{\bar{\mathbf{u}}} = \bar{\mathbf{S}}_{\mathbf{w}}(\bar{\mathbf{u}}) ; \quad \bar{\mathbf{u}}_{\bar{\mathbf{z}}} = \bar{\mathbf{M}}_{\mathbf{u}}(\bar{\mathbf{z}}) ; \quad \bar{\mathbf{z}}_{\bar{\mathbf{w}}} = \bar{\mathbf{F}}_{\mathbf{z}}(\bar{\mathbf{w}}) ; \quad \bar{\mathbf{z}}_{\bar{\mathbf{u}}} = \bar{\mathbf{S}}_{\mathbf{z}}(\bar{\mathbf{u}}). \quad (3.70)$$

For an efficient implementation, it is necessary to distinguish between two kinds of objective functions. If the objective function is *structural*, that is, it is a function of the state variables of the structural problem (see, e.g., (1.2)), its contribution needs to be included in the evaluation of the computational path of $\mathbf{S}$. For this case, the paths of the primal and reverse solvers are presented in Fig. 3.11.

Initially, the structural adjoint $\bar{\mathbf{u}}$ is evaluated neglecting the contribution of the crossed term $\bar{\mathbf{u}}_{\bar{\mathbf{z}}}$ (Fig. 3.11,a). An approximate solution of $\bar{\mathbf{u}}$ is used in the computation of the effects of the structural variables on the mesh and fluid adjoints, $\bar{\mathbf{z}}_{\bar{\mathbf{u}}}$ and $\bar{\mathbf{w}}_{\bar{\mathbf{u}}}$ respectively (Figs. 3.11,b,c). The fluid adjoint is then evaluated (Fig. 3.11,d). Although there is no direct dependency of J in $\mathbf{w}$ for *structural* objective functions, the solution of $\bar{\mathbf{w}} = \bar{\mathbf{F}}(\bar{\mathbf{w}})$ is non-trivial due to the source term $\bar{\mathbf{w}}_{\bar{\mathbf{u}}}$. Once the approximate solution of $\bar{\mathbf{w}}$ has converged, the crossed term $\bar{\mathbf{z}}_{\bar{\mathbf{w}}}$ is obtained (Fig. 3.11,e). This allows to further compute the mesh adjoint $\bar{\mathbf{z}}$ and lead to the calculation of the remaining term, $\bar{\mathbf{u}}_{\bar{\mathbf{z}}}$ (Fig. 3.11,f). The term $\bar{\mathbf{u}}_{\bar{\mathbf{z}}}$ is then added to the recomputation of $\bar{\mathbf{u}} = \bar{\mathbf{S}}(\bar{\mathbf{u}})$, and the process is iteratively run until convergence of the adjoint variables.

Only minor modifications are required in order to obtain the gradients for fluid objective functions, namely, the drag or lift coefficients or the aerodynamic efficiency. In this case, if the procedure were to be followed as in Fig. 3.11, the initial solution for $\bar{\mathbf{u}}$ would be trivial. Moreover, the contribution of J would need to be included in the evaluation of the computational path of $\mathbf{F}$. Although in this work we have limited ourselves to structural objective functions, the structure of the code for fluid objective functions has also been incorporated to SU2, and is presented in Fig. 3.12

Primal solver paths



Reverse solver paths and crossed-dependencies

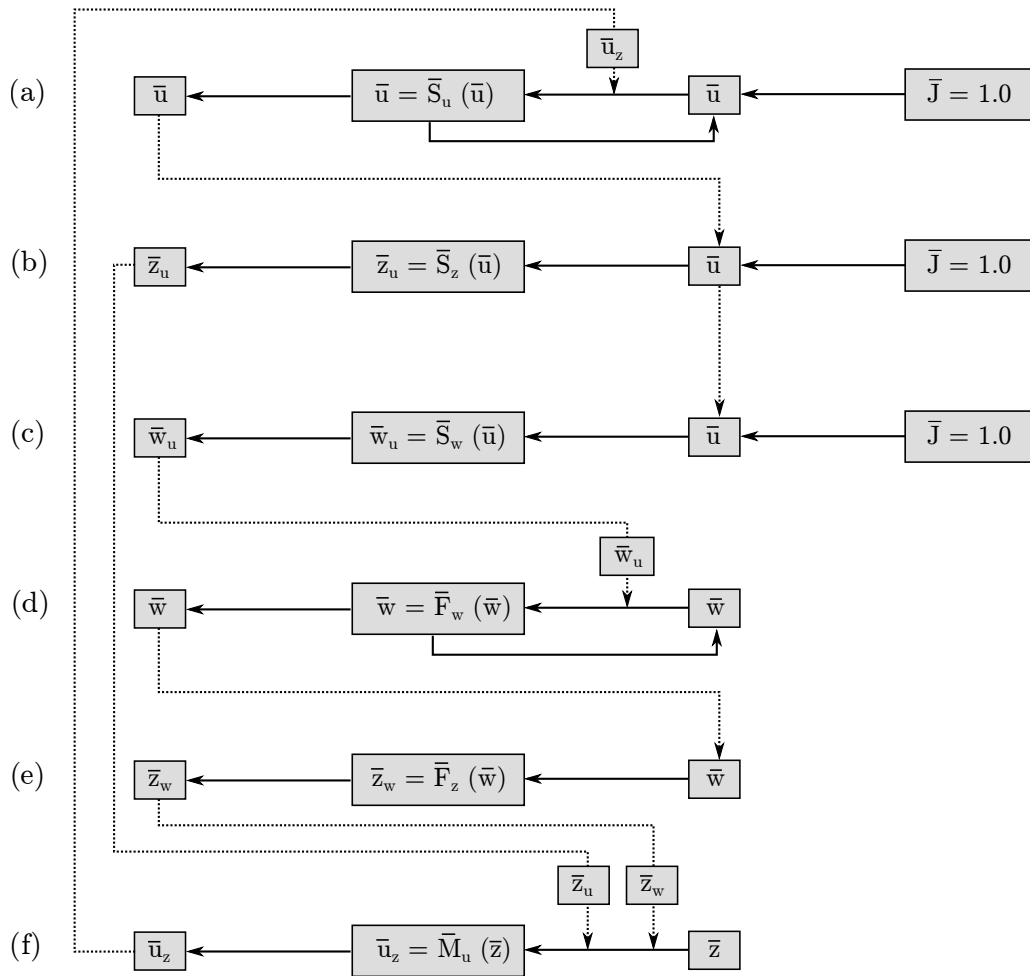
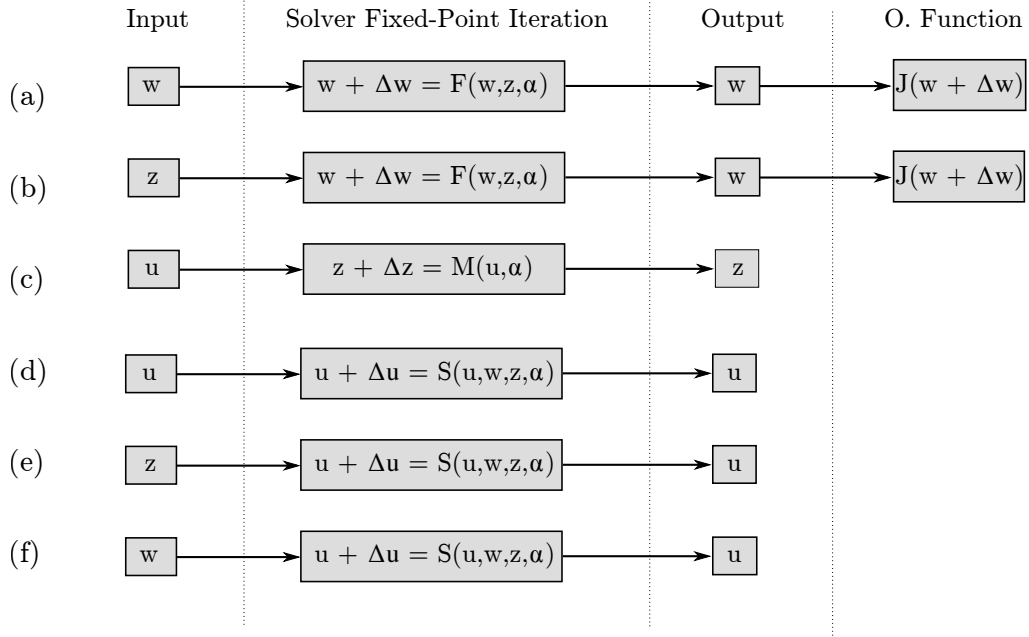


Figure 3.11: Primal and reverse computational paths for the FSI solver, using a structural-based objective function.

Primal solver paths



Reverse solver paths and crossed-dependencies

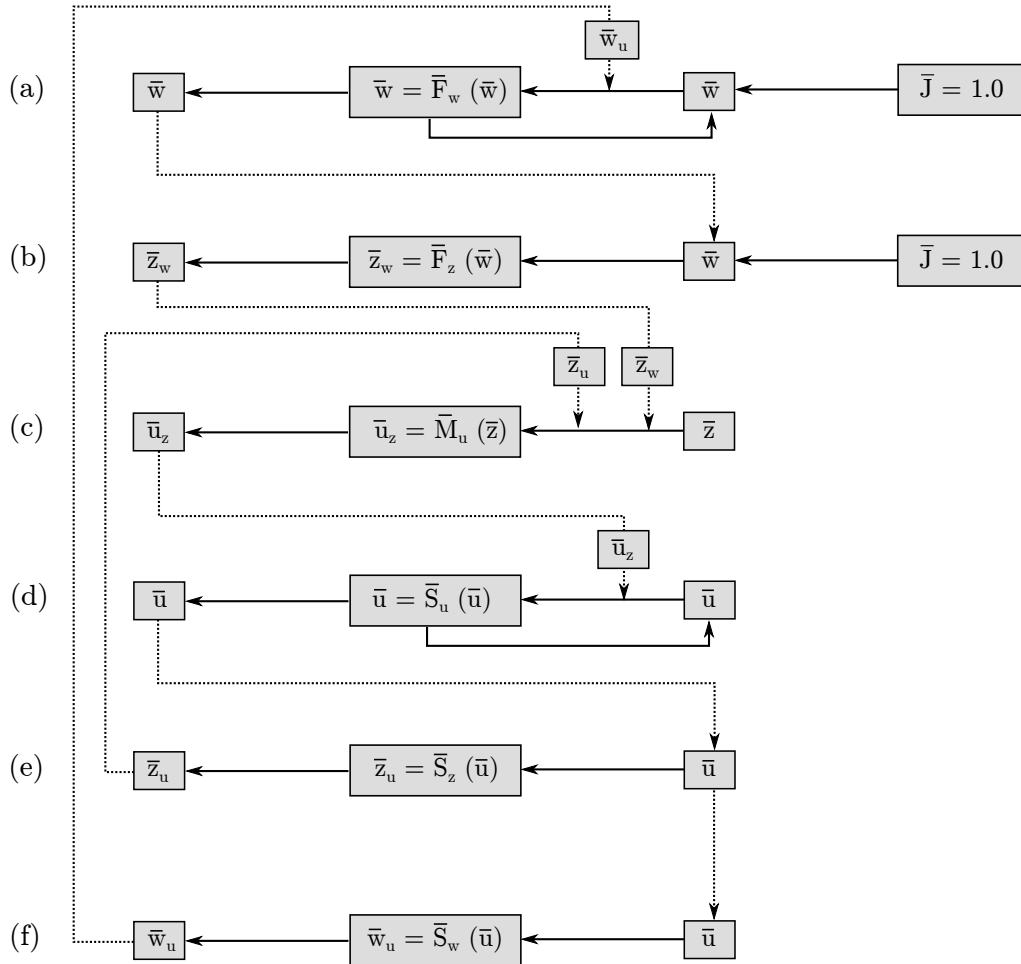


Figure 3.12: Primal and reverse computational paths for the FSI solver, using a fluid-based objective function.

3.5.3 Extension of the Coupled Adjoint Method to time-domain FSI

In this work, the calculation of the adjoint in coupled FSI has been limited to steady-state problems due to time constraints. However, we consider that there is no real limitation in the applicability of the method described in section 3.5.1 for problems involving time-domain phenomena. Although the method has not yet been fully implemented, the equations for its extension to time-domain are presented here, in order to ease further research in dynamic FSI optimisation. Let the problem in (3.64) be redefined for time domain problems using (3.31) and (3.54),

$$\mathbf{x} = \mathbf{G}(\mathbf{x}, \boldsymbol{\alpha}) \Leftrightarrow \begin{cases} \mathbf{q}^t &= \mathbf{S}(\mathbf{q}^t, \mathbf{q}^{t-1}, \mathbf{w}^t, \mathbf{z}^t, \boldsymbol{\alpha}), \\ \mathbf{w}^t &= \mathbf{F}(\mathbf{w}^t, \mathbf{w}^{t-1}, \mathbf{w}^{t-2}, \mathbf{z}^t, \mathbf{z}^{t-1}, \mathbf{z}^{t-2}, \boldsymbol{\alpha}), \\ \mathbf{z}^t &= \mathbf{M}(\mathbf{q}^t, \boldsymbol{\alpha}). \end{cases} \quad (3.71)$$

where the *extended* nodal variable $\mathbf{q}^t = [\mathbf{u}^t, \dot{\mathbf{u}}^t, \ddot{\mathbf{u}}^t]$ has been included to incorporate the effects of the structural velocity and acceleration, and the dependence on $\mathbf{z}^{t-1}$ and $\mathbf{z}^{t-2}$ allows one to compute the velocity of the fluid mesh nodes at time t , $\dot{\mathbf{z}}^t$, using backwards differences. This velocity is necessary for the ALE formulation in (2.11).

The optimisation problem results

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{n} \sum_{k=1}^n \hat{J}(\mathbf{q}^k, \mathbf{w}^k, \mathbf{z}^k, \boldsymbol{\alpha}) \\ \text{subject to} \quad & \mathbf{S}(\mathbf{q}^i, \mathbf{q}^{i-1}, \mathbf{w}^i, \mathbf{z}^i, \boldsymbol{\alpha}) - \mathbf{q}^i = \mathbf{0} & i = 1, \dots, n, \\ & \mathbf{F}(\mathbf{w}^i, \mathbf{w}^{i-1}, \mathbf{w}^{i-2}, \mathbf{z}^i, \mathbf{z}^{i-1}, \mathbf{z}^{i-2}, \boldsymbol{\alpha}) - \mathbf{w}^i = \mathbf{0} & i = 1, \dots, n, \\ & \mathbf{M}(\mathbf{q}^i, \boldsymbol{\alpha}) - \mathbf{z}^i = \mathbf{0} & i = 1, \dots, n, \end{aligned} \quad (3.72)$$

which allows to define the Lagrangian function

$$\begin{aligned} \mathcal{L} = & \frac{1}{n} \sum_{i=1}^n \hat{J}(\mathbf{q}^i, \mathbf{w}^i, \mathbf{z}^i, \boldsymbol{\alpha}) + \sum_{i=1}^n [\bar{\mathbf{q}}^i]^T [\mathbf{S}(\mathbf{q}^i, \mathbf{q}^{i-1}, \mathbf{w}^i, \mathbf{z}^i, \boldsymbol{\alpha}) - \mathbf{q}^i] \\ & + \sum_{i=1}^n [\bar{\mathbf{w}}^i]^T [\mathbf{F}(\mathbf{w}^i, \mathbf{w}^{i-1}, \mathbf{w}^{i-2}, \mathbf{z}^i, \mathbf{z}^{i-1}, \mathbf{z}^{i-2}, \boldsymbol{\alpha}) - \mathbf{w}^i] + \sum_{i=1}^n [\bar{\mathbf{z}}^i]^T [\mathbf{M}(\mathbf{q}^i, \boldsymbol{\alpha}) - \mathbf{z}^i]. \end{aligned} \quad (3.73)$$

Differentiating (3.73) with respect to the structural variables $\mathbf{q}$, and omitting the variables

α at this stage,

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mathbf{q}^n} &= \frac{1}{n} \frac{\partial \hat{J}^n}{\partial \mathbf{q}^n} + \left([\bar{\mathbf{q}}^n]^T \frac{\partial \mathbf{S}^n}{\partial \mathbf{q}^n} \right) - [\bar{\mathbf{q}}^n]^T + \left([\bar{\mathbf{z}}^n]^T \frac{\partial \mathbf{M}^n}{\partial \mathbf{q}^n} \right) = \mathbf{0}, \\
\frac{\partial \mathcal{L}}{\partial \mathbf{q}^{n-1}} &= \frac{1}{n} \frac{\partial \hat{J}^{n-1}}{\partial \mathbf{q}^{n-1}} + \left([\bar{\mathbf{q}}^n]^T \frac{\partial \mathbf{S}^n}{\partial \mathbf{q}^{n-1}} \right) + \left([\bar{\mathbf{q}}^{n-1}]^T \frac{\partial \mathbf{S}^{n-1}}{\partial \mathbf{q}^{n-1}} \right) - [\bar{\mathbf{q}}^{n-1}]^T \\
&\quad + \left([\bar{\mathbf{z}}^{n-1}]^T \frac{\partial \mathbf{M}^{n-1}}{\partial \mathbf{q}^{n-1}} \right) = \mathbf{0}, \\
\frac{\partial \mathcal{L}}{\partial \mathbf{q}^{n-2}} &= \frac{1}{n} \frac{\partial \hat{J}^{n-2}}{\partial \mathbf{q}^{n-2}} + \left([\bar{\mathbf{q}}^{n-1}]^T \frac{\partial \mathbf{S}^{n-1}}{\partial \mathbf{q}^{n-2}} \right) + \left([\bar{\mathbf{q}}^{n-2}]^T \frac{\partial \mathbf{S}^{n-2}}{\partial \mathbf{q}^{n-2}} \right) - [\bar{\mathbf{q}}^{n-2}]^T \\
&\quad + \left([\bar{\mathbf{z}}^{n-2}]^T \frac{\partial \mathbf{M}^{n-2}}{\partial \mathbf{q}^{n-2}} \right) = \mathbf{0}, \\
&\dots
\end{aligned} \tag{3.74}$$

Differentiating now the Lagrangian with respect to the fluid variables, it leads to

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mathbf{w}^n} &= \frac{1}{n} \frac{\partial \hat{J}^n}{\partial \mathbf{w}^n} + \left([\bar{\mathbf{q}}^n]^T \frac{\partial \mathbf{S}^n}{\partial \mathbf{w}^n} \right) + \left([\bar{\mathbf{w}}^n]^T \frac{\partial \mathbf{F}^n}{\partial \mathbf{w}^n} \right) - [\bar{\mathbf{w}}^n]^T = \mathbf{0} \\
\frac{\partial \mathcal{L}}{\partial \mathbf{w}^{n-1}} &= \frac{1}{n} \frac{\partial \hat{J}^{n-1}}{\partial \mathbf{w}^{n-1}} + \left([\bar{\mathbf{q}}^{n-1}]^T \frac{\partial \mathbf{S}^{n-1}}{\partial \mathbf{w}^{n-1}} \right) + \left([\bar{\mathbf{w}}^n]^T \frac{\partial \mathbf{F}^n}{\partial \mathbf{w}^{n-1}} \right) \\
&\quad + \left([\bar{\mathbf{w}}^{n-1}]^T \frac{\partial \mathbf{F}^{n-1}}{\partial \mathbf{w}^{n-1}} \right) - [\bar{\mathbf{w}}^{n-1}]^T = \mathbf{0} \\
\frac{\partial \mathcal{L}}{\partial \mathbf{w}^{n-2}} &= \frac{1}{n} \frac{\partial \hat{J}^{n-2}}{\partial \mathbf{w}^{n-2}} + \left([\bar{\mathbf{q}}^{n-2}]^T \frac{\partial \mathbf{S}^{n-2}}{\partial \mathbf{w}^{n-2}} \right) + \left([\bar{\mathbf{w}}^n]^T \frac{\partial \mathbf{F}^n}{\partial \mathbf{w}^{n-2}} \right) + \left([\bar{\mathbf{w}}^{n-1}]^T \frac{\partial \mathbf{F}^{n-1}}{\partial \mathbf{w}^{n-1}} \right) \\
&\quad + \left([\bar{\mathbf{w}}^{n-2}]^T \frac{\partial \mathbf{F}^{n-2}}{\partial \mathbf{w}^{n-2}} \right) - [\bar{\mathbf{w}}^{n-2}]^T = \mathbf{0} \\
&\dots,
\end{aligned} \tag{3.75}$$

and a very similar expression results when we differentiate with respect to the fluid mesh variables,

$$\begin{aligned}
&\dots \\
\frac{\partial \mathcal{L}}{\partial \mathbf{z}^{n-2}} &= \frac{1}{n} \frac{\partial \hat{J}^{n-2}}{\partial \mathbf{z}^{n-2}} + \left([\bar{\mathbf{q}}^{n-2}]^T \frac{\partial \mathbf{S}^{n-2}}{\partial \mathbf{z}^{n-2}} \right) + \left([\bar{\mathbf{w}}^n]^T \frac{\partial \mathbf{F}^n}{\partial \mathbf{z}^{n-2}} \right) + \left([\bar{\mathbf{w}}^{n-1}]^T \frac{\partial \mathbf{F}^{n-1}}{\partial \mathbf{z}^{n-1}} \right) \\
&\quad + \left([\bar{\mathbf{w}}^{n-2}]^T \frac{\partial \mathbf{F}^{n-2}}{\partial \mathbf{z}^{n-2}} \right) - [\bar{\mathbf{z}}^{n-2}]^T = \mathbf{0} \\
&\dots
\end{aligned} \tag{3.76}$$

Although the previous equations seem convoluted, they actually have a rather straightforward structure behind that builds upon the definitions provided in the previous sections of this chapter. Using a change of variables $t = n - 2$, the previous system of equations can be rewritten in a compact form,

$$\left\{ \begin{array}{l} \bar{\mathbf{q}} = \bar{\mathbf{S}}_t^t(\bar{\mathbf{q}}^t) + \bar{\mathbf{M}}_{\mathbf{q},t}^t(\bar{\mathbf{z}}) + \bar{\mathbf{S}}_t^{t+1}(\bar{\mathbf{q}}^{t+1}), \\ \bar{\mathbf{w}} = \bar{\mathbf{S}}_{\mathbf{w},t}^t(\bar{\mathbf{w}}^t) + \bar{\mathbf{F}}_t^{t+2}(\bar{\mathbf{w}}^{t+2}) + \bar{\mathbf{F}}_t^{t+1}(\bar{\mathbf{w}}^{t+1}) + \bar{\mathbf{F}}_t^t(\bar{\mathbf{w}}^t), \\ \bar{\mathbf{z}} = \bar{\mathbf{S}}_{\mathbf{z},t}^t(\bar{\mathbf{w}}^t) + \bar{\mathbf{F}}_{\mathbf{z},t}^{t+2}(\bar{\mathbf{w}}^{t+2}) + \bar{\mathbf{F}}_{\mathbf{z},t}^{t+1}(\bar{\mathbf{w}}^{t+1}) + \bar{\mathbf{F}}_{\mathbf{z},t}^t(\bar{\mathbf{w}}^t), \end{array} \right. \quad (3.77)$$

where the operator $\bar{\mathbf{F}}_{\mathbf{z},t}^{t+1}(\bar{\mathbf{w}}^{t+1})$ corresponds to the fluid solver, evaluated at time $t + 1$, and differentiated with respect to the mesh variables $\mathbf{z}$ at time t . The differentiation of the time-domain and crossed terms, and their incorporation into the coupled calculation, should not be significantly different from the procedures explained before in this chapter.

Chapter 4

Optimal Electromechanical Actuation using Dielectric Elastomers

In order to test the applicability of the adjoint method developed in section 3.4.1 to complex, non-linear problems, the optimal electromechanical actuation of flexible structures using dielectric elastomers (DEs) will be investigated in this chapter.

Solid materials can be categorised as either conductors or dielectrics, depending on the behaviour they present when they are subject to an electric field. In a conductor, the electrons and ions that form the solid are able to move along large distances. On the other hand, in a dielectric material such particles move relative to each other within the body [190].

Therefore, dielectrics deform under the action of an electric field. They may be further divided into hard dielectrics and soft dielectrics, depending on their deformability in an electric field [191]. Hard dielectrics are those who can only reach strains of under a 10% before the electric effects lead to material failure. On the other hand, soft dielectrics such as DEs may be deformed to strains over 100% by the application of an electric field [192].

The modelling of dielectric elastomers requires the incorporation of the electric effects to the structural equations, as presented in section 4.1. The linearisation of these electric terms is cumbersome, and notably complicates the computation of the adjoint sensitivities using conventional methods. This is shown in section 4.2. The incorporation of electric terms into the AD-based adjoint described in section 3.4.1 is practically immediate, as it will be outlined in section 4.3, demonstrating the potential of the methodology presented in this work.

4.1 Dielectric Elastomer Modelling

The basic behaviour of a DE transducer [190] is described in Fig. 4.1. A very soft dielectric membrane is actuated by means of an electric field. The membrane thickness, t , is reduced as a response to the electric field, which is generated by a voltage applied on two electrodes located on opposing faces of the membrane. This causes an expansion in area in the other two directions. The material of the electrodes is not mechanically relevant in the behaviour of the structure, as they are chosen with a much smaller stiffness. An example of material for the electrode is carbon grease [190].

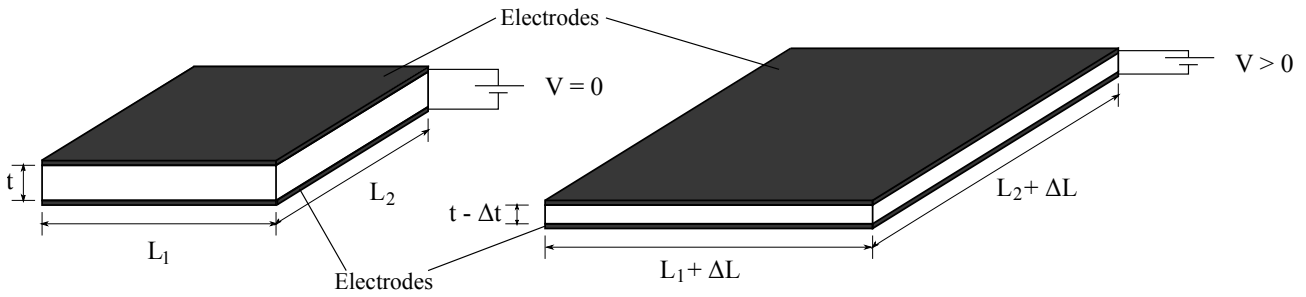


Figure 4.1: Dielectric Elastomer transducer.

These properties make dielectric elastomer transducers to be a very efficient and compliant actuator [20, 21, 190, 193–195]. They have a low weight and cost, while being scalable by removing the mechanical devices required for the actuation. Because of this, DEs are of big interest for some relevant subjects such as microrobotics or bioengineering [194, 195].

In order to model the actuation of flexible structures using dielectric elastomers, in this work we have adopted some simplifying hypothesis. We assume that the material is homogeneous, isotropic and electrically linear, as done by Goulbourne *et. al.* [195]. These assumptions result in a decoupled behaviour between the mechanical and electric response, where the mechanical part of the strain energy function Ψ is purely elastic and the electric part depends only on the electric field. The Cauchy stress tensor $\boldsymbol{\sigma}$ is written as [20, 190, 195]

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}_M + \boldsymbol{\sigma}_E, \quad (4.1a)$$

$$\boldsymbol{\sigma}_E = \epsilon \mathbf{E}_F \otimes \mathbf{E}_F - \frac{1}{2} \epsilon (\mathbf{E}_F \cdot \mathbf{E}_F) \mathbf{I}, \quad (4.1b)$$

where $\boldsymbol{\sigma}_M$ and $\boldsymbol{\sigma}_E$ are respectively the mechanical and the Maxwell stress tensors and $\mathbf{E}_F$ is the electric field in the current configuration. The dielectric constant $\epsilon = \epsilon_0 \epsilon_r$ is a function

of the vacuum dielectric constant ϵ_0 and the material relative constant ϵ_r [20]. In order to obtain the electric field in the current configuration, we define the electric field in the reference configuration $\tilde{\mathbf{E}}$ as done by Zhao and Suo [196]

$$\tilde{\mathbf{E}}_F = \frac{V}{t} \mathbf{n}_{\tilde{\mathbf{E}}}, \quad (4.2)$$

where V is the actuation voltage in the current configuration, t the thickness in the reference configuration and $\mathbf{n}_{\tilde{\mathbf{E}}}$ the unit vector normal to the membrane. The value of $\tilde{\mathbf{E}}$ is pushed forward into the current configuration using [191]

$$\mathbf{E}_F = \mathbf{F}_{\Phi}^{-T} \tilde{\mathbf{E}}_F, \quad (4.3)$$

Most elastomers undergoing large strains are incompressible, that is, their deformation is much larger than their change in volume. However, imposing the incompressibility condition $\Delta v = 0$ may be complex in a finite element framework. Therefore, it is common to impose a condition of “near-incompressibility”, allowing for a small volumetric deformation [147] by splitting the total strain energy function into its isochoric, $\tilde{\Psi}_{\infty}$, and volumetric, $U(J_{\Phi})$ components, as

$$\Psi = \tilde{\Psi}_{\infty}(\mathbf{C}) + U(J_{\Phi}), \quad (4.4)$$

where the volumetric component U is generally defined as

$$U(J_{\Phi}) = c(\kappa)(J_{\Phi} - 1)^2, \quad (4.5)$$

and $c(\kappa)$ is some function of the bulk modulus κ , for which different expressions have been proposed [20, 147, 182]. $c(\kappa)$ is, in general, a very large number to enforce the incompressibility condition, $J = 1$. The isochoric component, $\hat{\Psi}_{\infty}$, depends on the material model adopted. The deformation gradient, $\mathbf{F}$, may be decomposed into the isochoric $\tilde{\mathbf{F}}$ and volumetric $\mathbf{F}_{vol}$ components as [182]

$$\mathbf{F}_{\Phi} = \tilde{\mathbf{F}}_{\Phi} \mathbf{F}_{\Phi, vol}, \quad (4.6a)$$

$$\tilde{\mathbf{F}}_{\Phi} = J_{\Phi}^{-1/3} \mathbf{F}_{\Phi}, \quad (4.6b)$$

$$\mathbf{F}_{\Phi, vol} = J_{\Phi}^{1/3} \mathbf{F}_{\Phi}. \quad (4.6c)$$

Incompressible behaviour gives rise to a numerical problem known as volumetric locking, an artificial stiffening of the system. To avoid it, variational principles such as the three-field Hu-Washizu formulation may be adopted [197]. Although its implementation has not been incorporated in this work, the code has been structured to accommodate further developments on this front if they were deemed necessary.

There are several material models available in the literature for the isochoric component of (4.2). The ideal DE model [190, 191, 198] is widely extended, and we have implemented it in SU2 due to its simplicity. The isochoric stored energy function for this model is, according to Zhao and Suo¹ [196],

$$\hat{\Psi}_{\infty} = \frac{\mu}{2} [F_{iK}F_{iK} - 2\log(J_{\Phi}) - 3]. \quad (4.7)$$

Another commonly-used model for dielectric elastomers is the incompressible, Neo-Hookean material model. However, as pointed out by Suo [190], the Neo-Hookean incompressible model is unable to account for the stiffening effect undergone by membranes made out of DEs when approaching their limiting stretch. In particular, as it is described by Zhao and Wang [191], Neo-Hookean models only expand their area by a 40% before the so-called pull-in instability arises, which contradicts the experimental observations of over a 100% of deformation [192].

This instability is closely related to the process of thinning that occurs when the membrane deforms under an electric load. According to Eq. (4.2), this reduction in thickness also increases the value of the electric field $\tilde{\mathbf{E}}$, until a critical point is reached. It is possible to suppress the pull-in instability if we add the effect of the stiffening before its value is reached.

In order to do so, more sophisticated hyperelastic material models may be used. An extensive review of constitutive models for rubber-like material is given by Steinmann *et al.* [199]. The material models are classified in micro-mechanical and phenomenological. The former are more closely related to the physico-chemical structure of the material, while the latter are formulated either in terms of the principal stretches or the strain invariants.

Among the phenomenological models formulated in terms of the strain invariants, there are some material models that have been shown to be appropriate to model membranes made out of DEs. The first of them is the Gent model [200], which was successfully employed for actuated membrane wings by Buoso and Palacios [20, 21]. Another adequate model for this purpose is the Arruda-Boyce model [201], as pointed out by Zhao *et al.* [198] and also by Suo [190]. The

¹Their implementation in ABAQUS is freely-shared online at: <http://imechanica.org/node/4234>

strain energy functions for these two models are, respectively,

$$\hat{\Psi}_{Gent,\infty} = -J_m \frac{\mu}{2} \ln \left(1 - \frac{\bar{I}_C - 3}{J_m} \right), \quad (4.8a)$$

$$\hat{\Psi}_{A-B,\infty} = \mu \left[\frac{1}{2}(\bar{I}_C - 3) + \frac{1}{20n}(\bar{I}_C^2 - 3^2) + \frac{11}{1050n^2}(\bar{I}_C^3 - 3^3) + \dots \right], \quad (4.8b)$$

Steinmann *et al.* [199] also assert the importance of deriving not only the stress-strain reaction, but also the tangent operator for the material model used, in order to obtain quadratic convergence in the incremental solution of the Newton-Raphson procedure. This task may be particularly complex for the most involved stored energy functions, as for example when adding viscoelasticity into the model. Such rate-dependent effects have been shown to be relevant in some applications of actuated membranes (see, for example, the work by Buoso and Palacios [21]).

The assumptions of homogeneity, isotropy and electrical linearity mean that piezoelectric and electrostrictive effects are neglected [195]. Although there have been further efforts in the literature to account for a fully coupled electromechanical model (see e.g., Park *et al.* [193]) or to include the effects of electrostriction [194], the use of these simplifying assumptions have been found to be adequate for the applications within the scope of this work.

4.2 DE Actuation Sensitivities using Conventional Ad-joints

In this section, to make clear the main limitations of the conventional adjoint methodology presented in section 3.4, it will be applied in the development of an optimal actuation methodology for dielectric elastomers. From (2.36), we can write

$$\mathcal{S}(\mathbf{u}, \mathbf{E}_F) = \mathbf{T}(\mathbf{u}, \mathbf{E}_F) - \mathbf{F}_b - \mathbf{F}_\Gamma(\mathbf{u}) = \mathbf{0}, \quad (4.9)$$

where the internal stress term $\mathbf{T}(\mathbf{u}, \mathbf{E}_F)$ depends on the displacements of the nodes $\mathbf{u}$ and of the electric field $\mathbf{E}_F$.

From (2.18), we know that the internal stress term is

$$\mathbf{T}(\mathbf{u}, \mathbf{E}_F) = \int_v \boldsymbol{\sigma} : \nabla \delta \mathbf{v} \, dv. \quad (4.10)$$

Recalling now that under the assumptions described in section 4.1, the mechanical and the electrical responses are decoupled as shown in Eqn. (4.1a), it is possible to rewrite the internal stress term (4.10) as a superposition of states,

$$\mathbf{T}(\mathbf{u}, \mathbf{E}_F) = \mathbf{T}_M(\mathbf{u}) + \mathbf{T}_E(\mathbf{E}_F) = \int_v \boldsymbol{\sigma}_M : \nabla \delta \mathbf{v} \, dv + \int_v \boldsymbol{\sigma}_E : \nabla \delta \mathbf{v} \, dv. \quad (4.11)$$

Finally, the structural problem (4.9) is rewritten in terms of explicit dependencies only,

$$\mathcal{S}(\mathbf{u}, \mathbf{E}_F) = \mathbf{T}_M(\mathbf{u}) + \mathbf{T}_E(\mathbf{E}_F) - \mathbf{F}_b - \mathbf{F}_\Gamma(\mathbf{u}) = \mathbf{0}. \quad (4.12)$$

The tangent matrix, $\mathbf{K} = \partial \mathcal{S} / \partial \mathbf{u}$ is now assembled recalling that the initial stress term (2.40b) is now the superposition of the mechanical and electrical terms

$$\mathbf{K}_\sigma = \mathbf{K}_{M,\sigma} + \mathbf{K}_{E,\sigma} \quad (4.13a)$$

$$= \int_v \boldsymbol{\sigma}_M : [(\nabla(\Delta \mathbf{u}))^T (\nabla \delta \mathbf{v})] \, dv + \int_v \boldsymbol{\sigma}_E : [(\nabla(\Delta \mathbf{u}))^T (\nabla \delta \mathbf{v})] \, dv, \quad (4.13b)$$

Similarly, the constitutive term is assembled as

$$\mathbf{K}_c = \mathbf{K}_{M,c} + \mathbf{K}_{E,c} \quad (4.14a)$$

$$= \int_v \nabla \delta \mathbf{v} : \mathbf{c}_M : \boldsymbol{\epsilon} \, dv + \int_v \nabla \delta \mathbf{v} : \mathbf{c}_E : \boldsymbol{\epsilon} \, dv, \quad (4.14b)$$

where the spatial elasticity tensor for the electric component, $\mathbf{c}_E$ is computed from

$$\mathbf{c}_{E,ijkl} = J^{-1} \sum_{I,J,K,L} \mathbf{F}_{\Phi,iI} \mathbf{F}_{\Phi,jJ} \mathbf{F}_{\Phi,kK} \mathbf{F}_{\Phi,lL} \mathbf{C}_{E,IJKL}, \quad (4.15a)$$

$$\mathbf{C}_E = \frac{\partial \mathbf{S}_{PK,E}}{\partial \mathbf{E}_M}, \quad (4.15b)$$

$$\mathbf{S}_{PK,E} = J \mathbf{F}_\Phi^{-1} \boldsymbol{\sigma}_E \mathbf{F}_\Phi^{-T}, \quad (4.15c)$$

$$\boldsymbol{\sigma}_E = \epsilon (\mathbf{F}_\Phi^{-T} \tilde{\mathbf{E}}_F) \otimes (\mathbf{F}_\Phi^{-T} \tilde{\mathbf{E}}_F) - \frac{1}{2} \epsilon \left[(\mathbf{F}_\Phi^{-T} \tilde{\mathbf{E}}_F) \cdot (\mathbf{F}_\Phi^{-T} \tilde{\mathbf{E}}_F) \right] \mathbf{I}, \quad (4.15d)$$

and $\mathbf{E}$ is the Green strain tensor as described in (2.13).

The term $c_{E,ijkl}$ has the physical meaning of the variation of the electric stress with the strain. In practical terms, an increment in the deformation $\Delta \mathbf{u}$ modifies the value of the deformation gradient $\mathbf{F}$, thus affecting the push-forward operation on the reference electric field defined in (4.3). This variation is expected to be small, but it has not been previously derived in the literature to the best of our knowledge.

A possible reason for not having tackled this problem so far is that the exact tangent matrix $\mathbf{K}_{\mathcal{J}}$ is not needed for most applications, as it is sufficient as long as $\mathbf{K}_{\mathcal{J}}^*$ is an approximation good enough to converge the residual $\mathcal{S} = \mathbf{T} - \mathbf{F}$ to $\mathbf{0}$. Given that the derivation of $c_{E,ijkl}$ is convoluted, in this work it has been omitted. That means that only an approximation to the Jacobian is available, $\hat{\mathbf{K}}_{\mathcal{J},NL,E}$, and as a result the adjoint variable obtained from

$$\hat{\mathbf{K}}_{\mathcal{J},NL,E}^T \hat{\mathbf{u}} = -\frac{\partial J^T}{\partial \mathbf{u}}, \quad (4.16)$$

is no longer the solution of the problem

$$\mathbf{K}_{\mathcal{J},NL,E}^T \tilde{\mathbf{u}} = -\frac{\partial J^T}{\partial \mathbf{u}}. \quad (4.17)$$

The implications of this simplification will be discussed in section 6.1.1.

4.3 DE Sensitivities using AD-based Adjoints

Let the problem in (4.12) be rewritten as

$$\mathbf{u}^{n+1} = \mathbf{u}^n - [\mathbf{K}_{\mathcal{J}}(\mathbf{u}^n, \mathbf{E}_F)]^{-1} \mathcal{S}(\mathbf{u}^n, \mathbf{E}_F) =: \mathbf{S}(\mathbf{u}^n, \mathbf{E}_F), \quad (4.18)$$

and let the Lagrangian associated to the problem be

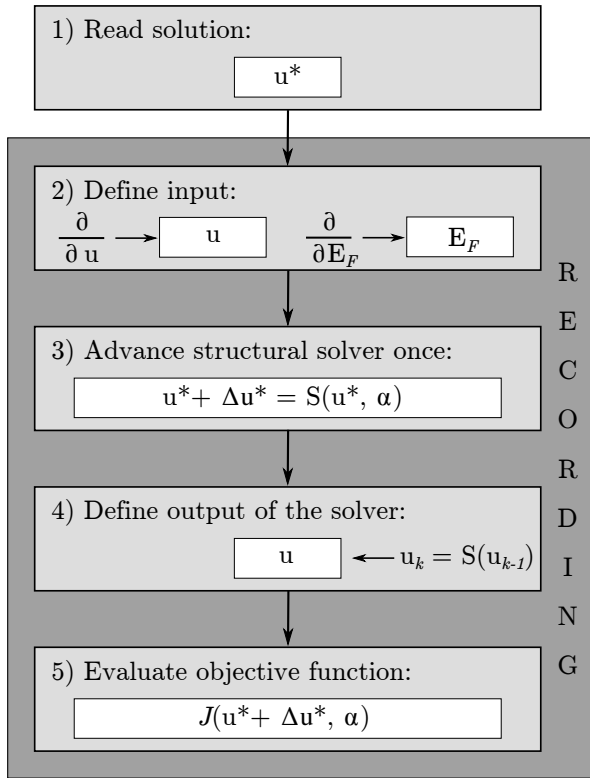
$$\mathcal{L}(\mathbf{u}, \bar{\mathbf{u}}, \mathbf{E}_F) = J(\mathbf{u}) + \bar{\mathbf{u}}^T [\mathbf{S}(\mathbf{u}, \mathbf{E}_F) - \mathbf{u}], \quad (4.19)$$

where the electric field $\mathbf{E}_F$ has been incorporated as a design parameter and $J(\mathbf{u})$ is defined in (3.40). The gradient $dJ/d\mathbf{E}_F$ results in

$$\frac{dJ}{d\mathbf{E}_F} = \bar{\mathbf{u}}^T \frac{\partial \mathbf{S}}{\partial \mathbf{E}_F}. \quad (4.20)$$

From (4.20), it can be observed that it is enough to define $\mathbf{E}_F$ as an *input* of the problem during the recording phase in order to obtain the gradients with respect to the electric field. This is shown in Fig. 4.2, and demonstrates the flexibility of this approach. As it was explained in section 3.4.1, the accuracy of $\mathbf{K}_{\mathcal{S},NL,E}$ is only relevant to the convergence of the primal problem, as the reverse problem inherits its convergence properties.

a) Primal Solver Path



b) Reverse Solver Path

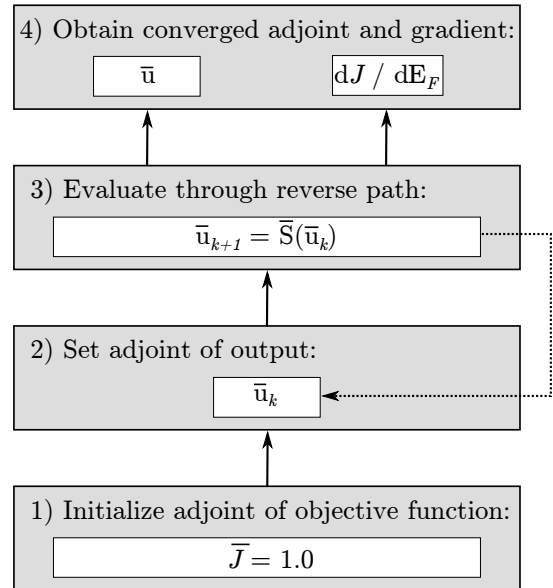


Figure 4.2: Primal (a) and reverse (b) computational paths for the solver $\mathbf{S}$ with electric actuation.

Chapter 5

Coupled Analysis of FSI Problems With Large Deformations

At the start of this project, SU2 was limited to the solution of fluid problems using a Finite-Volume solver. During the first stage of this work, a structural solver was natively in SU2 and coupled it with the fluid solver in the suite, in order to be able to solve coupled FSI problems in static and dynamic conditions. The theory behind these implementations was covered in chapter 2. Now, four test cases are presented to demonstrate the validity of the structural solver developed for coupled FSI problems involving large deformations. From simpler to more complex, the implementation of the linear finite elements using a linear elastic problem is verified first in section 5.1.1. The structural solver is tested for problems with large deformations and material non-linearities in section 5.1.2. Then, a membrane-wing problem is used to evaluate the applicability of the solver in steady-state FSI problems, in section 5.2.1. Finally, the accuracy of the coupled FSI solver is demonstrated for time-domain problems in section 5.2.2.

5.1 Finite Element Solver Validation and Verification

In this section, the structural solver that has been implemented in SU2 during this project is tested, via direct comparison with the commercial solver ABAQUS. The implementation of the solid finite elements is analysed in section 5.1.1 using a linear elastic clamped-clamped beam. Next, the solver is tested in section 5.1.2 for very flexible cantilevers in large-deformations regime.

5.1.1 Linear Solver

In order to verify the implementation of the Finite Element method for the structural solver, a clamped-clamped beam under distributed load was defined. The geometry of the problem is described in Figure 5.1, where $L = 6$ m, $H = 0.5$ m and $B = 0.2$ m. It was tested both using a 2D and a 3D model, under linear assumptions. Regarding the material properties, the elasticity modulus is $E = 3 \cdot 10^9$ Pa, the Poisson's ratio is $\nu = 0.3$, and the structural density is $\rho_s = 7854$ kg/m³.

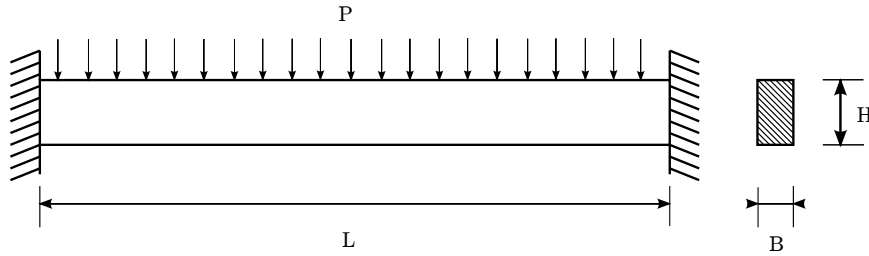


Figure 5.1: Beam validation case

First, a two-dimensional, static plane stress model has been analysed, by comparing the solution obtained using SU2 to the commercial code ABAQUS [202]. A structured mesh was generated with exactly the same finite element discretisation in both solvers. The static load condition is a distributed load, $P = 1$ kPa, as shown in Fig. 5.1.

The vertical displacement for both solvers is shown in figure 5.2. A convergence study of the relative error between solvers is shown in figure 5.3, demonstrating a good agreement. It may be observed that, as the mesh is refined in both axis and the number of elements increases, both codes converge to the same solution with a negligible error, with a rate of convergence roughly linear with the number of elements. The small discrepancies for the coarser meshes are likely due to the use of linear shape functions in SU2 as compared to quadratic elements in ABAQUS.

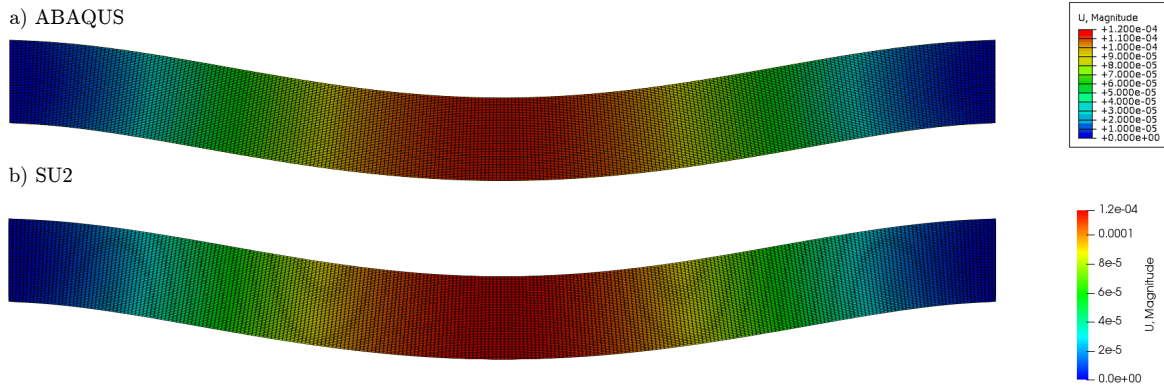


Figure 5.2: Nodal displacement magnitude. Results from ABAQUS (a) and SU2 (b), for 13311 elements. Displacements scaled 3000 times.

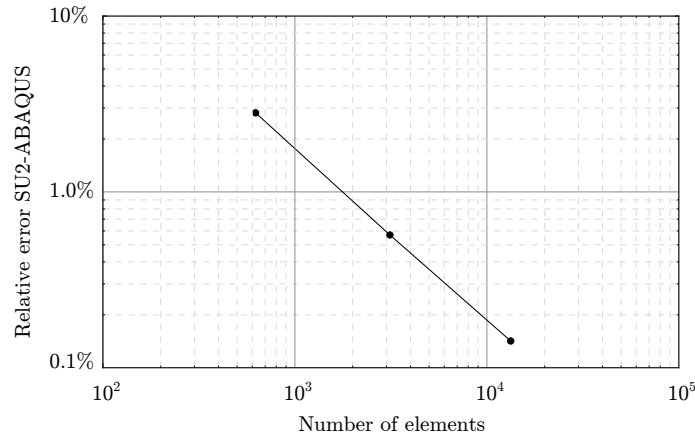


Figure 5.3: Nodal displacement magnitude, convergence study. Relative error in the maximum displacements between SU2 and ABAQUS.

Two-dimensional and three-dimensional models of the beam have been built in order to validate the element formulation for different kinds of elements. For the 2D case, 3-node (trias) and 4-node (quads) elements have been analysed, both in terms of the structured and unstructured meshes. Using quads, the difference between using a structured or an unstructured grid is almost negligible, remaining within a 1% error. The results vary when using trias, either structured or unstructured, in an order of magnitude of a 3-5%. The discrepancies between trias and quads are caused by the different number of Gaussian points used in the integration of the stiffness matrix: while the quad elements are integrated using 4 Gaussian points, the trias have been implemented with a single Gaussian point per element. Nevertheless, the results demonstrated to be physically accurate, and also the implementation of the formulation was shown to be adequate.

5.1.2 Non-linear Solver for Large Displacements

In this work, we are interested in the solution of coupled FSI problems involving non-linear structural behaviour such as large displacements and complex material behaviour. Several problems that will be tackled in the present thesis involve the analysis of very flexible cantilevers under bending conditions, for which finite elements using linear interpolation are known to suffer from detrimental artificial stiffening due to shear locking. The original finite volume nature of the SU2 solver has so far limited the implementation of quadratic finite elements. Although there are ongoing efforts to implement higher-order finite elements for fluid problems, which are expected to be extensible to structural problems, this feature is not yet available at the time of writing.

As a result, it has been considered necessary to carry out an assessment on the applicability of the current implementation of the non-linear structural solver to flexible cantilevers. The test case proposed is shown in Fig. 5.4. It involves the solution of a flexible cantilever under a body force, $\mathbf{F}_b$ (Fig. 5.4,a), and also under a *follower*, uniform pressure, $\mathbf{F}_r(\mathbf{u})$ (Fig. 5.4,b). The cantilever has been studied for a range of different values of stiffness, E , between 13 to 28 kPa, and four discretisations as shown in Table 5.1.

Table 5.1: Mesh discretisations

Test case	Number of elements	Elements in thickness	Elements in length
Coarse	360	6	60
Medium	1200	10	120
Fine	4800	20	240
Finest	16000	40	400

It is observed from Fig. 5.4 that the most flexible cases undergo very large displacements, which require a geometrically non-linear solver. We have adopted the hyperelastic Neo-Hookean material model, in order to incorporate material non-linearities to the problem. The displacement obtained at the point Z as defined in Fig. 5.4 is compared to the results obtained using the commercial solver ABAQUS with quadratic elements and the *finest* discretisation.

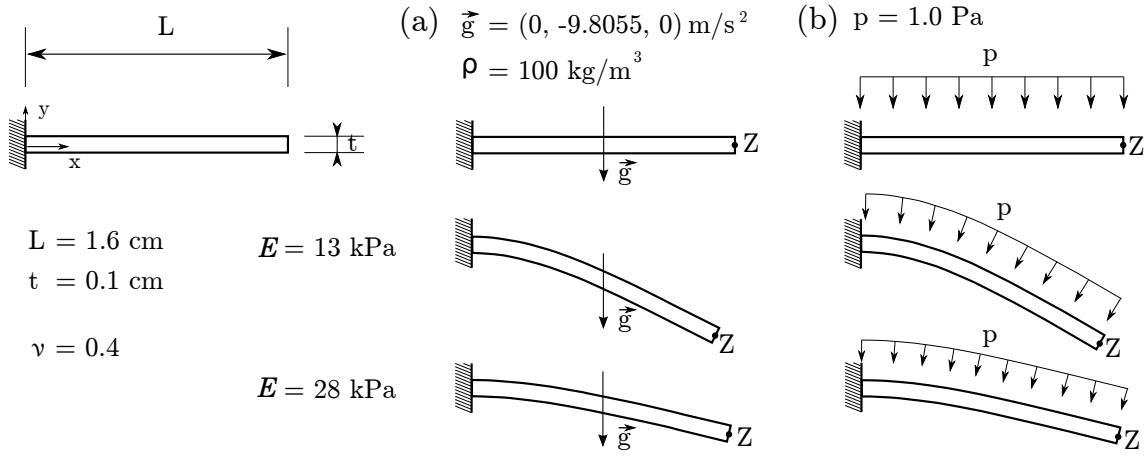
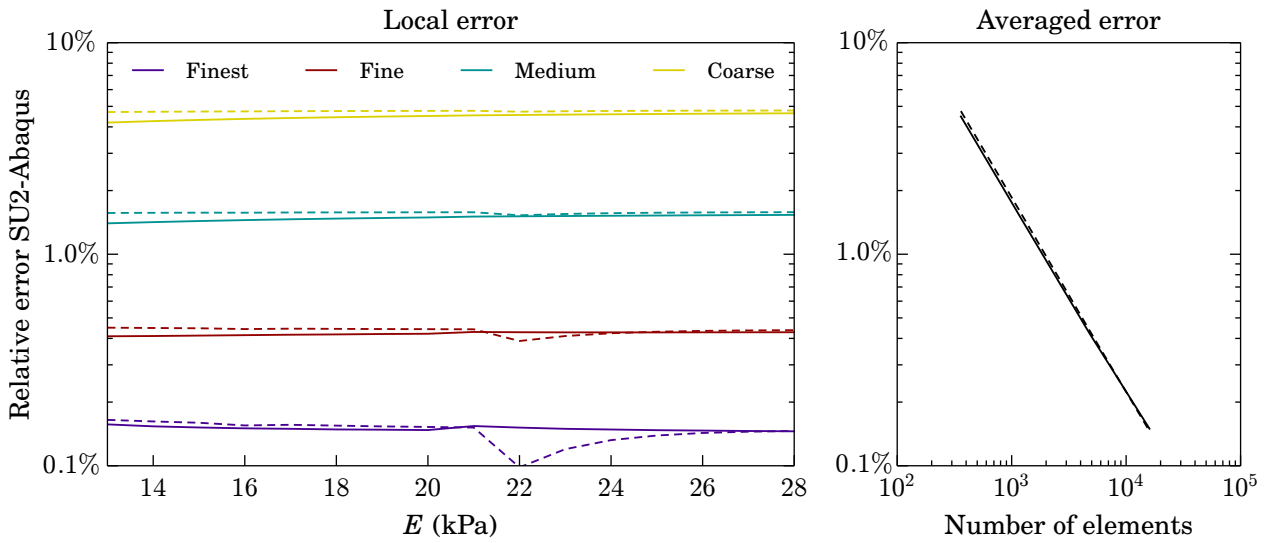


Figure 5.4: Convergence study definition for non-linear structural problems.

Figure 5.5: Nodal displacement magnitude, convergence study for non-linear problems. Relative error in the maximum displacements between SU2 and ABAQUS. The *continuous* lines correspond to the dead load, while the *dashed* lines correspond to the uniform, follower pressure.

It is shown in Fig. 5.5 that the *coarse* discretisation in Tab. 5.1 is unable to accurately solve the problem. The error for that discretisation ranges between 4-5%, which responds to the effect of the shear locking due to the linear element discretisation. For the same number of elements, the quadratic FEA solver in Abaqus only yields a 0.14% error with respect to the *finest*.

Again as in section 5.1.1, the rate of convergence is roughly linear with the number of elements, as presented in Fig. 5.4. Errors in the order of 1.5% are obtained for the *medium* discretisation, while they are further reduced to 0.5% for the *fine* discretisation. An error of the order of 0.15% is obtained for the *finest* discretisation using linear elements, showing that the implementation of the solver is adequate, and that it is applicable to problems involving large displacements when the discretisation of the structural domain is refined enough.

5.2 Coupled FSI Solver Validation and Verification

The implementation of the coupled solver in SU2 includes both a loosely-coupled, Conventional Serial Staggered (CSS) time coupling and a strongly-coupled, Block Gauss-Seidel method with relaxation, using a fixed or a dynamic Aitken's parameter, as explained in section 2.3.1. The spatial coupling has been limited to a common discretisation of the interface for fluid and structure, although the infrastructure for interpolation has been put in place. The structural solver is able to run either linear analysis or non-linear analysis both in terms of geometry and material properties, as presented in section 5.1. In order to address the verification and validation of the fluid solver with an ALE formulation, the reader may refer to section V.A.5 in the paper by Palacios *et al.* [125]. To investigate the implementation of the FSI problem, in this chapter we will present two different test cases. First, we study the behaviour of a membrane wing in a steady-state reattached condition [20, 203]. This test case was presented in the 2017 AIAA SciTech conference [142]. Next, we solve the dynamic FSI problem of a flexible beam attached on the downwind side of a rigid square. This problem has been used extensively as benchmark for Fluid-Structure Interaction applications [61, 67, 68, 153, 204–206] and was presented in Refs. [140, 141].

5.2.1 Membrane-Wing in Reattached Flow

First, we verify the implementation of the FSI solver on steady-state conditions. We solve the problem of a passive membrane mounted on two rigid supports, which is adapted from the work of Gordnier [203] and Buoso and Palacios [20]. These authors numerically simulated the experimental setup of Rojratsirikul *et al.* [207], although for a lower Reynolds number. The geometry of the problem, and the physical properties of the fluid and structural domains, are shown in Fig. 5.6.

We adopt the non-dimensional parameters that describe the Fluid-Structure Interaction from problem Buoso and Palacios [20]: $Et/(\rho_f u_f^2 c) = 50$, and $\rho_s t/(\rho_f c) = 0.589$. The discretisation of the fluid mesh is done using 160 nodes on each surface of the membrane and 76 nodes around the supports. The boundary layer mesh refinement has a total thickness of 1.5 mm discretised using 9 elements and a growth ratio of 1.06. The membrane wing is modelled using linear, solid

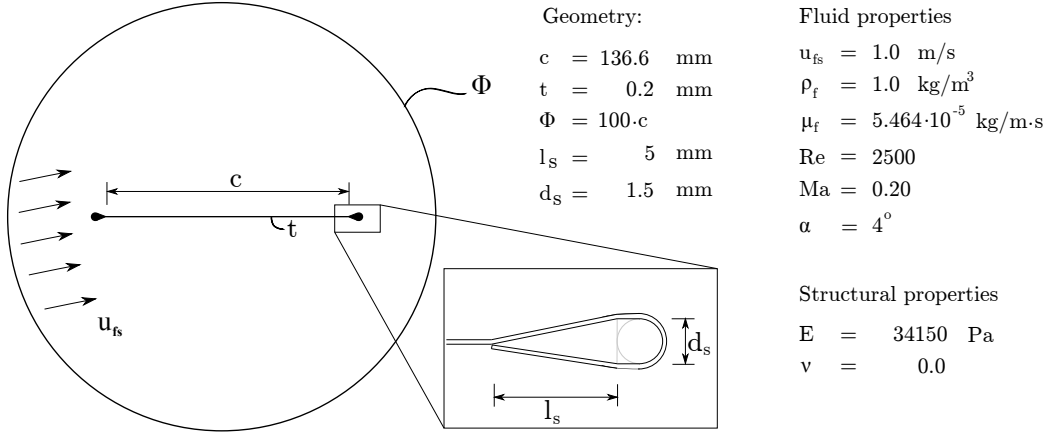


Figure 5.6: Membrane wing test case (not to scale) [142].

quadrilateral elements, matching with the fluid nodes in the interface and with 2 elements in the thickness direction. The membrane is modelled using a compressible Neo-Hookean hyperelastic material model, and a plane stress assumption. Finally, we neglect the Poisson effect, $\nu = 0.0$, in order to compare our results with the membrane model used by Gordnier [203]. Consequently, the thickness remains constant even when the membrane is stretched.

While this mesh is rather coarse compared to those in the literature, it has proven to be sufficient for the solution of a reattached laminar flow condition [203] with an angle of attack of 4° . The problem converges to a steady-state solution with no massive separation, which allows to use a lighter discretisation as compared to the work of Gordnier [203], where test cases involving large separation introducing dynamics to the problem are also studied.

The strongly-coupled, Block Gauss-Seidel algorithm converges the maximum displacement at the interface to a 1/20000th of the membrane thickness in under 100 iterations. We adopted a fixed relaxation parameter $\omega = 0.6$. The load is transferred using a cubic curve during the first $N = 50$ BGS subiterations, $\gamma^n = -2(n/N)^3 + 3(n/N)^2$. This prevents a large force imbalance in the r.h.s of the structural equation, while guaranteeing that the quality of the mesh remains adequate.

The deformation of the membrane is shown in Fig. 5.7 and compared with the results presented by Gordnier [203] and Buoso and Palacios [20]. The final, deformed position of the membrane agrees well with the results presented in the literature. As it has been explained throughout this thesis, the structural solver that has been implemented only allows for solid elements to be used. Therefore, on the one hand, clamped boundary conditions are adopted, which as shown in Fig. 5.7 affects the deflection in the areas immediately next to them. This behaviour propagates and the maximum camber is reduced. This model has a higher fidelity

than a pinned model, as it more closely resembles the physical problem.

On the other hand, pinned boundary conditions have also been considered to compare with the solution presented by Gordnier [203] using a membrane model. We enforce this boundary condition by physically reducing the moment of inertia at the supports. It is shown in Fig. 5.7 that the results with this modelling technique accurately match those obtained by Gordnier [203] and Buoso and Palacios [20], demonstrating the accuracy of the scheme. A plot of the velocity magnitude which shows the flow features and the deformation of the membrane is presented in Fig. 5.8.

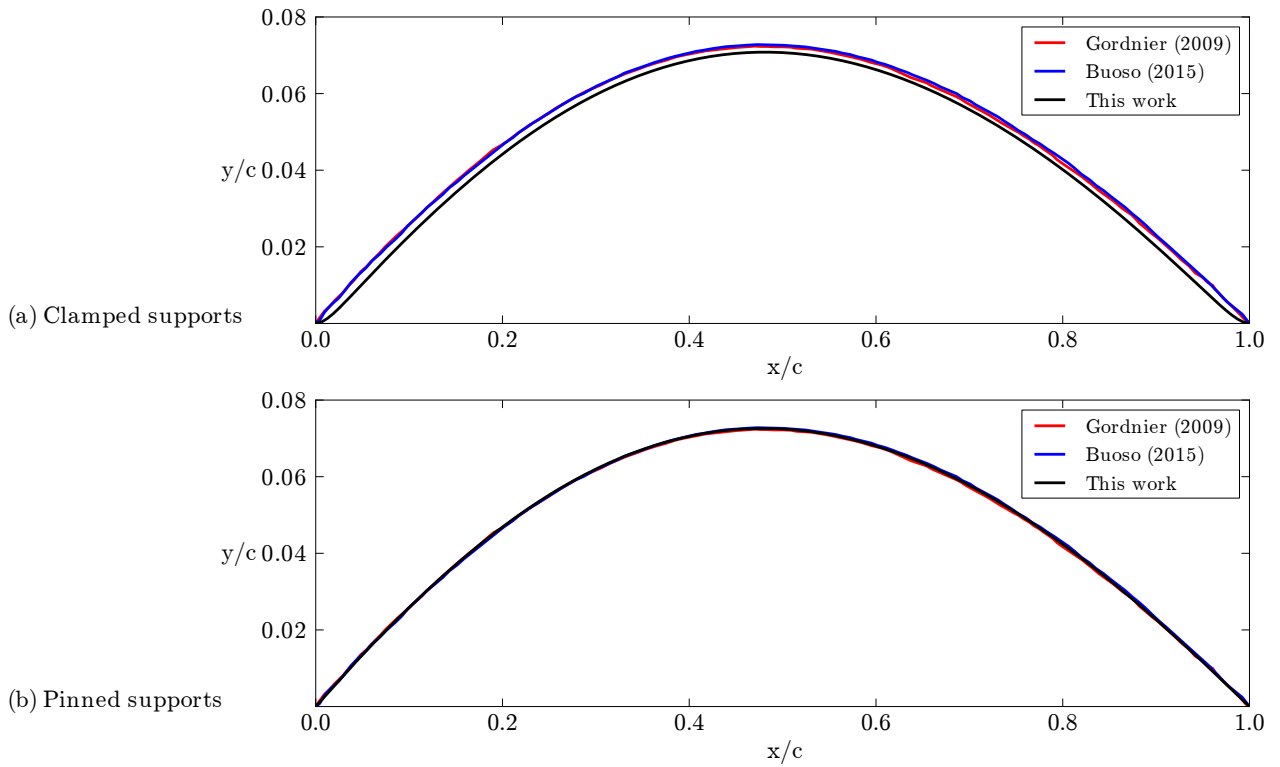


Figure 5.7: Midline surface deflection of the membrane for an angle of attack of 4° [142].

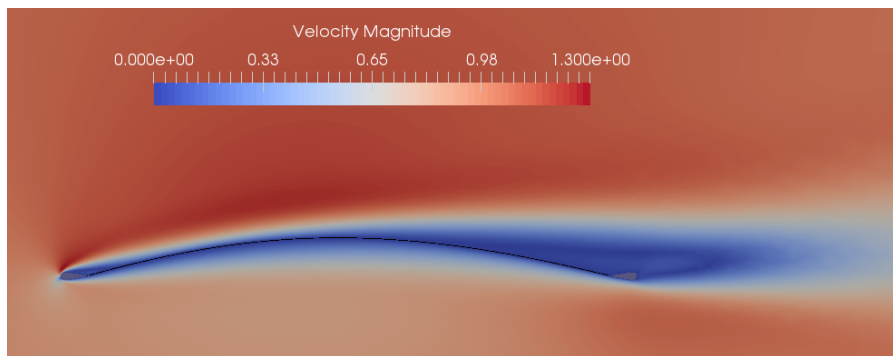


Figure 5.8: Velocity fields and membrane deformation.

5.2.2 Flexible Cantilever in Vortical Flow

We now test the implementation of the FSI solver on dynamic conditions. The problem of study was originally proposed in 1998 by Wall and Ramm, [204], and has since then been used as benchmark for numerous FSI implementations. The geometrical and physical description of the case is shown in Fig. 5.9 and Tab. 5.2. The problem is defined in 2D, and consists of a square cylinder immersed in a low-Reynolds number flow ($Re = 332$). The cylinder sheds vortices, which generate areas of low-pressure on its wake. A flexible cantilever is attached on the downwind side of the square, and receives an alternated force generated by the vortex street, therefore suffering from vortex-induced vibrations. For a linear model, the first natural bending mode of the structure corresponds to 3.03 Hz. The boundary conditions are set to inlet on the upwind side, outlet on the downwind side, and slip walls on the upper and lower boundaries. The square and the appendix are set to no-slip walls.

Two parameters that are extensively used in the literature as a validation criterion are the frequency and maximum amplitude of the vertical displacements at the tip of the cantilever. The reported frequencies range normally from 3.0 to 3.2 Hz, while the tip displacements range from 1.0 to 1.35 cm. A third criterion will be used in this work, which is the modulation of the response by higher frequency effects. Some authors, as for example Dettmer and Perić [68] or Froehle and Persson [206], have presented time histories that show the presence of multiple frequencies in the stationary response. As Dettmer and Perić already highlighted and as it will be seen here, convergence analysis show the rather complex nature of the dynamics of this configuration.

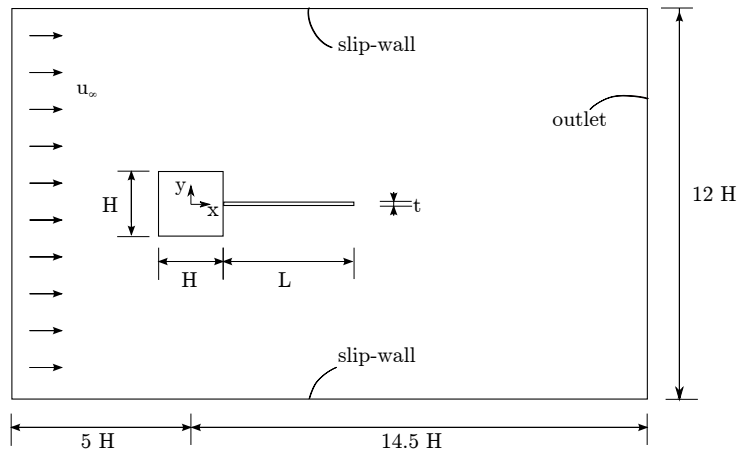
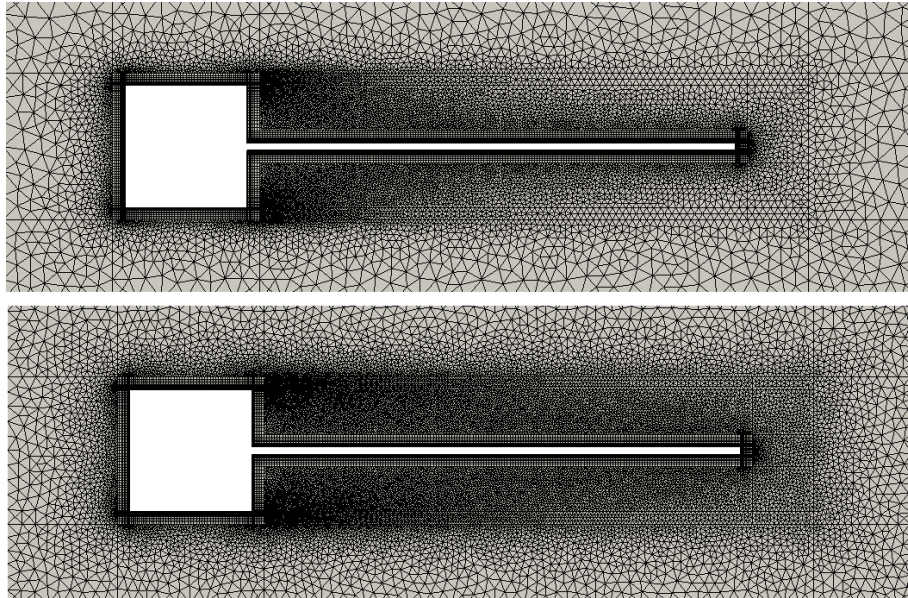


Figure 5.9: Fluid-Structure Interaction test case (not to scale) [140].

Table 5.2: Problem definition

Geometry					
H	1	cm			
L	4	cm			
t	0.06	cm			
Fluid			Structure		
P_{inlet}	5.7022	Pa	E	$2.5 \cdot 10^5$	Pa
P_{fs}	5.5453	Pa	ν	0.35	
ρ_f	1.18	kg/m ³	ρ_s	100	kg/m ³
μ_f	$1.82 \cdot 10^{-5}$	kg/m·s			
T	0.0164	K			
Re	332				
Ma	0.2				

Two different meshes have been used for the validation of the FSI solver. They will be again named *coarse* and *fine*. They combine triangular elements and boundary layer refinement to be able to capture the separation due to the bluff bodies at the tested Reynolds number. The *coarse* mesh has 18778 nodes, for 27216 of the *fine*. The cantilever is discretised using a linear FE mesh with 1128-quadrilaterals and 6 elements in the thickness. The meshes were generated using the open-source code Gmsh [208], and for the region closer to the body they are shown in Fig. 5.10.

Figure 5.10: Fluid discretisations: *coarse* (top) and *fine* (bottom) meshes for the FSI benchmark (Adapted from [140]).

The coupled problem is solved using a BGS approach with dynamic Aitken's relaxation. The convergence criteria on the interface is set to 1/1000 of the cantilever thickness, which

generally requires between 4 to 5 iterations to converge. A staggered approach was found to diverge for a time step $\Delta t = 0.0075$ s. Two subiterations of the BGS method, however, were enough to maintain the stability of the simulation for all the tested cases, as it was also reported by Wood *et al.* [205]. The structure is integrated using a Newmark time integration procedure, incorporating a 2% numerical damping in order to prevent the divergence of the solver due to high frequency structural effects. The coupled solution is started from a slightly perturbed, symmetric fluid state, previously computed with a static beam in order to avoid non-physical transient effects due to the initialisation. In terms of time accuracy, both domain discretisations have been run using two different time discretisations, with $\Delta t = 0.0075$ s as the reference time step and $\Delta t = 0.005$ s for an improved solution. The time histories of the vertical tip displacements and the Fourier transform analysis are shown in Fig. 5.11 for all four configurations.

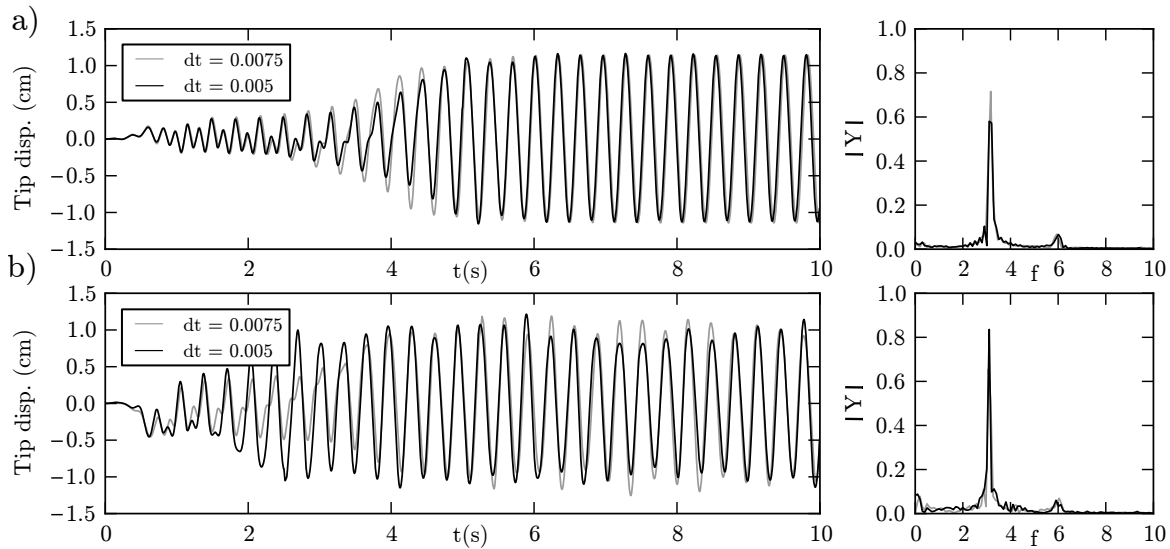


Figure 5.11: Time histories of the vertical displacements at the tip of the cantilever and FFT plots, for the a) *coarse*, b) *fine* and c) *finest* fluid discretisations, and the two time steps considered, $\Delta t = 0.0075$ s and $\Delta t = 0.005$ s [140].

It can be observed in Fig. 5.11a that the *coarse* mesh has been converged with the time step. The time histories for $\Delta t = 0.0075$ s and $\Delta t = 0.005$ s almost overlap. The frequency and amplitude measured match quite well for both cases, and no higher frequency effects are clearly appreciated in any of the time discretisations. It can also be observed that the beam starts to vibrate with a higher frequency, which according to the FFT plot is estimated to be around 6 Hz. After 5 seconds of simulation, the cantilever has reached a stationary state of vibration with an average frequency $\bar{f} = 3.15$ Hz, with a maximum amplitude $d_{max} = 1.15$ cm. These results agree well with most of the results in the literature [61, 68, 153, 204–206]. It is noted, however, that the vibration only occurs at one frequency once the stationary state has been reached; no

higher frequency effects are observed, which contradicts the observations made by some other authors [68, 204, 206]

However, for the *fine* discretisation, this is no longer true, see Fig. 5.11b. In this case, a higher frequency wave clearly modulates the amplitude of the vibration. This behavior matches the observations of Dettmer and Perić [68] regarding the appearance of high frequency effects for dense spatial discretisations. For $\Delta t = 0.0075$ s, the mean frequency measured is calculated to be $\bar{f} = 3.10$ Hz, ranging from 3.04 to 3.19 Hz, and with a maximum amplitude $d_{max} = 1.25$ cm. For $\Delta t = 0.005$ s, the mean frequency measured is $\bar{f} = 3.11$ Hz, ranging from 2.96 to 3.22 Hz, and the maximum amplitude $d_{max} = 1.21$ cm. Two peaks are observed at the FFT plot; the first peak is similar to the first mode of vibration of the appendix, while the second corresponds to 6 Hz of frequency, similar to the vortex shedding frequency of the square. The stationary state reached is equivalent for both time steps, so the solution is fully converged. A contour plot of the pressure field and the structural displacements is presented in Fig. 5.12 for both discretisations. Some vortical structures are pointed out, as they are remarkably more defined for the *fine* discretisation (Fig. 5.12b) respect to the coarse discretisation (Fig. 5.12a). For the *fine* mesh, some minor discrepancies observed between the two time steps are believed to have been caused by a fault in the restart routines of the code, not known at the time of running the simulations. The global behaviour, however, remained consistent when the simulations were re-run for the evaluation of the Aitken's dynamic relaxation parameter, that will be presented later in Fig. 5.13, and for which this fault had already been fixed.

We consider to be particularly relevant the fact that for every FFT plot of those shown in Fig. 5.11 there are two peaks, which correspond to the frequencies $f_1 \approx 3$ Hz and $f_2 \approx 6$ Hz. Even the coarsest test case shows some higher frequency effects during the transient period at the beginning of the simulation, which in that particular case are later damped out.

It is common when dealing with Vortex-Induced Vibrations (VIV) generated by bluff bodies to use the non-dimensional Strouhal number $St = fH/U$ to characterize the frequency of the response [209–211]. For this test case with $H = 1$ cm and $U = 51.3$ cm/s, the Strouhal numbers for the two frequencies f_1 and f_2 under consideration are $St_1 = 0.058$ and $St_2 = 0.117$.

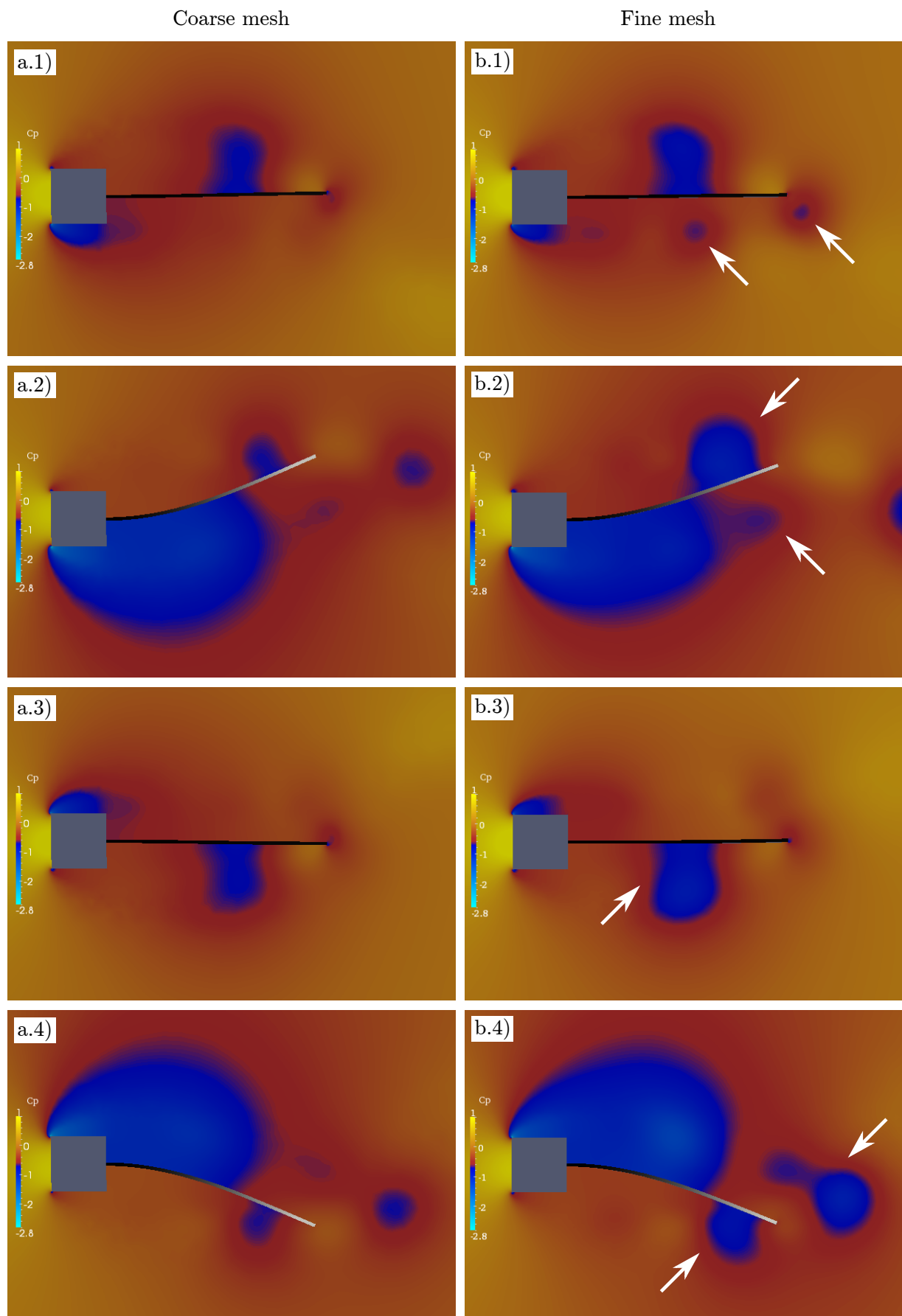


Figure 5.12: Pressure contours and structural displacements for a cycle of vibration for the *coarse* mesh (a) and for the *fine* mesh (b). $T=0$ (1), $T=\pi/2$ (2), $T=\pi$ (3), $T=3\pi/2$ (4). The arrow in (b) points out some vortical structures that are damped out by the coarser discretisation.

Hübner *et al.* [212] have solved a problem with very similar settings to those presented in Fig. 5.9. Their test case differs from the one tackled here in the structural settings, but also in the inflow velocity used, which is set to be 31.5 cm/s. They solve a simulation with a rigid structure, i.e., with no structural deformation, and a refined $\Delta t = 0.001$ s. Interestingly, they report that for the rigid case the square cylinder sheds alternate vortices on both sides that lead to a periodic behavior and a frequency $f = 3.7$ Hz. That would result on an alternate forcing acting over the structure at the frequency of the shedding. The Strouhal number of this case, with $H = 1$ cm, $U = 31.5$ cm/s, and $f = 3.7$ Hz, is $St = 0.117$. This value agrees with our computed value of St_2 . Moreover, a Fourier analysis of the forces acting over the cantilever when this is held rigid for the three meshes presented in this work results in peaks ranging from 5.15 to 5.5 Hz and their harmonics, thus a value of St ranging from 0.100 to 0.107.

The results obtained in this section are summarized in Table 5.3, and as it may be observed they agree very well with the literature. We conclude that the higher frequency effects reported in several papers and that result in a modulation of the response of the cantilever are due to the vortex shedding frequency of the square cylinder. In order to capture these vortex shedding effects, it is necessary to refine the fluid domain up to a level of discretisation that prevents the vortical structures from damping out.

Table 5.3: Summary of results and convergence study

	Average frequency (Hz)	Max tip displ. (cm)	High freq. effects	Strouhal number
Wall and Ramm [204]	3.08	1.31	Yes	0.06
Matthies and Steindorf [153]	2.99	1.34	No	0.058
Dettmer and Peric [68]	2.96 - 3.31	1.1 - 1.4	Yes	0.058 - 0.065
Wood <i>et al.</i> [205]	2.78 - 3.125	1.1 - 1.2	No	0.054 - 0.061
Kassiotis <i>et al.</i> [61]	3.17	1.0	No	0.062
Habchi <i>et al.</i> [67]	3.25	1.02	-	0.063
Froehle and Persson [206]	3.18	1.12	Yes	0.062
This work, <i>coarse</i> mesh	3.15	1.15	No	0.061
This work, <i>fine</i> mesh	3.10-3.11	1.21-1.25	Yes	0.060

With the objective of further testing our implementation, we have investigated the ability of the Aitken's dynamic parameter and the first order displacement predictor to reduce the number of BGS subiterations. In Fig. 5.13, we compare the time histories of the vertical displacement at the tip obtained using different relaxation parameters against a BGS strongly-coupled strategy in which we use a fixed relaxation parameter $\omega_{fixed} = 0.5$. This approach converges slowly to

the solution, in about 8-10 iterations per time step.

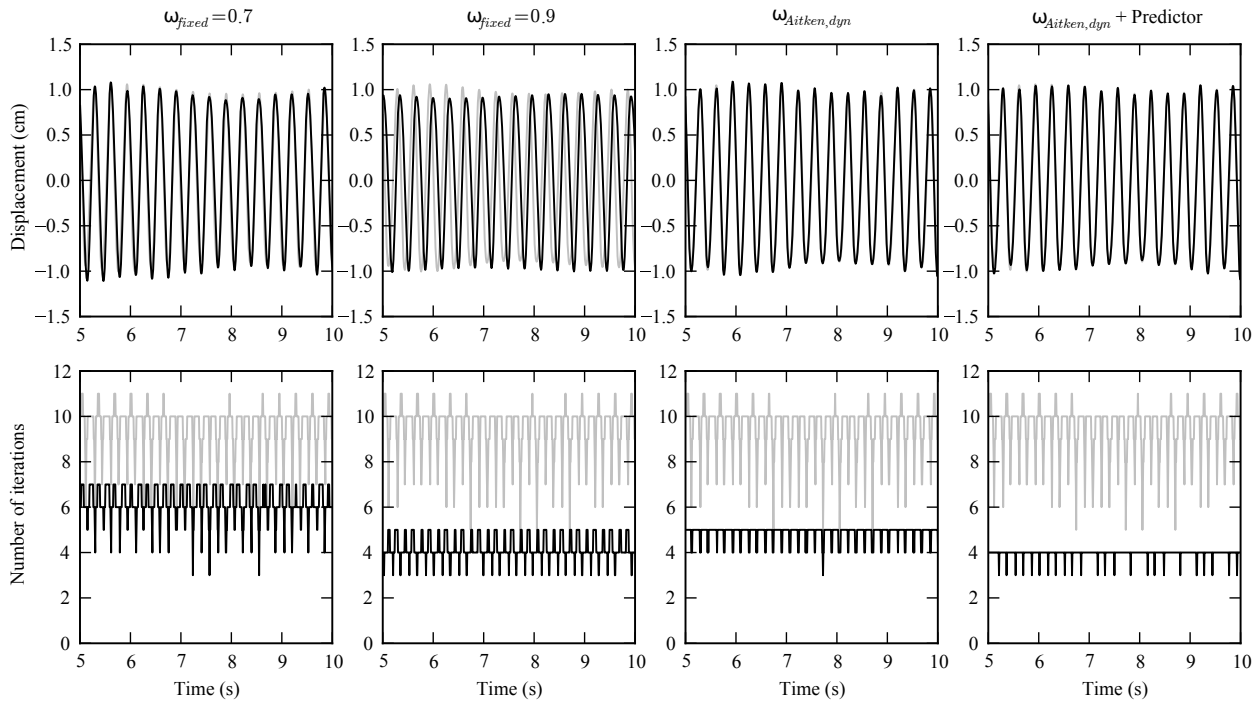


Figure 5.13: Time histories of the vertical tip displacements for different values of the relaxation parameter for the BGS FSI solver. The base solution (in grey) is obtained with $\omega_{fixed} = 0.5$, and no predictor [143].

By increasing the fixed relaxation parameter to $\omega_{fixed} = 0.7$, we reduce the computational time by 33%. However, as it may be seen in Figure 5.13, a higher relaxation parameter introduces some deviations in the solution, that we believe are closely related to the value of ω being too large in the first subiteration, ω^0 . This effect becomes more clear when we increase the fixed relaxation parameter to $\omega_{fixed} = 0.9$, when both the frequency and the amplitude are affected and the amplitude modulation is almost damped out.

Finally, the use of the Aitken's dynamic relaxation parameter ($\omega_{Aitken,dyn}$) clearly improves the convergence of the scheme. In particular, the number of BGS subiterations is reduced to 4-5 per time step, resulting in the computational time being a 38% shorter. By using a first order predictor on the first BGS subiteration, we can further reduce the number of iterations to 3-4 and the computational time by a 56% with respect to the baseline case, while obtaining effectively the same solution. It is important to note that, in these two cases, we have limited the value of the relaxation parameter in the first BGS iteration to $\omega^0 = 0.5$, then allowing it to adapt dynamically in the remaining subiterations.

Chapter 6

Verification of the Sensitivities

Computed with the AD-Based Adjoint Method

The second stage of this project involved the extension of the AD-based adjoint solver that was available in SU2 for fluid problems to structural and coupled FSI problems, as described in chapter 3. In this chapter, the verification of the sensitivities obtained using this technique is presented. They will be compared to the sensitivities obtained using the forward mode of AD, which represent the *exact* gradient obtained to the level of accuracy of the solver. The adjoint sensitivities will also be compared with the values obtained using finite differences, and with the gradients obtained using the conventional adjoint when these are available.

Section 6.1 demonstrates the applicability of the structural adjoint method described in section 3.4 to problems involving geometrical and material non-linearities under body loads and follower forces. Next, section 6.2 covers the sensitivity verification of the novel coupled FSI adjoint method previously described in section 3.5.1. The applicability of the adjoint to a problem involving a flexible, vertical wall under the action of a horizontal viscous flow is studied, and its use for the membrane wing problem from section 5.2.1 under electromechanical actuation is also tested.

6.1 Flexible Structures In Vacuum

First, the adjoint sensitivities of flexible structures in vacuum, that is, with no fluid forces involved, will be studied. Section 6.1.1 covers the applicability of the AD-based Discrete Adjoint (ADDA) methodology to static problems involving a dead load and a follower pressure. Using the stiffness of the structure E as a design variable, we aim to recover the undeformed configuration for the problem presented in section 5.1.2. The advantages of the ADDA technique with respect to conventional adjoints are discussed, and the accuracy of the method is demonstrated.

In section 6.1.2, the applicability of ADDA to problems involving electromechanical actuation is evaluated. The membrane wing in section 5.2.1 is prestretched, and then relaxed using an electric field to recover the un-prestretched equilibrium position. Finally, in section 6.1.3 the applicability of the method described in section 3.4.2 is tested for dynamic problems. The flexible cantilever in section 5.2.2 is displaced from its equilibrium position and then released. A non-linear damper at the tip will be responsible to damp out the vibrations of the cantilever and return it to its reference configuration. The adjoint will use the stiffness E and the mass ρ as design variables and study the gradients of a function that minimizes the vibrations at the tip of the cantilever, therefore leading to a quicker damping.

6.1.1 Static FEA

In order to study the accuracy of the implementation of the ADDA method for structural problems, let the objective function be defined as in (3.40),

$$J(\mathbf{u}) = (\mathbf{u} - \mathbf{u}_{tgt})^T \mathbf{C}(\mathbf{u} - \mathbf{u}_{tgt}). \quad (6.1)$$

We adopt the problem solved in section 5.1.2, and define the target geometry $\mathbf{u}_{tgt}$ as the undeformed configuration. This is shown in Fig. 6.1. The design variable is, in this case, the stiffness of the cantilever, E . We cover the same range of stiffness as in section 5.1.2, that is, $E \in [13, 28]$ kPa. The material model is neo-Hookean, and the Poisson ratio is $\nu = 0.4$.

First, we solve the problem for a uniform body load (Fig. 6.1,a). The objective function J is computed for the previous range of stiffness, in increments of $h = 250$ Pa. Therefore, it is possible to compute the gradients dJ/dE using central differences, (3.3).

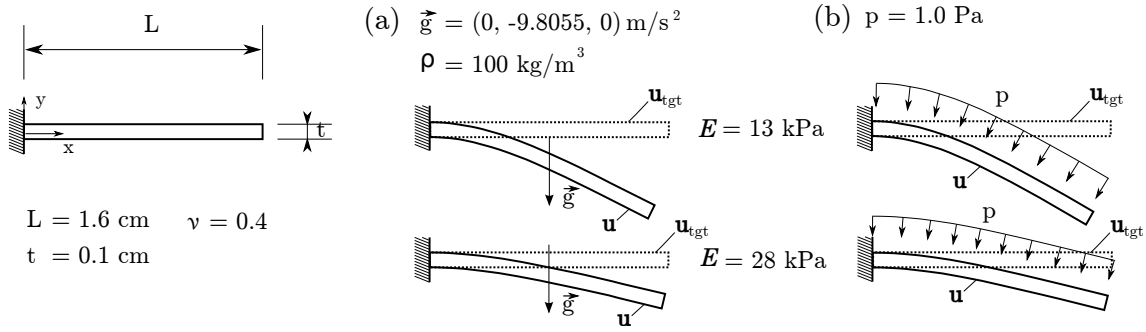


Figure 6.1: Test cases for the computation of the static adjoint sensitivity in flexible structures.

As explained in section 3.4, the body forces $\mathbf{F}_b$ in (3.43) remain independent of the deformation $\mathbf{u}$. Therefore, the linearisation of the structural problem $\mathcal{S}(\mathbf{u})$ yields a symmetric tangent matrix $\mathbf{K}_{\mathcal{S}}$, which can be directly used to solve the adjoint problem,

$$\mathbf{K}_{\mathcal{S}} \tilde{\mathbf{u}} = -\frac{\partial J^T}{\partial \mathbf{u}}. \quad (6.2)$$

In this section, we have computed the sensitivities using the adjoint variable $\tilde{\mathbf{u}}$, and will refer to them as the *Conventional Adjoint* sensitivities.

The value that is taken as the reference, or *exact* sensitivity, is the value computed using the forward mode of AD. This technique was described in section 3.2.1, and is able to propagate the gradient of a single input throughout the full solution process (in this case, the stiffness modulus E). Therefore, it yields a value of the sensitivity that is accurate to the level of accuracy of the solver.

Finally, we solve the adjoint problem using the method described in section 3.4.1. The values of the objective function and the sensitivity computed using the forward mode are presented in Fig. 6.2, together with the relative error with respect to the forward mode sensitivity obtained using finite differences, the conventional adjoint and the ADDA method.

From Fig. 6.2, it can be observed that for $2h=500$ Pa, the central difference scheme yields sensitivities that are accurate to a level of relative error between 0.01 and 0.04 %, which would normally be acceptable in most engineering applications of the gradient. The conventional adjoint yields an almost exact representation of the sensitivity, with a negligible error in the order of $1\text{E-}10$. The gradient obtained with the ADDA method is even more accurate than the conventional adjoint. The error obtained with respect to the forward mode is in the order of $1\text{E-}12$.

Next, we solve the problem for a follower, uniform pressure (Fig. 6.1,b). For this case, the

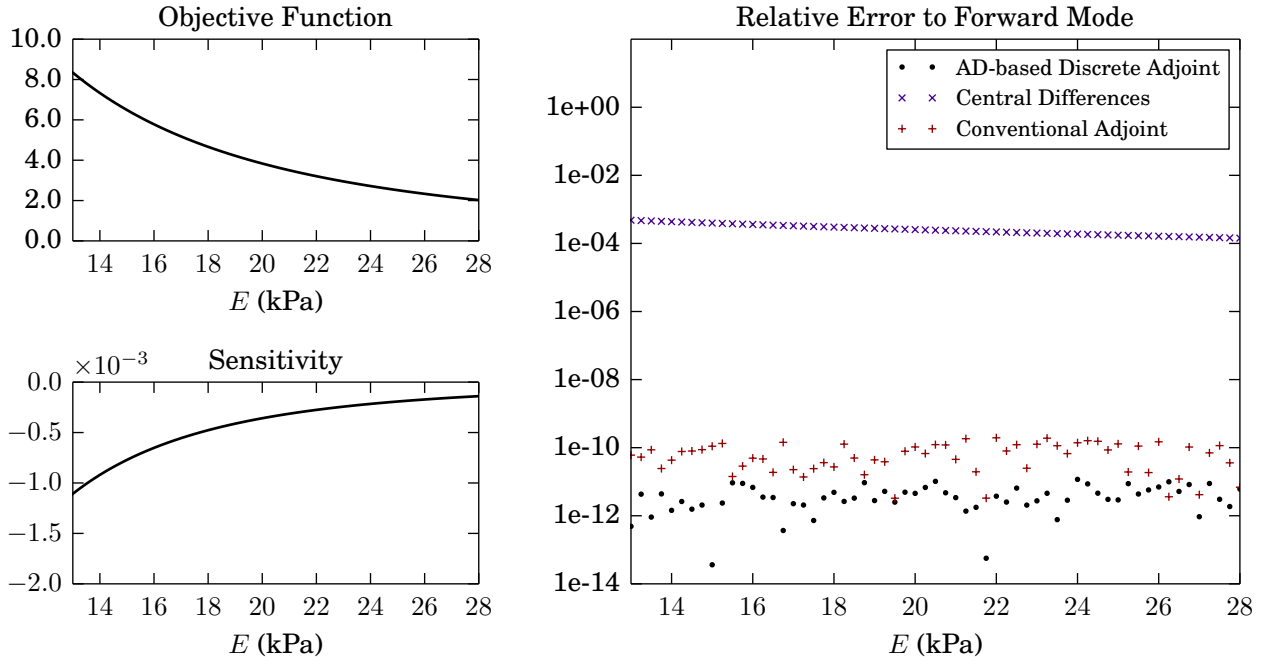


Figure 6.2: Objective function (top left), forward mode sensitivity (bottom left) and relative error with respect to Forward Mode of the AD-based Discrete Adjoint, Central Differences ($2h = 500$) and Conventional Adjoint, for an uniform body load. Adapted from [141].

surface forces are a function of the structural displacements, $\mathbf{F}_\Gamma(\mathbf{u})$. Therefore, the Jacobian of the problem is no longer symmetric, as explained in section 2.2.3.

For the convergence of the primal problem, we use an approximate Jacobian matrix, $\hat{\mathbf{K}}_\mathcal{J}$, where the term $\mathbf{K}_{ext}$ in (2.39) is neglected. We deliberately use the same approximate matrix to solve the conventional adjoint, which is now approximated using

$$\hat{\mathbf{K}}_\mathcal{J} \hat{\mathbf{u}} = -\frac{\partial J^T}{\partial \mathbf{u}}, \quad (6.3)$$

in order to assess the effects of using an approximate tangent matrix for the non-AD adjoint calculation. Although it is relatively simple in this case to obtain the exact tangent matrix, this may not always be the case, as exemplified in chapter 4. The gradients for this test case are presented in Fig. 6.3, along with the values of the objective function and the relative errors of the central differences scheme, the ADDA method and the (approximate) conventional adjoint.

From Fig. 6.3, it can be observed that using an approximate tangent matrix yields errors in the conventional adjoint within the order of the 3-7%, when compared to the solution of the forward mode. This fact demonstrates the large impact that a non-exact Jacobian has over the calculation of the adjoint sensitivity, and assesses the effects of neglecting the term $\partial \mathbf{F}_\Gamma(\mathbf{u}, \mathbf{w}, \mathbf{z}) / \partial \mathbf{u}$ in the construction of the structural tangent matrix for FSI problems. The

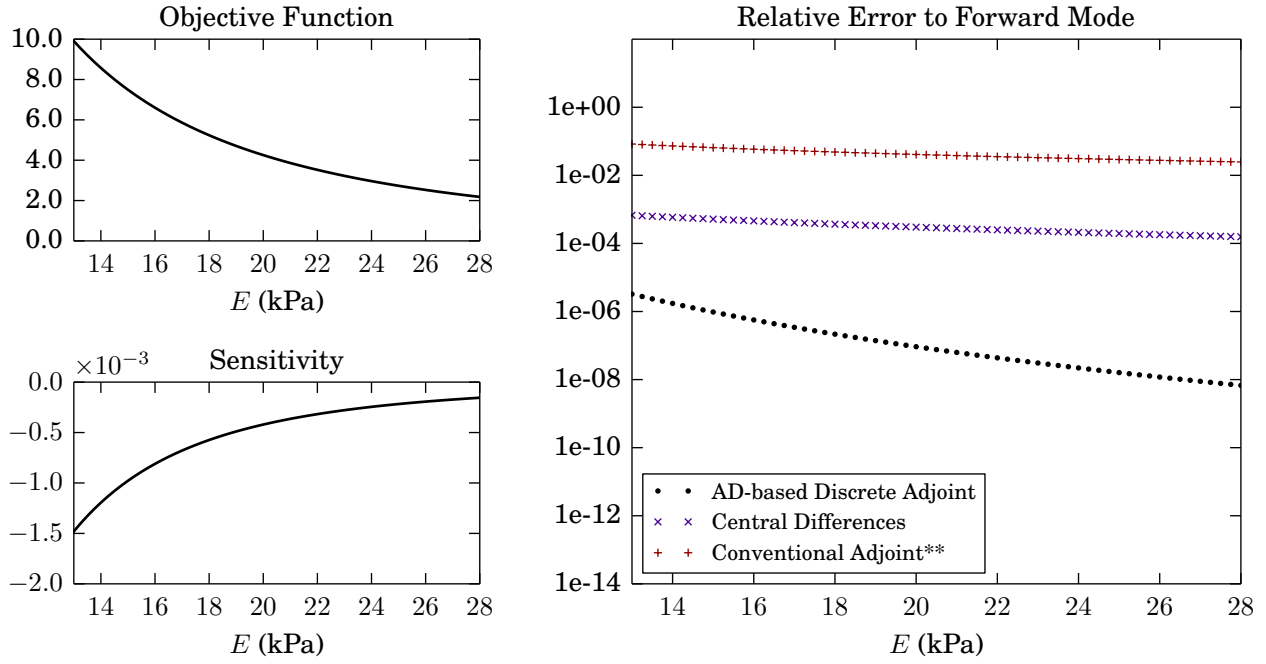


Figure 6.3: Objective function (top left), forward mode sensitivity (bottom left) and relative error with respect to Forward Mode of the AD-based Discrete Adjoint, Central Differences ($2h = 500$) and Conventional Adjoint (**indicates it is obtained using an approximate tangent matrix), for a follower, uniform pressure. Adapted from [141].

value of $\mathbf{F}_r(\mathbf{u}, \mathbf{w}, \mathbf{z})$ is normally not explicitly available, and its linearisation is complex and normally not required for the solution of the physical problem. It is shown, however, that for very flexible cases it can produce a large error in the computation of gradients.

On the other hand, the ADDA method yields relative errors that generally remain below 10^{-6} , which demonstrate its accuracy even when the primal solver does not employ the exact value of $\mathbf{K}_{\mathcal{S}}$. It can be noticed that there is indeed a certain variability on the accuracy of the scheme with the value of E , which is about two orders of magnitude more accurate for the stiffer cases. One likely explanation for this behaviour is that all cases were run to the same convergence criteria and a maximum number of iterations. When the adjoint is recorded, one additional simulation is run (see section 3.4.3), and the new value of $\mathbf{u}^* + \Delta\mathbf{u}^*$ is used to evaluate the objective function, J . This value of $\Delta\mathbf{u}^*$ is likely to be larger in the most flexible cases.

Finally, the accuracy of the central difference scheme remains practically unchanged with respect to the previous case, as expected given that the method of computation of the sensitivities is not affected by the previous considerations.

6.1.2 Actuated FEA

We are now interested in evaluating the applicability of the ADDA method to electrically actuated DEs, as described in chapter 4. We choose the geometry and properties of the clamped membrane wing presented in section 5.2.1. In line with the previous section, two loading states are applied: Test case (a) applies an uniform, dead load to the membrane ($\rho = 200 \text{ kg/m}^3$), and test case (b) applies a uniform, follower pressure $p = 0.4 \text{ Pa}$. The loads were chosen to obtain a maximum camber similar to the solution of the membrane wing test case in section 5.2.1, as shown in Fig. 6.4.

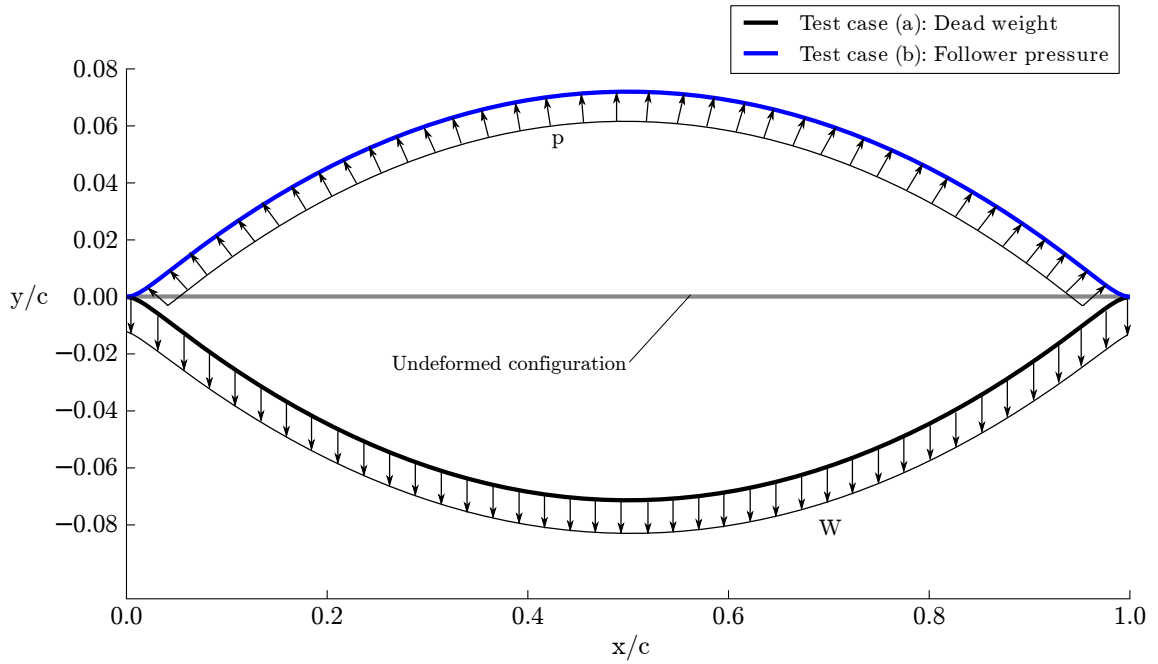


Figure 6.4: Mean deflection of the membrane for a dead weight W , $\rho = 200 \text{ kg/m}^3$, and for a (follower) pressure $p = 0.4 \text{ Pa}$. Adapted from [142].

The stress introduced by an electric field $\mathbf{E}_F$ in the membrane is only able to relax it, as described in (4.1b). In order to actuate the membrane within a certain range of deformation, a state of pretension is introduced in the membrane by applying a prestretch. Five levels of prestretching, ϵ_0 , are used in this section, namely, 1%, 2%, 5%, 10% and 50%. They are able to increasingly stiffen the membrane. For an unactuated membrane, the equilibrium position under the different levels of prestretch is shown in Fig. 6.5.

The target geometry, $\mathbf{u}_{tgt}$, is chosen as the geometry of the unactuated and not-prestretched membrane for each load condition, as shown using a dotted line in Fig. 6.5. An increasing electric field $\mathbf{E}_F$ is able to augment the compliance of the membrane, as shown in Fig. 6.6 for

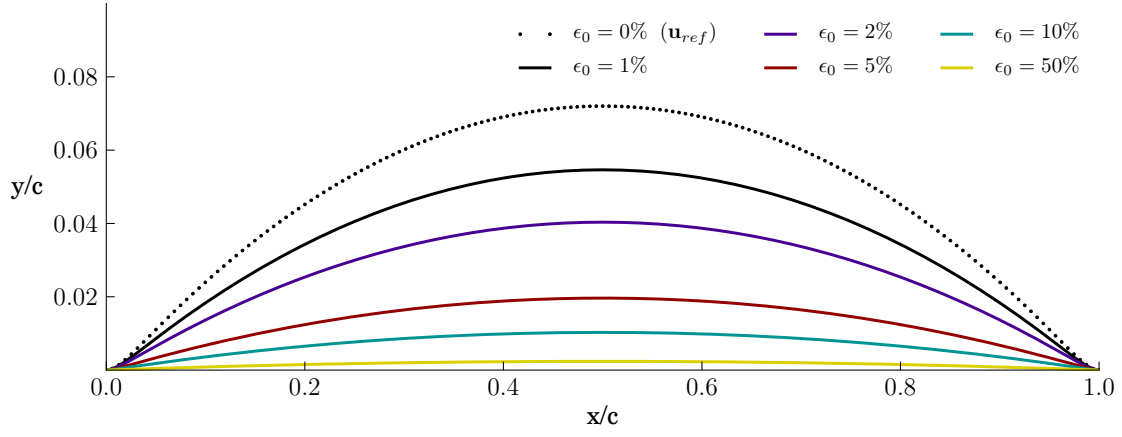


Figure 6.5: Mean deflection of the membrane for a follower pressure $p = 0.4$ Pa under different levels of prestretch, ϵ_0 . Adapted from [142].

a prestretch $\epsilon_0 = 5\%$.

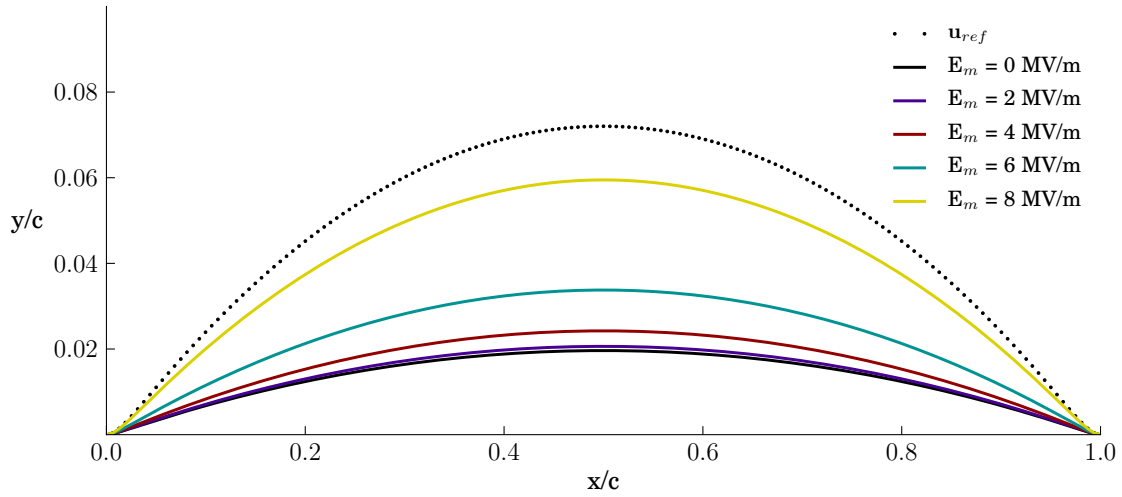


Figure 6.6: Mean deflection of the membrane in vacuum with a prestretch $\epsilon_0 = 5\%$ under different levels of electromechanical actuation. E_F is the modulus of the electric field, V/t. Adapted from [142].

From Fig. 6.6, it can be observed that the response is highly non-linear with the voltage V . An electric field of magnitude $E_F = \|\mathbf{E}_F\| = 4.0$ MV/m barely affects the membrane, while the equilibrium state remains close to the unactuated state. Doubling the actuation to $E_F = 8.0$ MV/m almost trebles the maximum camber. For this particular case, a value of $E_F = 8.6$ MV/m is required to retrieve the target geometry $\mathbf{u}_{tgt}$. This value corresponds to a voltage of 1.72kV for a dielectric constant $\epsilon_r = 4.8$, adopted from the work of Buoso and Palacios [20].

The values of the objective function in (6.1), $\mathbf{C} = 10^6 \mathbf{I}$, are presented in Fig. 6.7, along with the sensitivities with respect to the modulus of the electric field, E_F , obtained using the forward mode of AD. The dashed lines correspond to the values of the uniform pressure, while the continuous lines correspond to the values obtained for the dead weight.

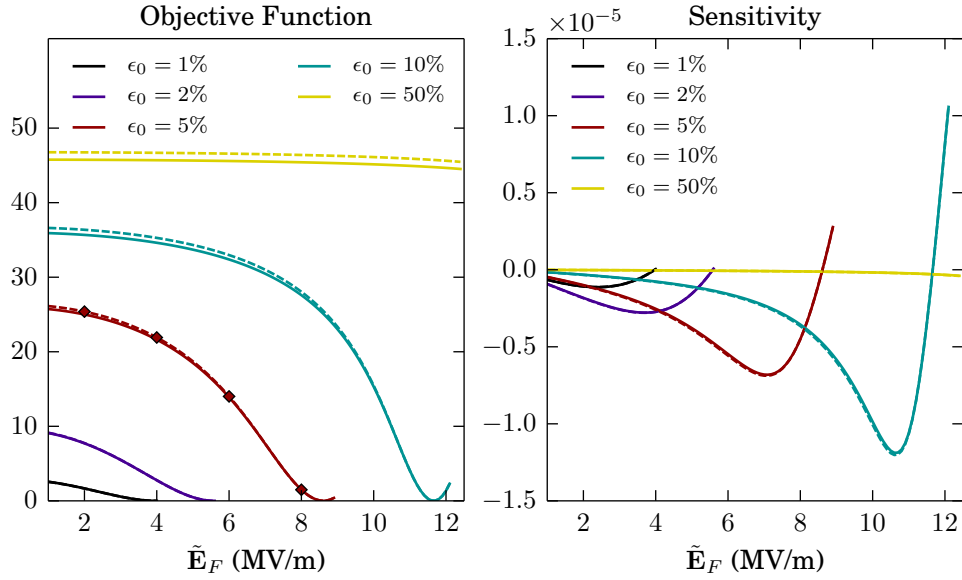


Figure 6.7: Objective function and sensitivities for different prestretch ϵ_0 and actuation conditions, E_F . The values of J marked with a diamond correspond to the geometries presented in Fig. 6.6. Adapted from [142].

First, we compute the sensitivities using the conventional discrete adjoint method, using an approximate Jacobian where the effects of the follower forces and the spatial elasticity tensor for the electric component, c_E in (4.15a), are neglected. The relative error for the five different magnitudes of prestretch is shown in Fig. 6.8, for both test cases.

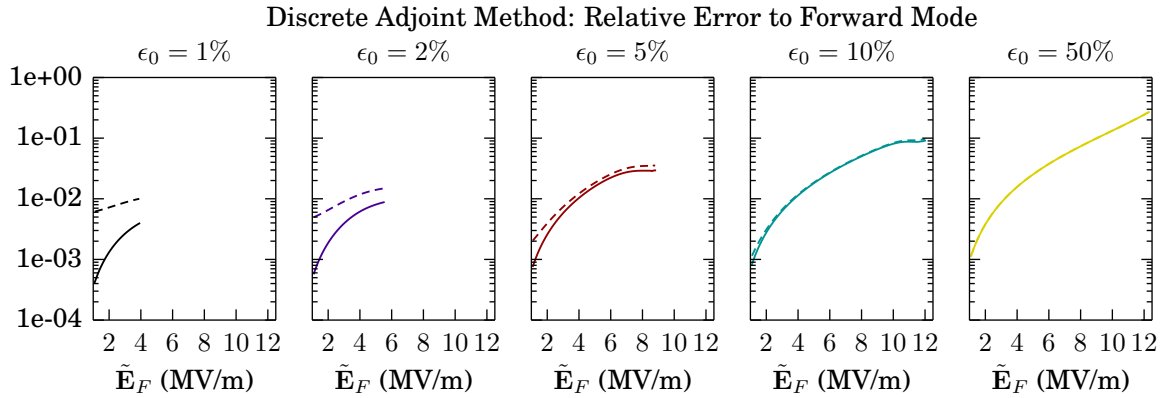


Figure 6.8: Relative error of the sensitivities computed using the conventional discrete adjoint with an approximate Jacobian, with respect to the forward mode of AD. Five different magnitudes of prestretch are considered. Load conditions: dead load (continuous line) and follower pressure (dashed line). Adapted from [142].

It can be observed that the error for the case of follower forces is up to one order of magnitude larger than for the case of an uniform dead load, particularly for the most flexible cases. This is caused by neglecting the contribution of the surface pressures to the Jacobian, $\mathbf{K}_{ext}$ in (2.39), as also explained in section 6.1.1. However, it is also noted that the error of the conventional adjoint grows consistently with the increase in the electric field modulus, E_F , for all five levels

of prestretch. The relative error ranges from values under 0.01%, to maximum values in the order of 10%. This behaviour responds to the contribution of the term $\mathbf{K}_{E,c}$ in (4.14), which is also neglected. This contribution is proportional to the Maxwell component of the stress, $c_E \propto \sigma_E$, and thus its effect grows with an increasing electric field.

Using a Central Differences scheme with a constant increment $h = 0.1$ MV/m, it can be observed (Fig. 6.9) that the behaviour of the error remains almost independent of the prestretch and also of the load condition. The error is, however, a function of the increment, $\mathcal{O}(h^2)$. The same pattern is repeated in all cases. The error is small for regions in which the slope of the objective function is small. The value of the $2h = 0.2$ MV/m, that was arbitrarily chosen, minimizes the error at an optimum that depends on the problem (see, e.g., the case of a prestretch $\epsilon_0 = 5\%$ for an electric field $E_m = 5.8$ MV/m). Finally, the error grows in the regions where the gradient changes more rapidly, that is, close to the minimum value of the objective function.

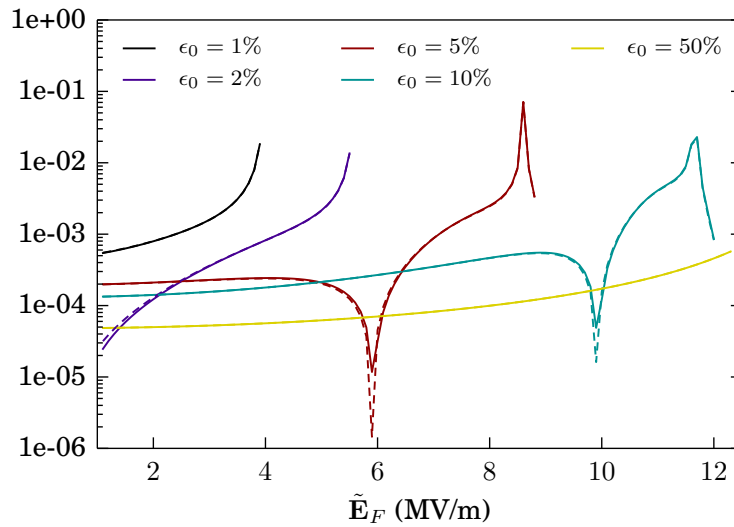


Figure 6.9: Relative error of the sensitivities computed using a Central Differences scheme, with a constant increment $2h = 0.2$ MV/m, with respect to the Forward mode. Five different magnitudes of prestretch are considered. Load conditions: dead load (continuous line) and follower pressure (dashed line). Adapted from [142].

The ADDA outperforms any of the previous methods for every simulation that has been run in this work, as shown in Fig. 6.10. The gradient is improved by more than 5 orders of magnitude with respect to the central differences scheme for the most flexible cases, and over 6 orders of magnitude with respect to the conventional discrete adjoint with the approximate Jacobian. Furthermore, the results of the AD-based discrete adjoint are effectively independent of the loading condition, although they seem slightly sensitive to the magnitude of the electric field modulus E_F . This effect is thought to be due to the value of $\Delta \mathbf{u}^*$ in the recording

process for the most non-linear cases, similarly to section 6.1.1. All cases were run to the same convergence criteria and a maximum number of iterations, but the convergence of the primal structural solver is poorer as the value of the electric field increases because of the stronger non-linearity generated by the electric effects.

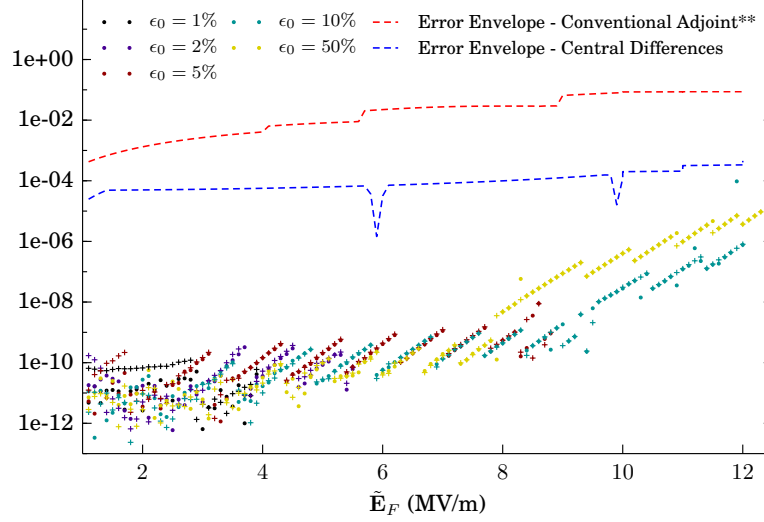


Figure 6.10: Relative error of the sensitivities computed using the ADDA with respect to the Forward mode. Five different magnitudes of prestretch are considered. Load conditions: dead load ($\bullet$) and follower pressure ($+$). **Indicates that an approximate Jacobian is used. Adapted from [142].

6.1.3 Dynamic FEA

The AD-based Discrete Adjoint method for structural applications was extended to dynamic problems in section 3.4.2. The accuracy of the gradients obtained with this technique is tested using a flexible cantilever under the action of a non-linear damper, as shown in Fig. 6.11. The cantilever is displaced from its equilibrium position using a uniform load, until the tip reaches a displacement of 1.5 cm. This corresponds to a 37.5% of its length, and thus strong geometrical non-linearities are expected in the analysis. Then, the cantilever is released and it starts a vibration with the highest component in its first bending mode.

The non-linear damper introduces a *follower* force,

$$\mathbf{F}_{damper} = -c_d \cdot \dot{\mathbf{u}}, \quad (6.4)$$

where $\dot{\mathbf{u}}$ is the velocity vector for all the DOF of the nodes upon which the damper actuates. Therefore, $\mathbf{F}_{damper}$ contributes to damp out these vibrations. The geometrical and material

properties of the cantilever are adopted from the coupled FSI problem in section 5.2.2, that is, $E = 250$ kPa, $\rho = 100$ kg/m³, and $\nu = 0.35$. The choice is justified in order to test the applicability of the method to dynamic, coupled problems at a further stage.

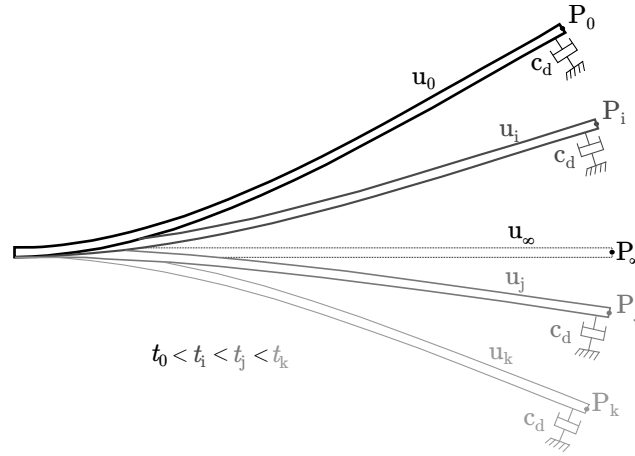


Figure 6.11: Test case description for the dynamic adjoint.

The instantaneous objective function for a given time position i is now defined as the distance between the position of point $\mathbf{P}$ in Fig. 6.11, $\mathbf{u}_i$, with respect to its undeformed position $\mathbf{u}_{tgt}$, as

$$J_i(\mathbf{u}_i) = \sqrt{(\mathbf{u}_i - \mathbf{u}_{tgt}) \cdot (\mathbf{u}_i - \mathbf{u}_{tgt})}. \quad (6.5)$$

That means that only the displacements at the tip are evaluated, and the objective function that we aim to evaluate is the average of the values of J_i for the time window of interest, $[0, t_F]$, where t_F is the final time step, as

$$J(\mathbf{u}) = \frac{1}{t_F} \sum_{i=0}^{t_F} J_i(\mathbf{u}_i). \quad (6.6)$$

For $t_F = 2.0$ s, the time history of the instantaneous and averaged value of the objective function is shown in Fig. 6.12.

The averaged objective function in (6.6) has been evaluated for a range of values of $E \in [235, 265]$ kPa, and $\rho \in [94, 106]$ kg/m³, and it is shown in Fig. 6.13. It can be observed that ρ is the dominant variable in the problem with the larger gradients, which can be explained as the result of a smaller inertia in the cantilever that allows for the vibrations to be more quickly damped out.

The sensitivities of the objective function with respect to E and ρ are computed using the forward mode of AD and presented in Fig. 6.14. It can be observed that, although the function

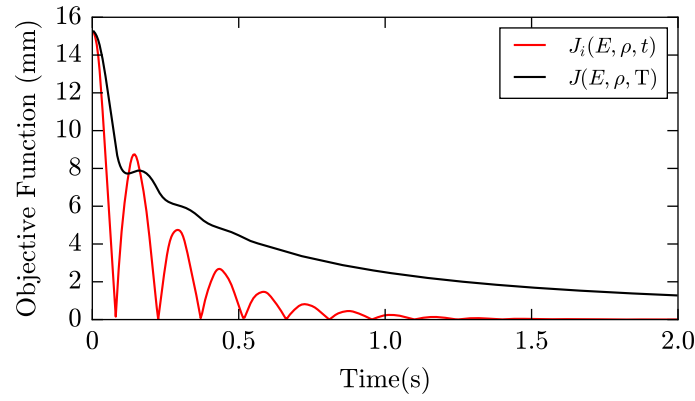


Figure 6.12: Time history of the instantaneous (J_i) and the averaged (J) value of the objective function.

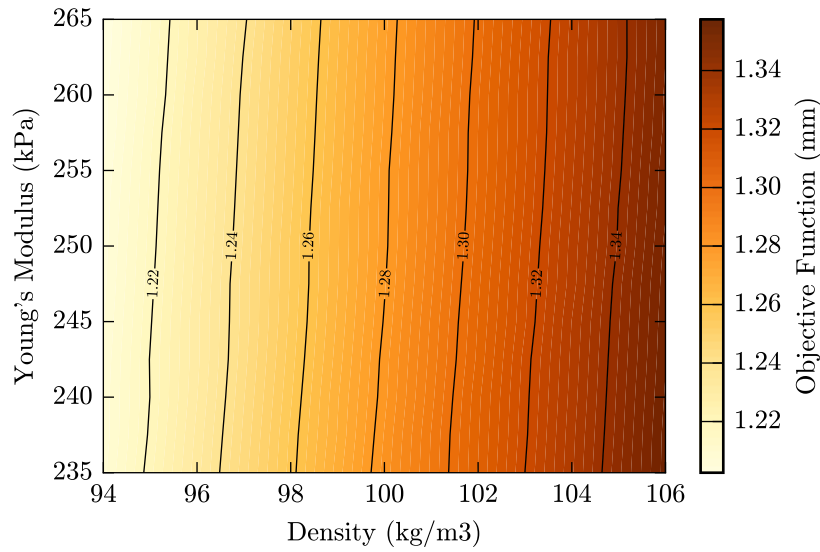


Figure 6.13: Values of the objective function in the region of study.

seems rather smooth, the values of the sensitivity present a large variability that should be accounted for.

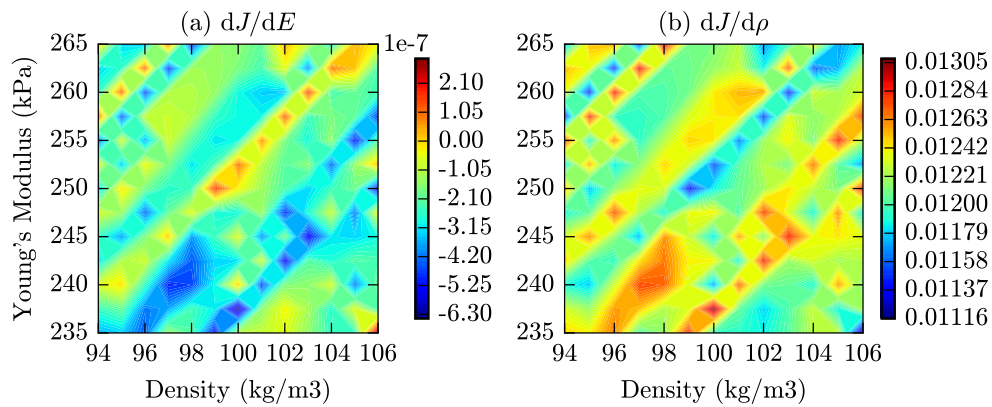


Figure 6.14: Sensitivities computed using the forward mode of AD.

One single ADDA simulation is enough to compute both dJ/dE and $dJ/d\rho$ at once using the adjoint method. The relative error with respect to the sensitivities obtained using the forward mode is shown in Fig. 6.15. It can be observed that this error remains below the threshold of 0.01% for almost all cases, demonstrating the accuracy of the implementation.

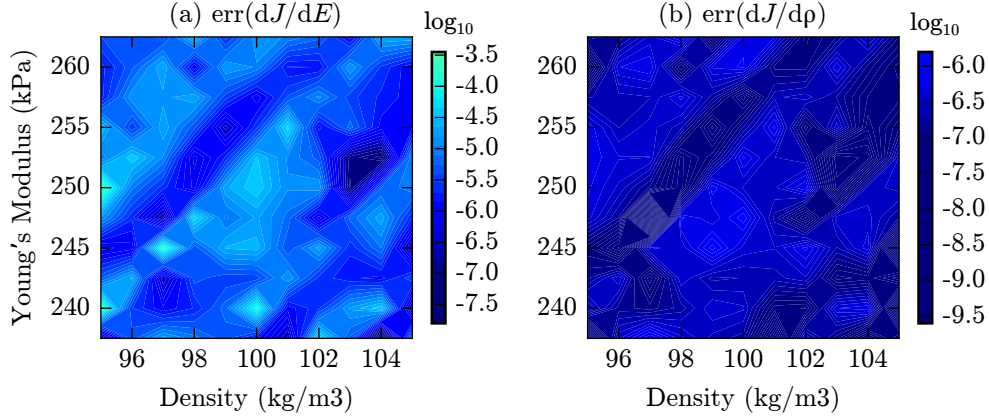


Figure 6.15: Relative error ($\log_{10}$) of the sensitivities computed using the ADDA method with respect to the forward mode.

The sensitivities are now computed using central differences. The increment is set to be a 1% of the reference values for E and ρ , that means $h_E = 2500$ Pa and $h_\rho = 1$ kg/m³. It is shown in Fig. 6.16 that the relative error obtained for these increments is large, even beyond 100% for some cases in the value of dJ/dE . The error is larger for this sensitivity than for $dJ/d\rho$, which denotes that this second function must be smoother. This behaviour of the error agrees with the observed in Fig. 6.15 for the ADDA method.

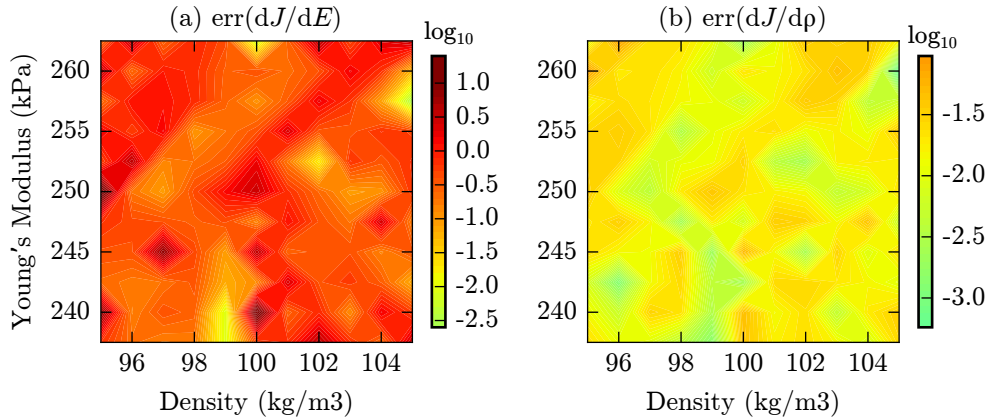


Figure 6.16: Relative error ($\log_{10}$) of the sensitivities computed using Central Differences with respect to the forward mode.

Indeed, fixing the value of E to 250 kPa, the behaviour of $J(\rho)$ shows to be relatively smooth, as presented in Fig. 6.17. The values of the sensitivities are also presented in Fig. 6.17, where it can be observed that the accuracy of the gradients computed using central differences and

$h_\rho = 1 \text{ kg/m}^3$ is reduced. The size of h_ρ can be further reduced to 0.1 kg/m^3 , which leads to an improved solution that more closely resembles the gradient obtained using the forward mode - and the ADDA method.

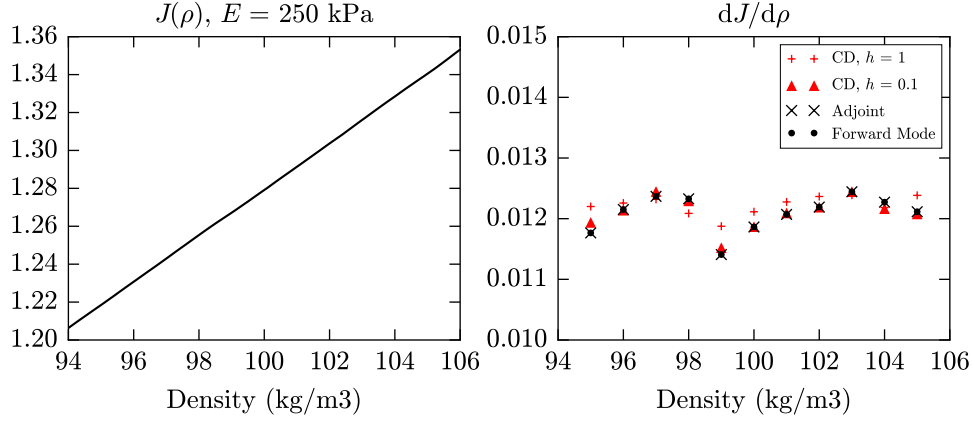


Figure 6.17: Objective function (a) and gradient with respect to the density (b) for a fixed value of $E = 250 \text{ kPa}$.

Fixing the value of ρ to 100 kg/m^3 leads to a more uneven function for $J(E)$, which agrees well with the larger error obtained for central differences in Fig. 6.16. Again for this case, reducing the size of the increment h_E in one order of magnitude to 250 Pa leads to a more accurate value of the sensitivity, that nevertheless remains far from the very accurate value obtained using ADDA. This demonstrates that ADDA not only reduces the number of simulations that are required for a larger number of design variables, but also that it provides a consistently accurate value of the sensitivities.

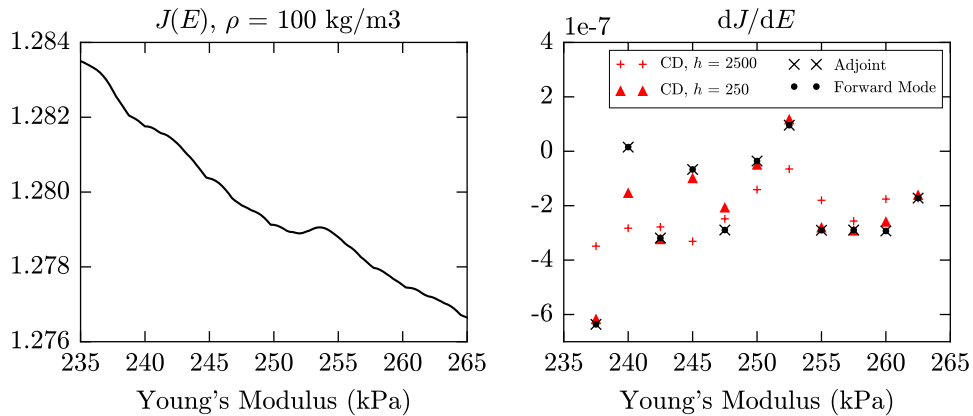


Figure 6.18: Objective function (a) and gradient with respect to the Young's Modulus (b) for a fixed value of $\rho = 100 \text{ kg/m}^3$.

6.2 Coupled FSI Problems with Large Deformations

In this section, we study the applicability of the ADDA method as described in section 3.5.1 to coupled FSI problems. First, a vertical, flexible wall with the same properties as the one used in section 6.1.1 is immersed in a horizontal, viscous flow (section 6.2.1). The accuracy of the gradients obtained using the ADDA method is verified, and the main advantage of this technique for problems involving multiple design variables is highlighted. Next, the method is tested for a prestretched and actuated membrane using the problem settings of section 5.2.1. Using this setup, the ability of the methodology to compute gradients for coupled, actuated problems is demonstrated in section 6.2.2.

6.2.1 Vertical Wall in Viscous Flow

We now redefine the flexible cantilever problem of section 6.1.1, to be subject to a fluid loading condition, as shown in Fig. 6.19. The primal problem becomes a fully-coupled, steady-state FSI problem. The properties of the fluid domain are set so that the order of magnitude of the deformation remains consistent with the structural problem in vacuum described previously in this work. However, the forces acting over the structure are now determined by the solution of the flow field, which at the same time is affected by the deformation of the wall.

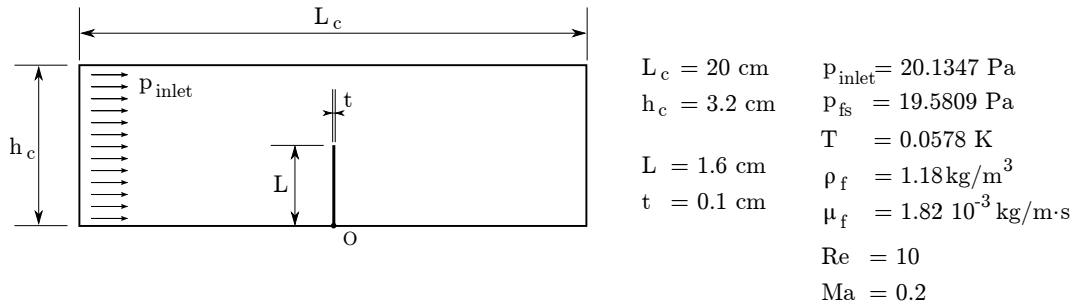


Figure 6.19: Test case description: vertical wall in viscous flow. Adapted from [141].

We have limited the size of the problem in order to simplify this investigation and reduce the computational time involved. However, the complexity of the physics involved is not decreased, maintaining the material and geometrical non-linearities in the structural domain while fully resolving the Navier-Stokes equations including viscous effects. The deformation of the cantilever for the two extreme cases of the range $E \in [13, 28] \text{ kPa}$ is shown in Fig. 6.20, where a

maximum amplitude of roughly 40% of the height of the cantilever is reached.

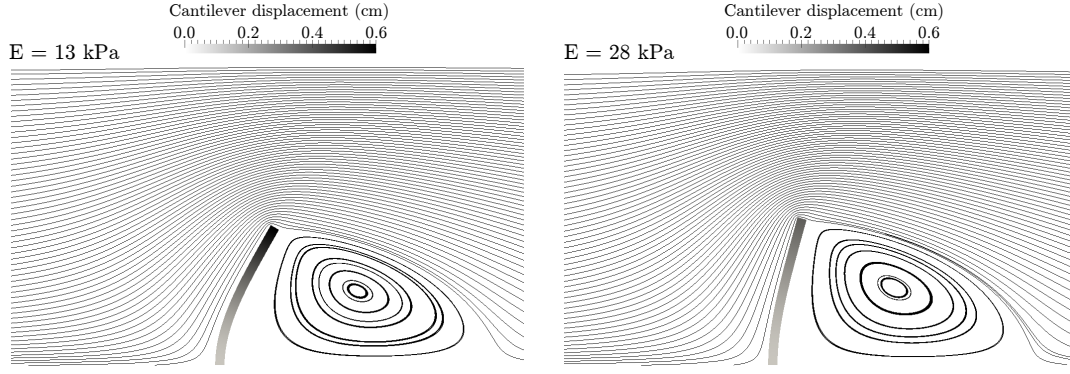


Figure 6.20: Flow streamlines and cantilever deformation with $E = 13$ and 28 kPa [141].

We adopt the shape $\mathbf{u}_{tgt}$ in (6.1) to be the undeformed configuration, $\mathbf{u}_{tgt} = \mathbf{0}$ much as in section 6.1.1. The values of the objective function and the sensitivity dJ/dE are presented in Fig. 6.21, together with the relative errors for the ADDA method and central differences. The conventional adjoint method has not been attempted as the exact Jacobian of the problem in (2.55) is not available. In fact, deriving this Jacobian is very challenging [42, 57, 91] and one of the purposes of this thesis is avoiding it altogether.

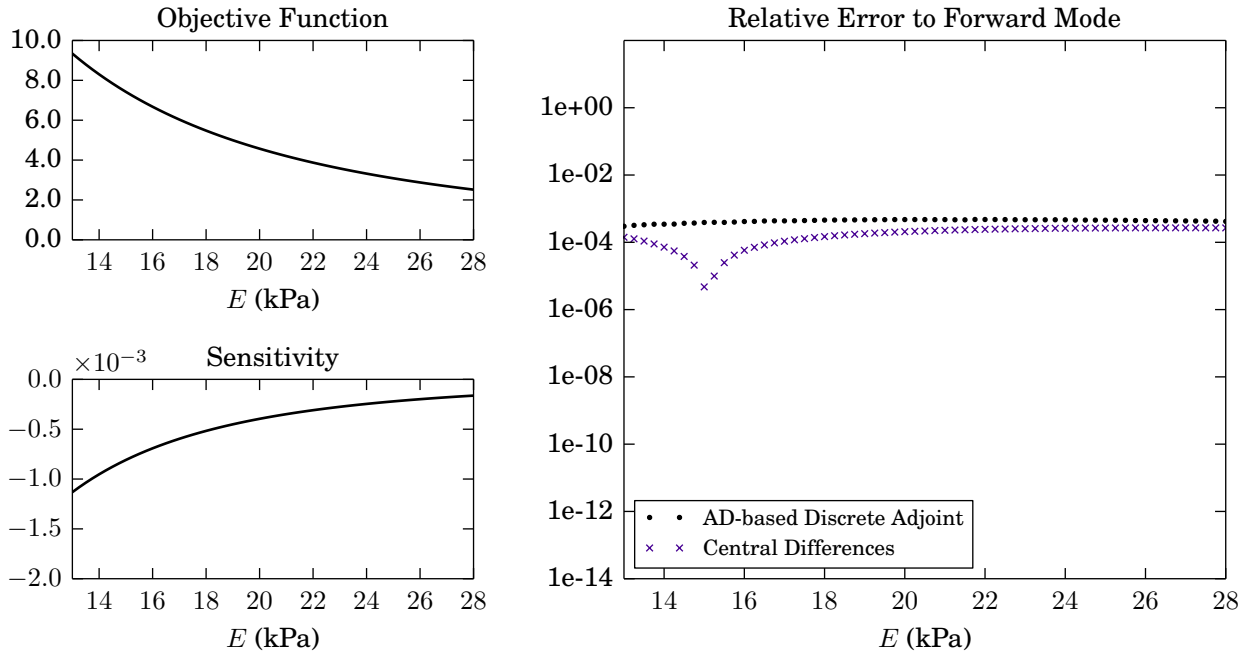


Figure 6.21: Objective function (top left), sensitivity (bottom left) and relative error between ADDA method and central differences with respect to the forward mode of AD. Adapted from [141].

The relative error obtained with central differences is in the order of $1E-4$, consistently with the values obtained in section 6.1.1 for the same increment, $2h = 500$ Pa. The gradient obtained using ADDA remains in that same order of magnitude, with a very good approximation to

the exact gradient computed using the forward mode. For a fully-coupled, non-linear Fluid-Structure Interaction problem, the relative error remains within a 0.05%. It is explained by Albring [120] that some approximations are made in the general reversal procedure for the fixed-point solvers, which are likely related to the increase in relative error with respect to the single-discipline, structural problem.

The main advantage that is offered by the ADDA method with respect to the forward mode of AD, and also with respect to central differences, is its ability to provide the gradients with respect to any number of inputs or design variables. For example, let $\alpha = [E, \nu]$ be the vector of design variables, which now incorporates the Poisson ratio. One single ADDA simulation is able to obtain dJ/dE and $dJ/d\nu$ at the same time. It is shown in Fig. 6.22 that after approximately 40 iterations of the BGS method in Algorithm 4 both gradients seem converged. It must be noted in Fig. 6.22 that a cubic function is used to transfer the loads in the primal solution for 40 subiterations of the BGS method, while no relaxation was not found necessary in the adjoint solution.

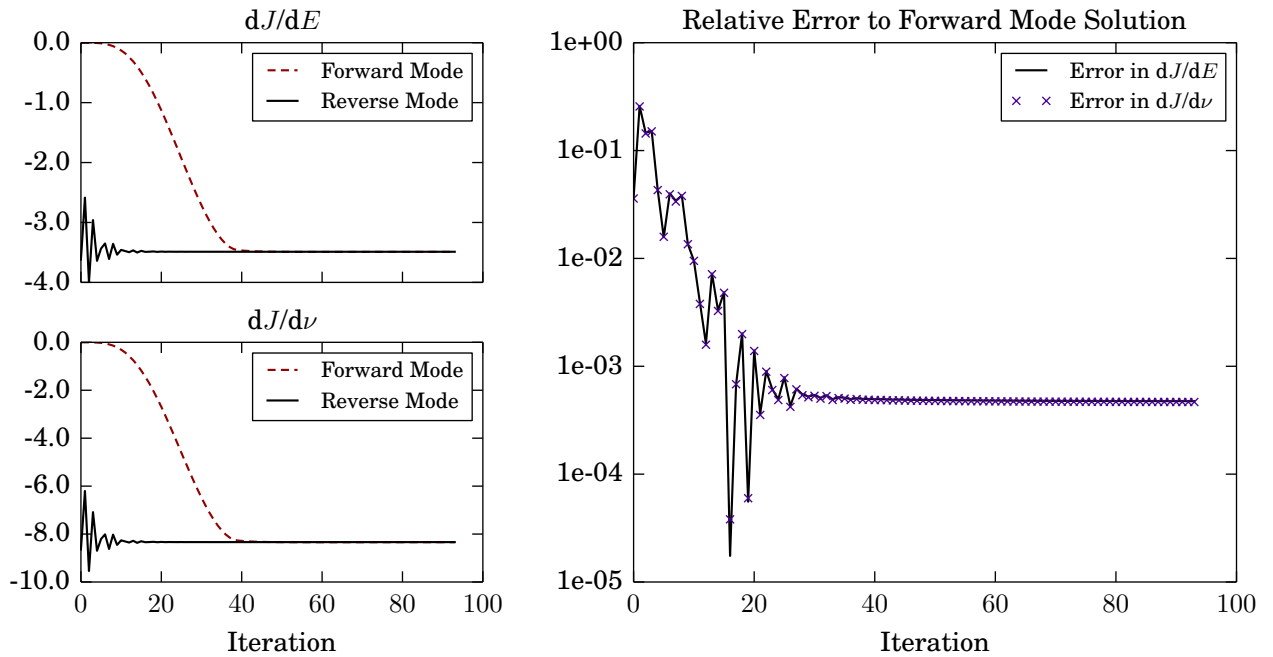


Figure 6.22: Convergence of dJ/dE and $dJ/d\nu$ using ADDA. Adapted from [141].

Moreover, another disadvantage of the central differences scheme is that the accuracy of the gradients is strongly dependent on the size of the increment, h , which normally requires a parametric study for each variable as the size of the step differs depending on their magnitude. This is shown in Fig. 6.23.

For this particular case, the ADDA method shows an excellent computational performance,

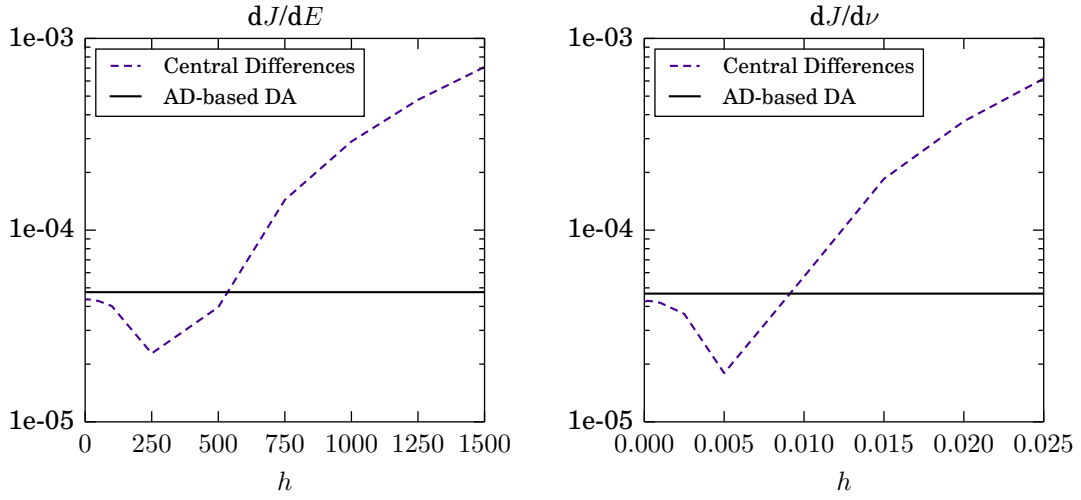


Figure 6.23: Relative error for central differences for different values of h , compared with the relative error of the ADDA, with respect to the gradient computed using the forward mode of AD. Adapted from [141].

both in terms of memory footprint and computational time, as shown in Tab. 6.1. With regards to the former, the overhead for the adjoint computation is approximately 6 times the memory used for the primal solution of the FSI problem. This footprint compares positively with respect to other examples in the literature, as for example the overhead reported by Mader *et al.* [117], which used approximately 10 times more memory when running the reverse mode of AD for a single-discipline flow adjoint problem, which at the same time was applied selectively and only to the routine that calculates the flow residual.

Table 6.1: Performance study of the ADDA solver. Memory footprint, computational time and convergence [141].

	RAM (Mb)	Time (rel.)	Convergence of theAD-based Discrete Adjoint					
FSI	68	t	Relative Error to FM		$\log_{10}$ of residual reduction			
FSI - FM	99	1.778 t	dJ/dE	$dJ/d\nu$	$\bar{\rho}$	$\overline{\rho E}$	$\bar{u}_1$	$\bar{u}_2$
FSI - RM	404	0.283 t	4.9476 E-04	4.8625 E-04	-3.024	-2.953	-5.210	-5.298
		0.734 t	4.7479 E-04	4.6660 E-04	-4.080	-4.013	-6.536	-6.854
		0.913 t	4.7344 E-04	4.6527 E-04	-4.847	-4.780	-7.215	-7.599
		1.291 t	4.7319 E-04	4.6502 E-04	-5.971	-5.904	-7.4475	-8.064
		1.575 t	4.7317 E-04	4.6500 E-04	-6.825	-6.758	-7.416	-7.997
		1.925 t	4.7317 E-04	4.6500 E-04	-7.867	-7.800	-7.443	-8.024

It is also shown in Tab. 6.1 that the ADDA takes 73.4% the computational time of the convergence of the primal solver to reduce the residuals of the flow and structural adjoints in 4 and 6 orders of magnitude respectively. Although these adjoints can be further converged extending the computational time, in practical terms it is interesting that using only a 28.3%

of the time invested in the primal solution is enough to obtain values of the gradient that are below a 0.05% error with respect to the forward mode solution.

As a result, the ADDA method provide a noticeable advantage for optimal design with a large number of design variables. To illustrate this, we discretise the cantilever into 10 regions, as shown in Fig. 6.24, and we compute the sensitivities for the objective function with respect to the Young's modulus in each of them. Larger sensitivities are expected towards the root of the cantilever as there is a bigger moment generated by the fluid forces. Therefore, the discretisation is not uniform and the regions are smaller towards the bottom of the wall. The vector of design variables is now $\alpha = [E_1, E_2, \dots, E_{10}]$.

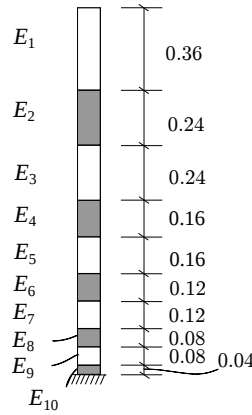


Figure 6.24: Geometrical discretisation of the cantilever into 10 regions, each corresponding to one design variable. Adapted from [141].

Using an uniform value for all design variables, $E_i = 21 \text{ kPa } \forall i \text{ in } (1, 10)$, we run 10 FSI simulations with the forward mode activated, in order to obtain the sensitivities for all 10 regions. As shown in Tab. 6.1, the simulations with the forward mode enabled have a larger computational cost than the regular FSI simulations.

Using central differences to compute the gradients requires running 21 FSI simulations: one to obtain the value of $J(\alpha)$, 10 to evaluate J for each independent design variable incremented, $J(\alpha_i + h) \forall i \text{ in } (1, 10)$, and another 10 to obtain each value of $J(\alpha_i - h) \forall i \text{ in } (1, 10)$. The values using each of these two methods are compared in Fig. 6.25 with the sensitivities obtained using the ADDA method, which only requires one FSI primal solution and one reverse simulation.

The ADDA method outperforms central differences in 9 out of the 10 regions, with a relative error below 0.01%. It can be observed from Fig. 6.25 that the absolute error of ADDA with respect to the forward mode remains in the order of 1E-9, which is linked to the fact that the sensitivities are all evaluated simultaneously. Central differences give nearly constant relative errors, due to the two additional simulations to convergence of the primal solver that are

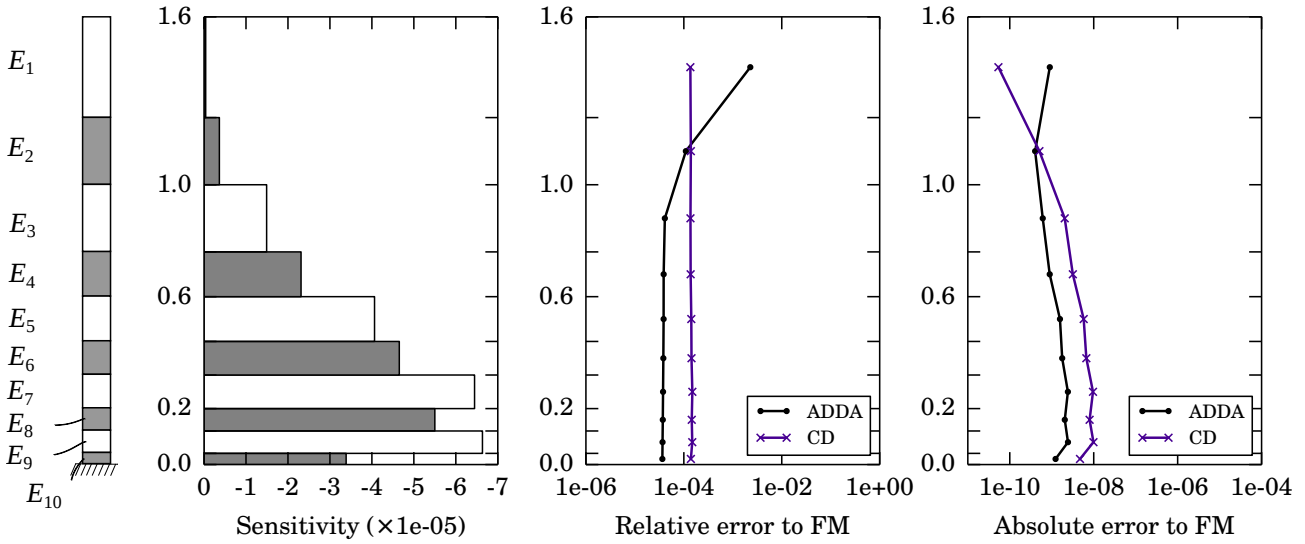


Figure 6.25: Far left: Design variable distribution. Left: Sensitivities for the design vector. Centre: Relative error of the sensitivities computed using ADDA and central differences (CD), $2h = 500$, with respect to the value computed using the forward mode (FM). Right: Absolute error of the same magnitude. Adapted from [141].

required per design variable. For optimisation purposes, the interest is normally on the design variables that have a larger sensitivity (thus, a larger impact on the objective function). The ADDA implicitly targets those values by maintaining a low absolute error in all cases, which highlights its advantageous properties for optimal design in coupled problems.

6.2.2 Electro-Mechanically-Actuated Membrane Wing

Finally, we extend the problem of the electromechanically actuated membrane in section 6.1.2 to a fully coupled, FSI problem with the properties described in section 5.2.1. We choose the prestretch condition $\epsilon_0 = 5\%$, because it offers a wide range of compliance for the actuation range $E_m \in (2-8) \text{ MV/m}$, as it is shown in Fig. 6.6. We maintain the target geometry, $\mathbf{u}_{tgt}$ in (6.2), to be the equilibrium position of the unactuated, unprestretched membrane as previously done in section 6.1.2.

The objective function and the sensitivities computed using the forward mode of AD are shown in Fig. 6.26. The values of the sensitivities computed using ADDA and central differences, $h = 0.1 \text{ MV/m}$, are also presented and compared with the value obtained using the forward mode. The accuracy of the gradients obtained using ADDA is demonstrated, as they show a very good agreement with the forward mode, with relative errors below 0.01%.

In summary, this chapter has demonstrated the robustness of the ADDA method to evaluate sensitivities in structural and coupled problems. While only simple geometries were considered,

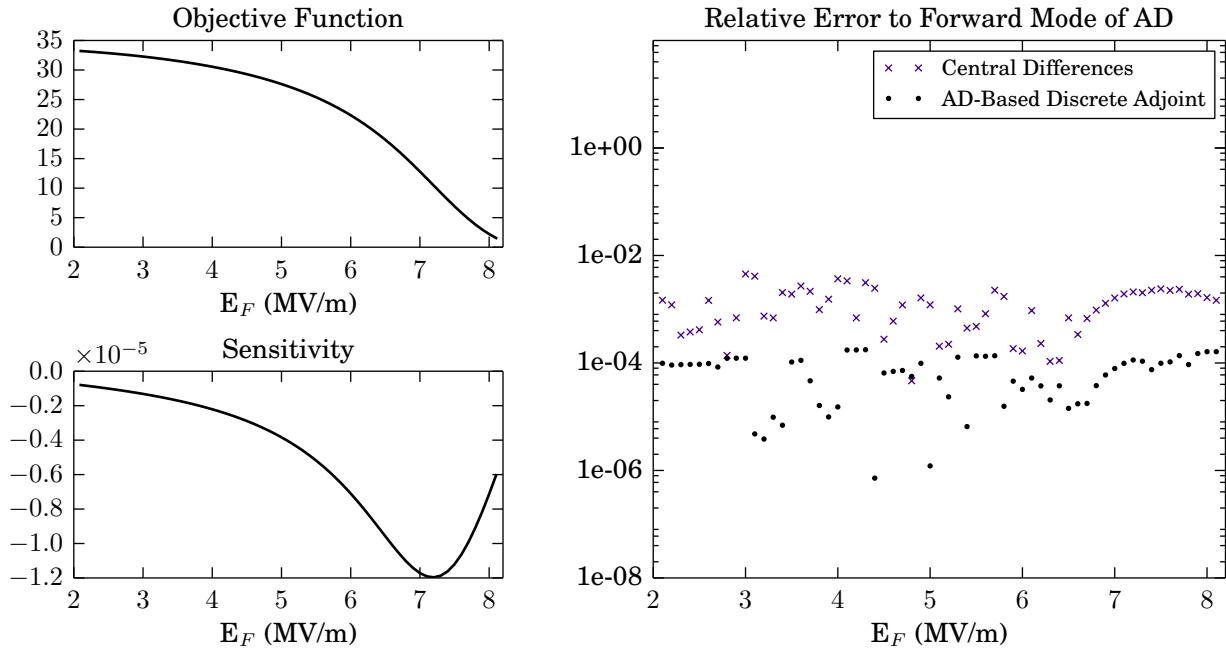


Figure 6.26: Objective function (top left), sensitivity (bottom left) and relative error between ADDA method and central differences with respect to the forward mode of AD, for an electromechanically actuated membrane wing. Adapted from [142].

we dealt with complex constitutive laws, strong geometrical non-linearities, and closely coupled multi-disciplinary problems involving viscous flows. The results obtained for the sensitivities encourage the applicability of the method in gradient-based optimal design, as it will be done in the next chapters.

Chapter 7

Optimal Design and Actuation of Structures with the ADDA Method

In this chapter, gradient-based optimal design and actuation in very flexible structures using the ADDA method for gradient calculation will be investigated. A Python framework has been used in order to set up the optimisation problem. A bespoke framework had to be developed, starting from scripts from aerodynamic shape optimization for the CFD solver that required major rework for structural and coupled FSI problems. This environment is described in section 7.1. It is in charge of selectively calling the primal and adjoint solvers available in SU2. The values of the objective function and the gradient will be used by the sequential least-squares programming (SLSQP) optimiser available in SciPy [213] to update the design variables and drive the optimisation. Next, three test cases will be presented in order to demonstrate SU2's ability to perform structural gradient-based optimisation. Section 7.2 solves the inverse problem of obtaining a pre-defined stiffness distribution from different initial guesses. Section 7.3 evaluates the ability of the code to find optimal electromechanical actuation distributions. Finally, section 7.4 demonstrates the capabilities of the method in dynamic simulations by finding an optimal distribution of mass to faster damp out structural vibrations starting from an out-of-equilibrium state.

7.1 Software architecture

The Python framework available in SU2 at the start of this project is described in Fig. 7.1, and it was originally intended for optimal aerodynamic shape design. An object “Project” is instantiated at the beginning of the simulation, which reads in the configuration file *.cfg and generates two objects: a “Config” object and a “State” of the optimisation object. The configuration file contains all the information required for the primal and adjoint simulations plus all the information regarding the optimisation: physical problem, design variables, tolerances, boundaries, objective function, etc. Then, the information is redistributed into the “State”, which stores the values of the optimisation (objective function, variables, gradients, file input/output information). Both objects are passed into a “Design” object, one for each optimisation stage or iteration, which wraps the binaries of the primal and adjoint problems, generates a file structure and runs the simulations as needed.

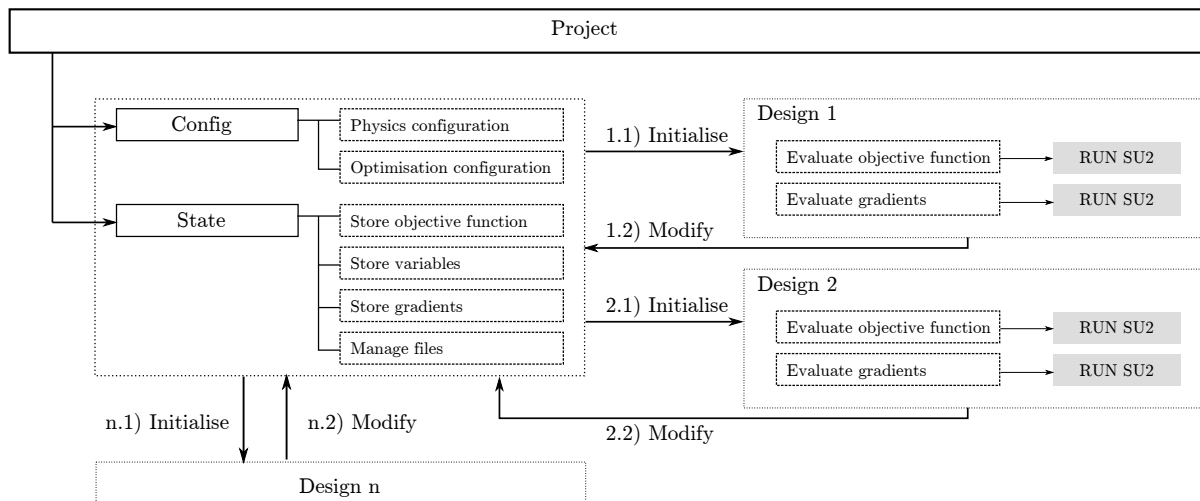


Figure 7.1: Python framework available in SU2 for optimal aerodynamic shape design.

This structure proved to be too rigid for structural problems and incapable of dealing with multi-physics FSI problems. The framework was unable to work with information generated by different fields, and the optimisation configuration was too strictly devised for shape optimisation: the way of dealing with different objective functions, design variables and gradients was fixed and new options were not easily incorporated.

Within this project, a new, more flexible optimisation framework for SU2 was prototyped for structural and multiphysics problems, although it was not incorporated to the code distribution

as its adaptation to other kinds of problems is not yet complete due to time limitations¹. The outline of the code structure is shown in Fig. 7.2. The configuration input has been split between the primal, adjoint and optimisation configurations, allowing for a greater flexibility in the definition of the problem. Three new classes for design variables, objective function and physical problem definition have been generated, which exploit Python's object-oriented characteristics. They generate different child classes depending on the kind of optimisation problem that we intend to run, and handle the input of functions, variables and gradients, and the output of different files for different fields, in a more abstract way that increments the flexibility of the code.

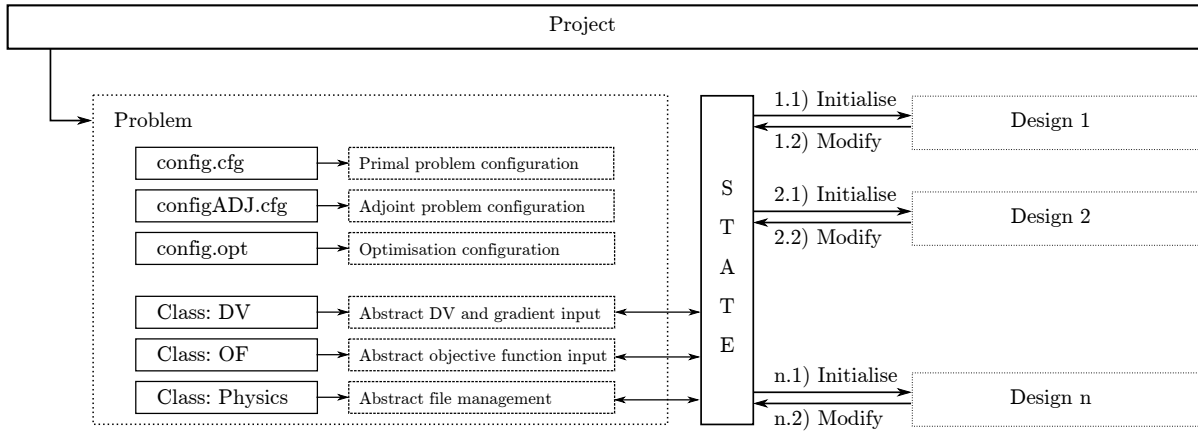


Figure 7.2: New Python framework available in SU2 for structural and multiphysics problems.

7.2 Optimal Parameter Identification

In this section, the applicability of the ADDA method in gradient-based optimisation of flexible structures will be demonstrated. We choose the geometry of the vertical cantilever beam in section 6.2.1. The structural properties are set to $E_{ref} = 20$ kPa, $\nu = 0.4$. The cantilever is subject to an uniform, follower pressure, $p = 2$ Pa. We divide the cantilever into 32 regions, as shown in Fig. 7.3(a). The design variables, $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_{32}]$, are multipliers to the stiffness in each of these regions; that means, for region i , its stiffness is determined as $E_i = \alpha_i E_{ref}$. We define a parabolic distribution of stiffness, $\alpha_{Objective}$, which follows the equation

$$\alpha_i = \frac{1.5}{961}i^2 - \frac{93}{961}i + 2.0, \quad (7.1)$$

¹This framework can be found in the branch *feature_optimization* in the code's repository.

as shown in Fig. 7.3(b). This distribution of stiffness yields a deformed shape under the actuating pressure (Fig. 7.3,c), which defines the value of $\mathbf{u}_{tgt}$ to be used in the objective function,

$$J(\mathbf{u}) = (\mathbf{u} - \mathbf{u}_{tgt})^T \mathbf{C}(\mathbf{u} - \mathbf{u}_{tgt}). \quad (7.2)$$

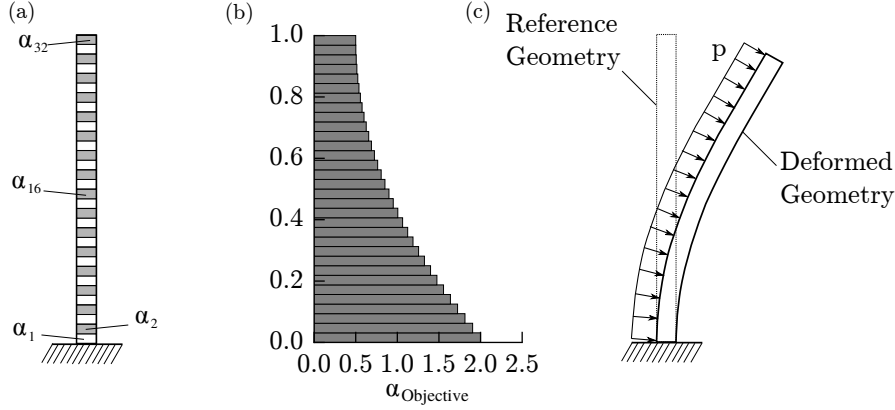


Figure 7.3: Objective distribution of stiffness and deformed geometry for $p = 2$ Pa.

We test the ability of the optimiser to perform gradient-based optimisation using the ADDA method for the computation of the gradients. The objective is to retrieve the objective distribution of design variables $\alpha_{Objective}$ starting from different initial guesses. First, the optimisation is started from an uniform initial guess, $\alpha_i = 1.0 \forall i$ in $(1,32)$. The SLSQP optimiser is allowed to run for up to 200 iterations. The initial guess, some intermediate stages and the final, optimised distribution of stiffness are shown in Fig. 7.3. We observe that the optimised distribution accurately represents $\alpha_{Objective}$ except for the top 10 regions, which are the regions with a smaller sensitivity as it was also discussed in section 6.2.1.

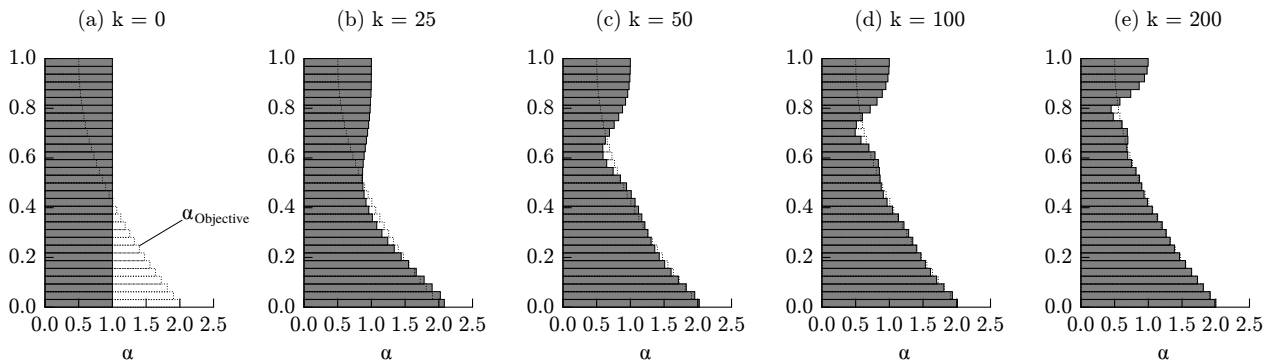


Figure 7.4: Optimisation evolution of the inverse problem for an uniform initial guess.

The optimised solution is clearly improved when a linear distribution is used (Fig. 7.4). In this case, the initial guess is stiffer than the known solution, and we observe that the optimiser is able to match the stiffness distribution for almost every region, starting from the root and moving

from bottom upwards. This behaviour agrees well with the distribution of the sensitivities, as expected.

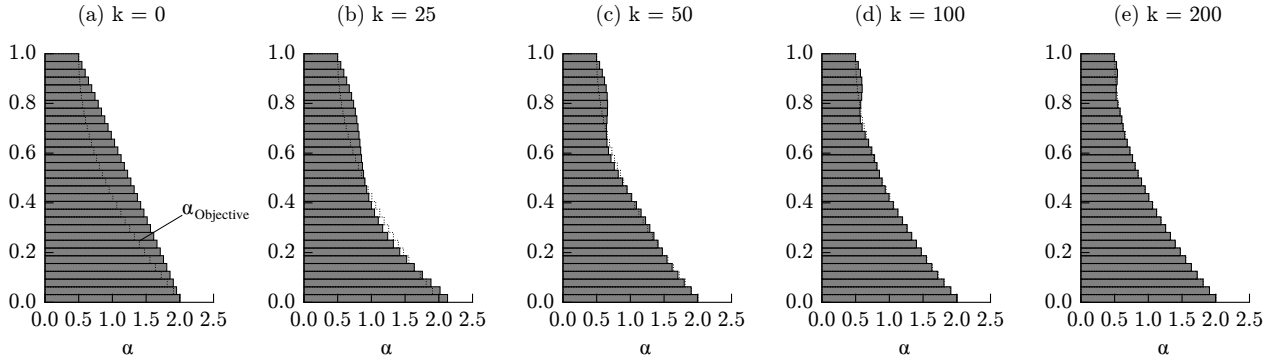


Figure 7.5: Optimisation evolution of the inverse problem for a linear initial guess.

Another attempt was run starting from a parabolic distribution, $\alpha_i = \frac{0.75}{961}i^2 - \frac{46.5}{961}i + 1.5$, which resembles the distribution in Eqn. 7.1 but yields different values at the root and tip. In this case, the optimiser is also able to accurately match most of the objective distribution, apart from some minor discrepancies at the tip. This behaviour shows that the optimiser struggles to find an accurate solution when the initial guess at the regions with smaller sensitivity are far from the optimal solution.

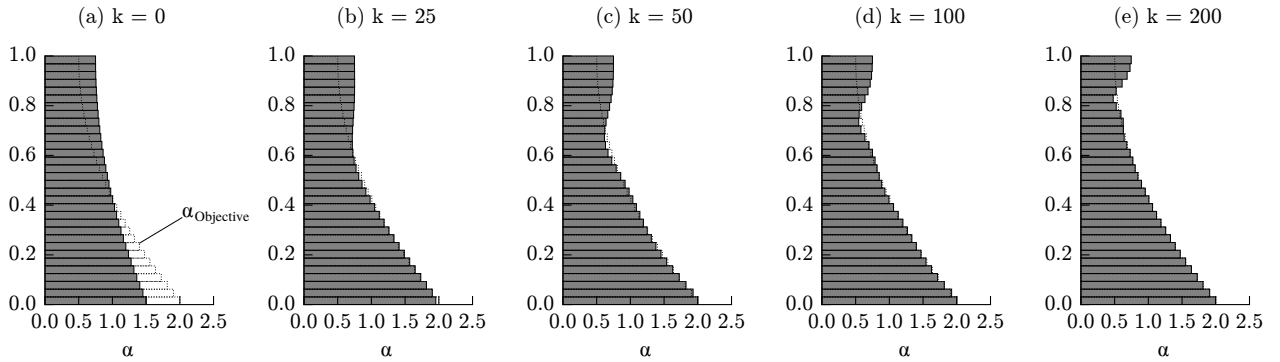


Figure 7.6: Optimisation evolution of the inverse problem for a parabolic initial guess.

The history of the optimisation process for the previous three cases is shown in Fig. 7.7. We observe that in all cases an accurate representation of the target geometry $\mathbf{u}_{tgt}$ is obtained, as the minor discrepancies at the tip barely affect the value of J in (7.1).

A final test was run using again a parabolic initial guess, but improved in the sense that the exact value at the tip of the cantilever is set from the start, $\alpha_i = \frac{0.5}{961}i^2 - \frac{31}{961}i + 1.0$. It can be observed that just 25 iterations are enough to accurately obtain our objective distribution of stiffness $\alpha_{Objective}$, while the remainder of the optimisation just refines their value to an almost exact solution.

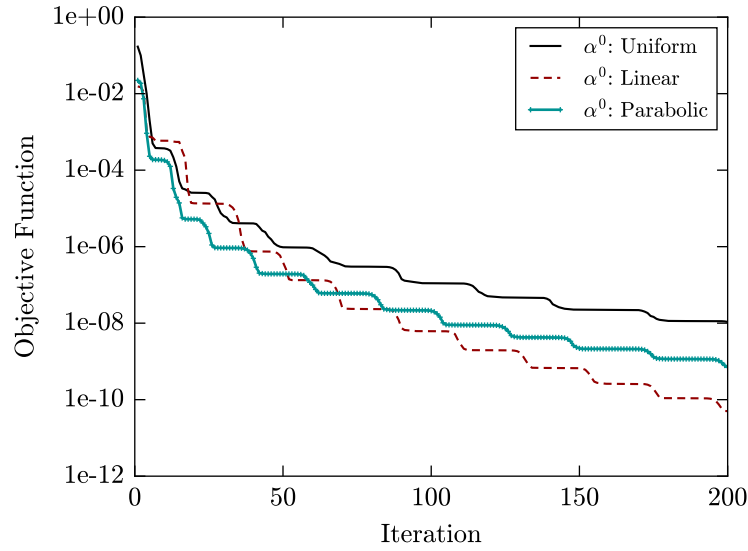


Figure 7.7: Optimisation history of the inverse problem for the uniform, linear and parabolic initial guesses.

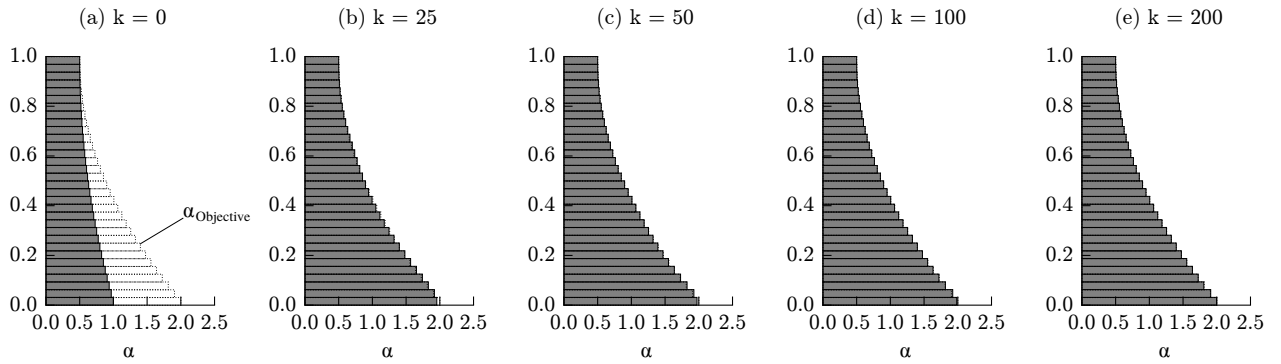


Figure 7.8: Optimisation evolution of the inverse problem for an improved parabolic initial guess.

It is shown from the optimisation history in Fig. 7.9 that the value of J is recovered almost exactly after 200 iterations, with a value below 10^{-12} . This is the global minimum of this particular problem, given that it was defined as an inverse problem aiming to retrieve a known solution. This simple example demonstrates the validity of the ADDA methodology for the solution of inverse problems in very flexible structures.

7.3 Optimal Electromechanical Actuation

In this section, the adjoint method for DEs as described in section 4.3 is tested to verify whether it may be used to actuate very flexible, thin structures. The geometry of the problem chosen for this test, Fig. 7.10, is the same as for the flexible cantilever used for the FSI benchmark in section 5.2.2. The reason for this is that one of the potential applications of the method

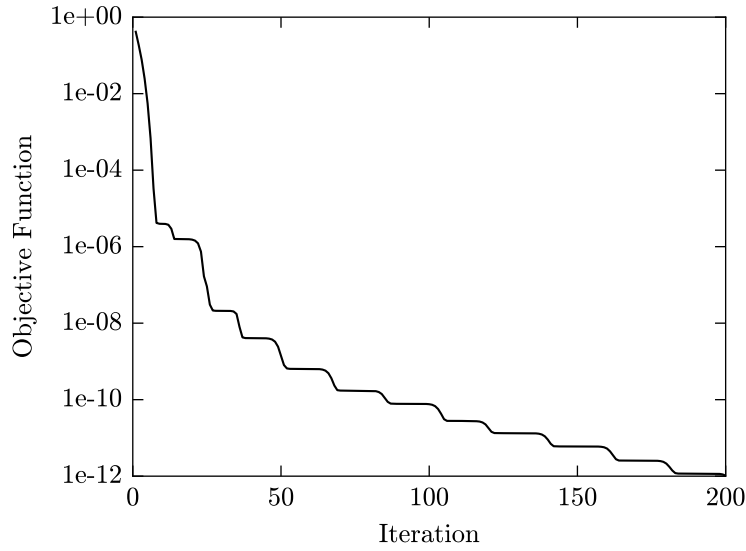


Figure 7.9: Optimisation history of the inverse problem for the improved parabolic initial guess.

described in section 3.5.3 for coupled FSI dynamics may be to control the cantilever deflections for that test case. This is a similar problem as that proposed by Bazilevs *et al.* [106], but using dielectric elastomers as a control parameter instead of a vertical load at the tip, as we believe this would be more realistic.

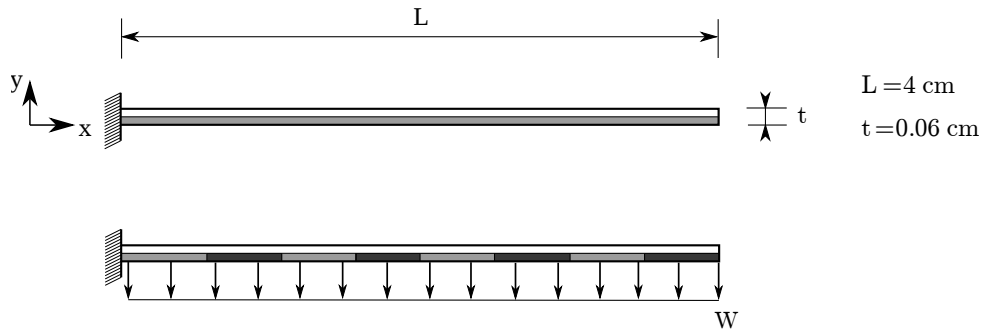


Figure 7.10: Cantilever bimorph and distribution of DE actuators in the bottom half.

The actuation consists of two layers of electroactive material, in a similar way as done by Ask *et al.* [194]. When the dielectric permittivity, ϵ , is positive, the electrical actuation can only relax the material, as explained in section 6.1.2. As a result, only the bottom layer will be actuated, to induce a recovery moment that counteracts the negative vertical loads in the direction of the axis y , such as gravity.

The cantilever is subject to its own weight, for a uniform density of 20 kg/m^3 . The discretisation was done with 4 elements in the thickness, for a total of 512 linear, 4-node elements. For the stiffness parameters of the example in 5.2.2, $E = 250 \text{ kPa}$, $\nu = 0.35$, this body load generates a vertical deflection in the tip of more than 0.6 cm , which is beyond 15% of the length

of the cantilever (Fig. 7.11). Therefore, the structure is in the geometrically non-linear regime. The material model chosen at this time is the compressible, Neo-Hookean material model. The permittivity of the dielectric is $\epsilon = \epsilon_0 \epsilon_r$, where ϵ_0 is the vacuum permittivity $\epsilon_0 = 8.8542 \cdot 10^{-12}$ F/m, and ϵ_r is the material relative permittivity. This has been chosen to be the value of the VHB4910 membrane, $\epsilon_r = 4.8$ from the work of Buoso and Palacios [20].

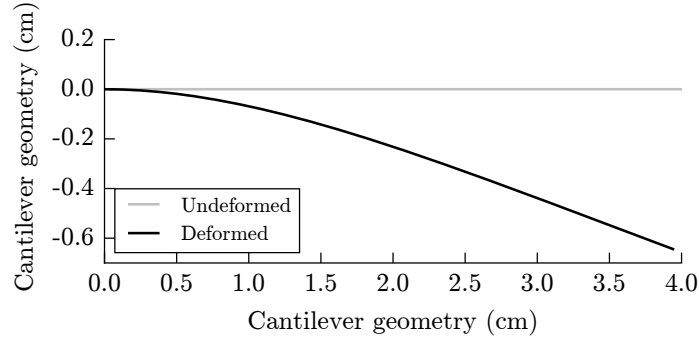


Figure 7.11: Deformed geometry under no actuation, $E = 2.5 \cdot 10^5$ Pa, $\nu = 0.35$, $\rho = 20$ kg/m³.

The objective is to minimise the deformation of the cantilever, resulting that the value of $\mathbf{u}_{tgt}$ in (6.1) is $\mathbf{0}$. The deformation of the cantilever is a function of the bending moment introduced by the dead loads, $M(x)$, x being the longitudinal direction of the cantilever, which in linear theory corresponds to a parabola for a clamped-free beam. The ideal optimal actuation of the beam would introduce a bending moment of opposite sign, $-M(x)$, which would determine a continuous distribution of the voltage in the electrodes of the elastomer. In this work, we have adopted a discrete approach, in which the objective will be to minimise the objective function J using 1, 8 and 64 electric field design variables distributed along the bottom half of the bimorph, as exemplified in Fig. 7.12a),b) for 1 and 8 actuators respectively. The bending moment introduced by the actuation will consequently be discrete, and is also qualitatively shown in Fig. 7.12.

In Fig. 7.13, the objective function and the optimal geometry obtained for the three test cases is shown. The initial guess is set in all cases to an uniform value of 4 MV/m. For the case of 1 unique actuator along the bottom half of the cantilever, the final geometry still remains far from the undeformed configuration. This is because an uniform actuation moment is unable to counteract the bending moment generated by the dead loads of the beam. Using 8 actuators yields an objective function which is reduced in 2 orders of magnitude with respect to the value of J obtained with 1 design variable. The actuated geometry closely resembles the undeformed configuration. An even more improved solution is obtained when 64 actuators are used, as

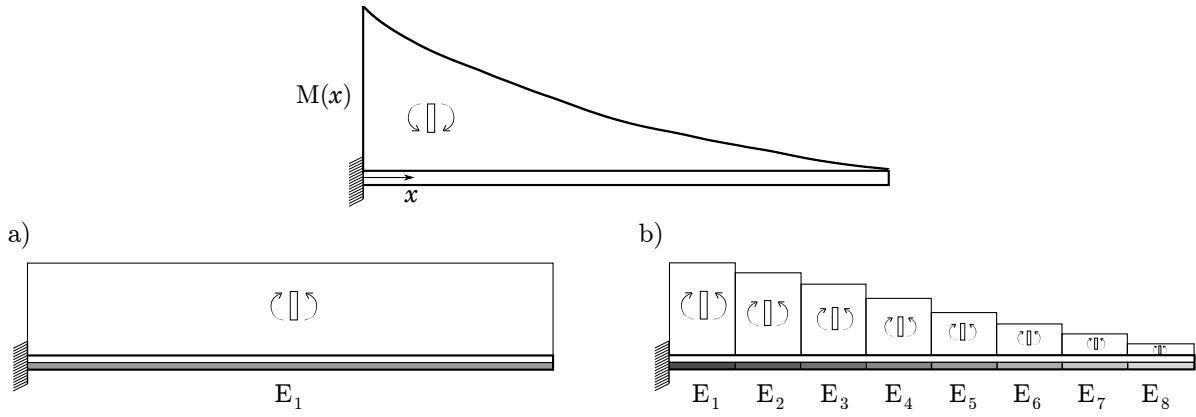


Figure 7.12: Cantilever bending moment, and actuation moments due to Electric Field using 1 (a) and 8 (b) design variables.

expected from a more refined discretisation.

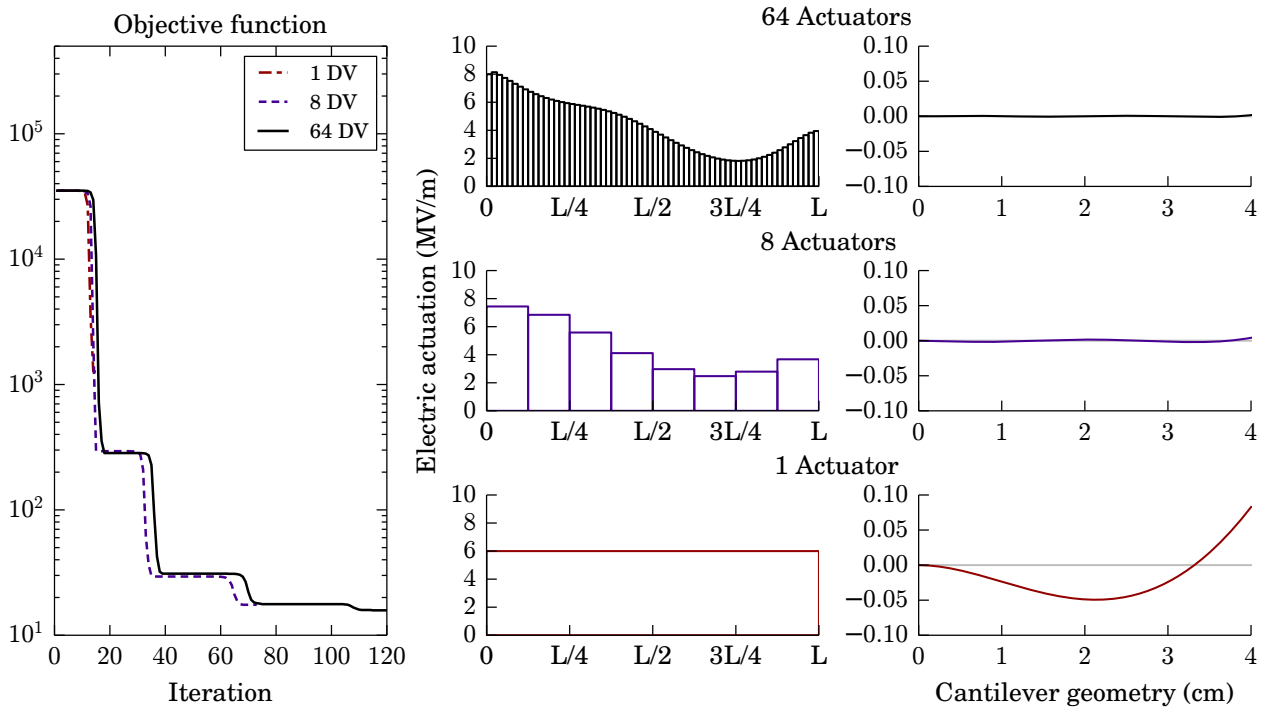


Figure 7.13: Time history of the optimisation (left), optimised actuation (centre) and final shape of the bimorph (right) for 1, 8 and 64 actuators.

In order to test whether a global optimum had been obtained, two additional cases were run using different initial guesses. In particular, the optimised values obtained from an initial guess of a linear distribution from 6 MV/m at the root to 0.1 at the tip, and a parabolic distribution with the same values at root and tip, are presented in Fig. 7.14.

In all cases, the final distribution of electric actuation yields a geometry that closely resembles the undeformed configuration. It may be observed that the highest actuation torque is applied close to the root of the cantilever in all cases. However, the distributions differ depending on the initial guess. The values of the objective function for all five cases presented in this section

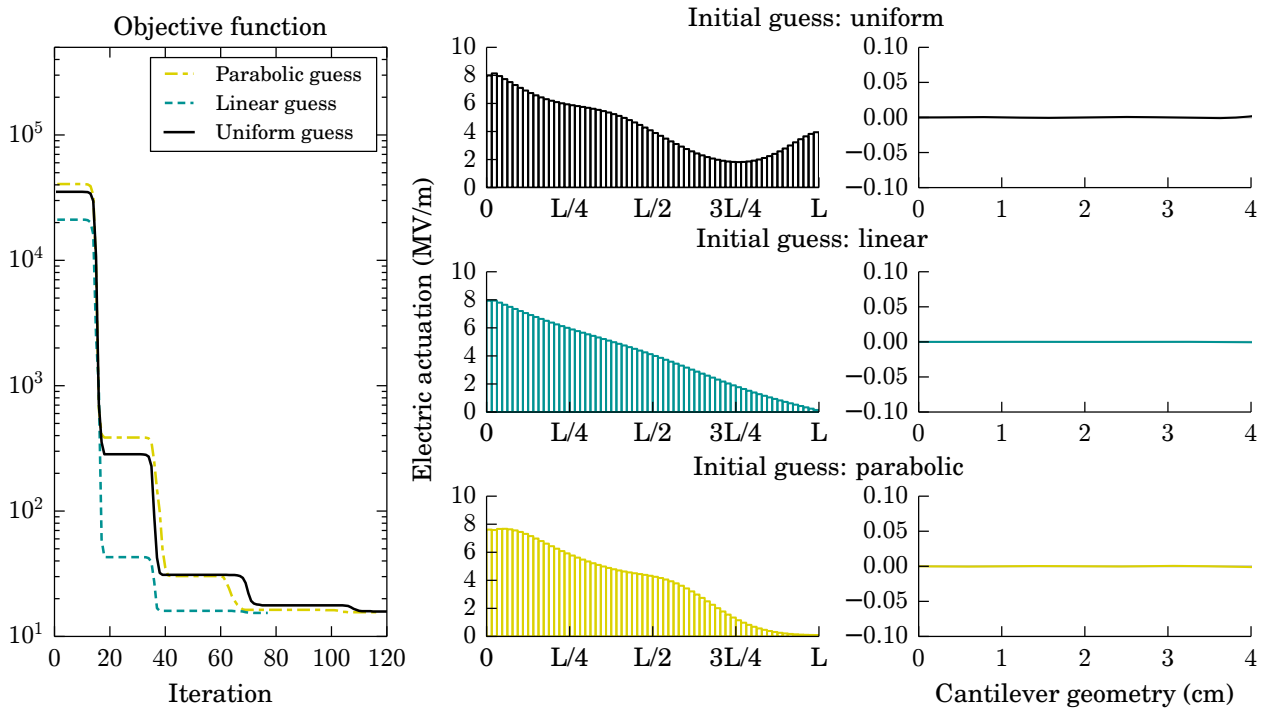


Figure 7.14: Time history of the optimisation (left), optimised actuation (centre) and final shape of the bimorph (right) for an uniform, linear and parabolic initial guess in the actuation distribution, and 64 actuators.

are shown in Tab. 7.1. It can be observed that the minimum value of J corresponds to the linear distribution. This agrees well with the assumption in linear regime that the moment distribution is parabolic, as the electric stress is quadratic with the electric field as shown in (4.1b).

Table 7.1: Value of the objective function for different initial guesses and number of actuators.

Number of actuators:	1	8	64		
Initial guess:	(a) Uniform		(b) Linear	(c) Parabolic	
Objective function J	1269.89	17.51	15.85	15.44	15.56

7.4 Optimal Mass Distribution in Dynamic Conditions

We now tackle the problem of optimal design of flexible cantilevers under dynamic conditions. The settings of the optimisation problem are described in section 6.1.3, along with the demonstration of the accuracy of the gradients obtained using the ADDA method. We now study the ability of the adjoint-based optimisation framework to provide an optimised distribution of density for the cantilever in Fig. 6.11. Our objective is to damp out the vibrations of the

cantilever as fast as possible defining the distribution of inertia for a given value of damping, c_d in (6.11), within a time window of 1 s. The 40 design variables, $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_{40}]$, multiply the reference density $\rho = 100 \text{ kg/m}^3$. The objective function is defined as

$$J(\mathbf{u}) = \frac{1}{t_F} \sum_{i=0}^{t_F} \left[\sqrt{(\mathbf{u}_i - \mathbf{u}_{tgt}) \cdot (\mathbf{u}_i - \mathbf{u}_{tgt})} + 10^3 \left(\frac{\sum_{i=1}^{n_\alpha} \alpha_i}{40} - 1.0 \right)^2 \right], \quad (7.3)$$

where the first term in the sum evaluates the distance from the tip of the cantilever at time t_i with respect to the target position, $\mathbf{u}_{tgt}$, which corresponds to the undeformed configuration in this case. The second term enforces that the total amount of density (or mass, as the volume is maintained constant) remains unchanged throughout the optimisation process. This is necessary to avoid trivial solutions such as the complete removal of the mass from the cantilever - which would lead to a steady-state solution with no dynamics involved.

The reference and the optimised behaviour and density distribution are presented in Fig. 7.15 for values of $c_d \leq 1 \%$. In these cases, the damper is unable to damp out the vibrations within the time window of study, resulting in an *underdamped* behaviour. In order to optimise the distribution of density, the optimiser takes the mass away from the tip of the cantilever, which is the region which introduces the largest inertial effects, and distributes it evenly next to the root. The optimal redistribution of density is able to bring the cantilever behaviour to a *critically damped* situation, for the case in which the $c_d = 1 \%$.

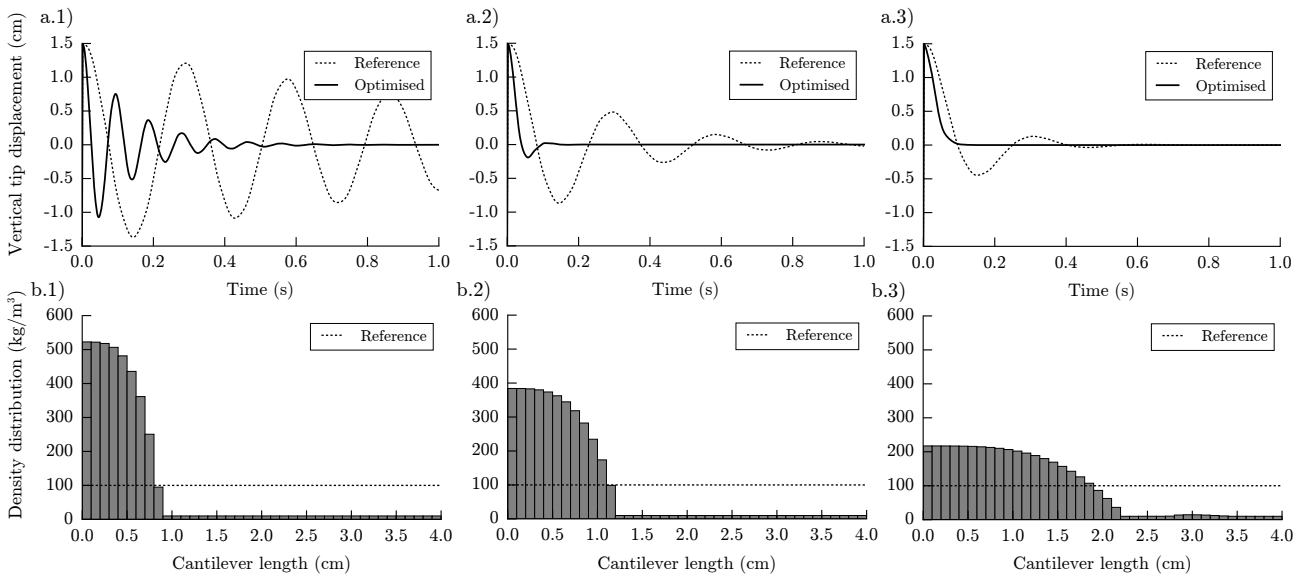


Figure 7.15: Time history of the tip displacement (a) and optimised distribution of density (b), for the following values of c_d : (1) $c_d = 0.1\%$ (2) $c_d = 0.5\%$ (3) $c_d = 1\%$.

A similar behaviour is observed for the case of $c_d = 2 \%$. The reference case is very close

to a *critically damped* behaviour, and the optimiser manages to redistribute the density in a way that the tip displacements are damped immediately. In this case, however, some mass is maintained at the tip, in the region near the actuation of the damper, as shown in Fig. 7.16.

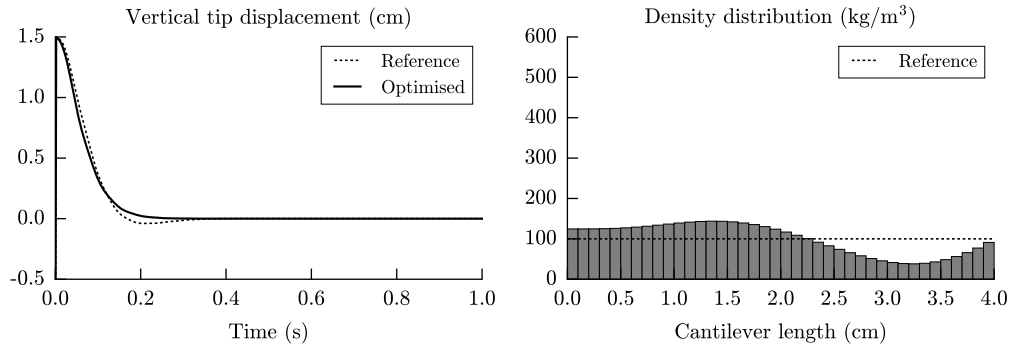


Figure 7.16: Time history of the vertical tip displacement (left) and optimised distribution of density (right), for the $c_d = 2\%$.

Similar characteristics, although magnified, are observed when the damping is *critical* without any density redistribution. This happens in the region of $c_d \in (2.5 - 4.5)\%$, as presented in Fig. 7.17. It can be observed that for the cases $c_d = 2.5\%$ and $c_d = 4.5\%$ the optimised distribution is very similar, with the mass clustered near $x = L/2$ and $x = L$. Although the total value of the averaged objective function J is reduced, this reduction is negligible given that the behaviour of the cantilever barely changes, as it can be observed from the tip displacements shown in Fig. 7.17,a.1) and a.3).

For the intermediate case, $c_d = 3.5\%$, the optimiser distributes the mass towards the tip of the cantilever. From the vertical tip displacements in Fig. 7.17,a.2), this leads to a smaller damping when the cantilever is released (as there is a larger mass at the tip, that is, a larger inertia), however, the slope of the damped stage is increased (as the mass is larger, so is the velocity and therefore the damping force). The total value of J , as for the other two cases, remains almost unchanged.

Finally, for the cases where the damping is super-critical, the optimiser finds a new mass distribution which is almost constant within range $c_d \in [5, 8]\%$. These cases are already *overdamped*, meaning that the tip of the cantilever does not reach the equilibrium position within the time window considered. The optimiser distributes the mass increasing it in the first half and then linearly decreasing it until the tip, as shown in Fig. 7.18

The results shown in this section are summarised in Fig. 7.19. It can be observed that while large reductions in the value of J have been obtained in the *underdamped* region, and moderate improvements were also obtained in the *overdamped* region, the value of J remains almost

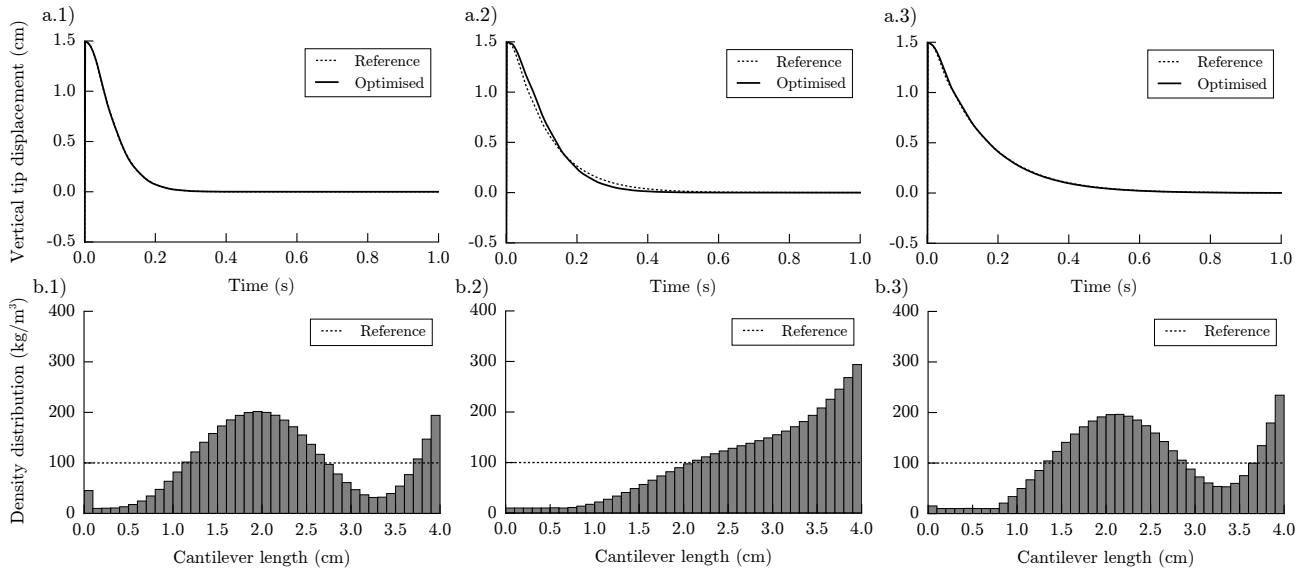


Figure 7.17: Time history of the vertical tip displacement (a) and optimised distribution of density (b), for the following values of c_d : (1) $c_d = 2.5\%$ (2) $c_d = 3.5\%$ (3) $c_d = 4.5\%$.

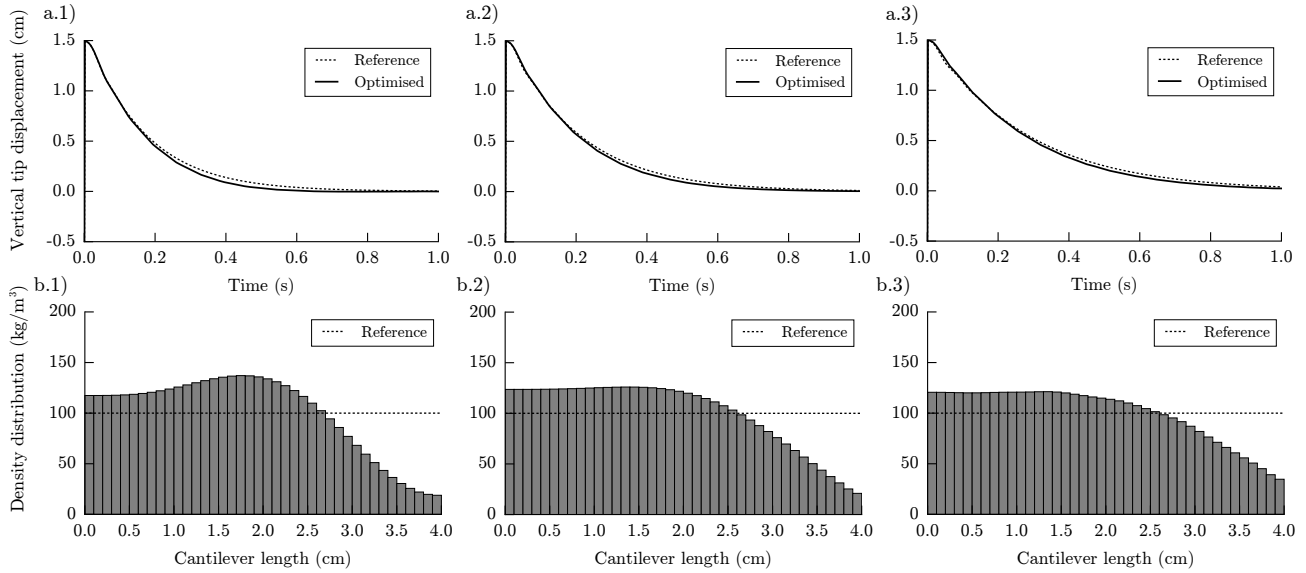


Figure 7.18: Time history of the vertical tip displacement (a) and optimised distribution of mass (b), for the following values of c_d : (1) $c_d = 5E-02$ (2) $c_d = 6E-02$ (3) $c_d = 8E-02$.

unchanged for the *critically damped* cases. The results presented in this section demonstrate the potential of adjoint-based gradient optimisation in dynamic structural problems with multiple design variables.

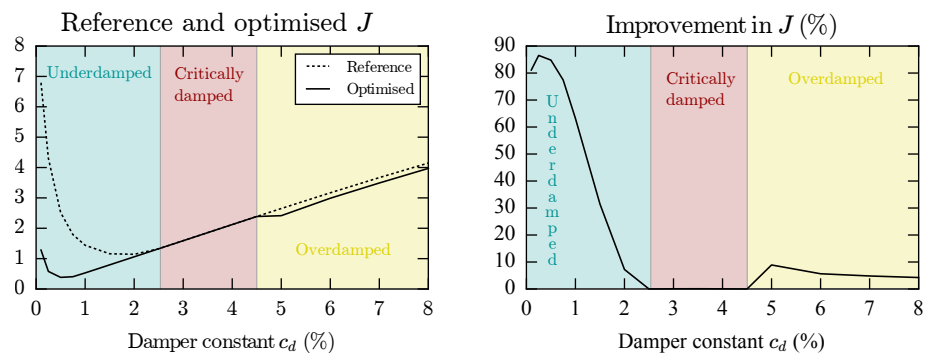


Figure 7.19: Reference and optimised values of J (left), and % reduction in J obtained after the optimisation (right).

Chapter 8

Optimal Design and Actuation of Structures under Fluid Loads

In this chapter, the goal is to explore the applicability of the coupled ADDA method to fully-coupled FSI problems involving simultaneously large displacements and viscous effects. In order to do so, the optimal design and actuation of vertical walls immersed in horizontal flows has been studied. Although this problem involves a simplified geometry, its choice is justified as it poses no restrictions to the complexity of the governing equations while reducing preprocessing and computational effort. In section 8.1, the physics of the target design space before optimisation are investigated. Section 8.2 covers the optimal design of the wall using the structural stiffness as design variable. Finally, section 8.3 shows the applicability of the method in an optimal actuation problem using electromechanical means.

8.1 Reference configuration

The parametrised FSI domain is defined in Figure 5.9.

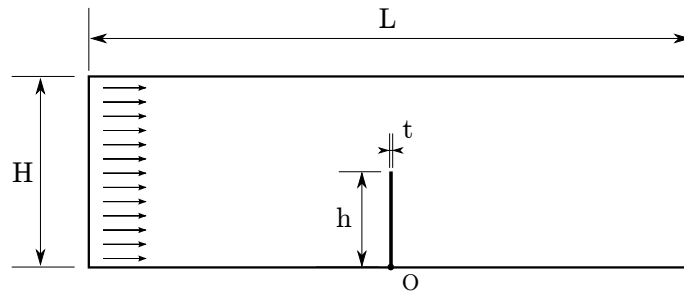


Figure 8.1: Parametric FSI domain.

The closed-channel in Figure 8.1 is formed of two slip walls in the upper and lower boundaries. For an incompressible flow, the distance between the top surface of the wall and the upper boundary determines the speed of the flow through the gap, and ultimately the total forces to which the wall will be subject. We solve for two different fluid domain geometries, defined in Table 8.1 as test cases D16 and D2. In the first case the effects of the wall to the flow in the channel will remain local, as the upper boundary of the channel remains far from the wall and this is relatively small in comparison to the flow domain. As the total height of the channel is reduced, the height of the wall becomes comparable to the size of the gap, and this leads to higher drag forces acting upon the structure.

Table 8.1: Geometrical description.

Test case	H	L	h	t
D16	$16 \cdot h$	$48 \cdot h$	1.6 cm	0.1 cm
D2	$2 \cdot h$			

The flow enters the domain through the inlet in the left boundary in Figure 8.1. Both the horizontal flow velocity and the flow density are unity, this results in a Reynolds number $Re=10$ based on h . Consequently, the flow conditions are steady-state with a laminar separation at the wall edge and the subsequent reattachment, as shown in Fig. 8.2. In this test case, the artificial compressibility solver available in SU2 [125] has been used for the incompressible flow.

The structural properties of the problem are shown in Table 8.2. We define two reference test cases with different values of the Young's modulus, E , namely E20 and E40, and we adopt a neo-Hookean hyperelastic material model for the structural domain using a plane strain description for the 2D model.

Table 8.2: Structural properties.

Test case	E_{ref}	ν
E20	20 kPa	0.4
E40	40 kPa	

The wall undergoes large displacements, particularly for the case in which the stiffness is lower (E20) and the flow domain is smaller (D2), which results in a higher flow velocity and loading upon the wall (Fig. 8.2). In this case, the tip displacement reaches a 60% of the height of the wall. A stiffer wall (E40) is able to reduce the tip displacement, and a larger flow domain (D16) reduces the loads due to the larger gap through which the flow can circulate (15h vs 1h).

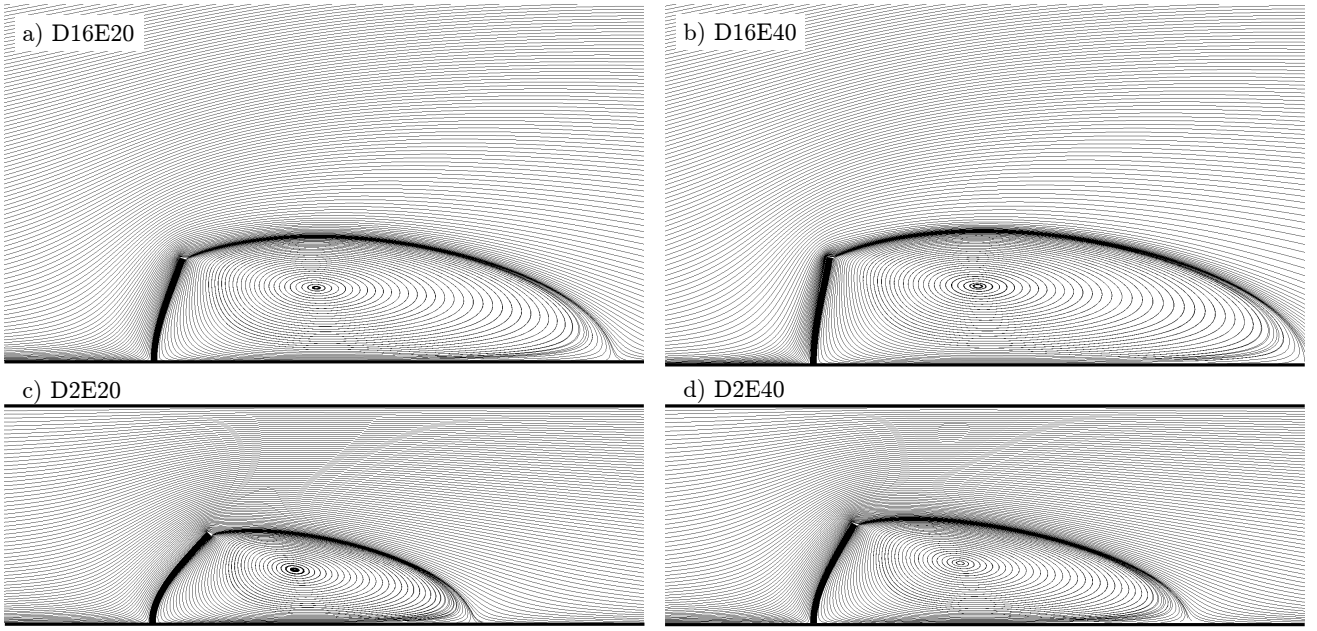


Figure 8.2: Streamlines and wall deformation for the four nominal cases.

8.2 Optimal Design of Vertical Wall in Viscous Flow

In this section, we tackle the optimal design of the coupled problem described in section 8.1. The objective is to redistribute the stiffness in the wall in order to achieve a predefined geometry, $\mathbf{u}_{tgt}$, in the coupled system static equilibrium. The distribution of the design variables and the target geometry, $\mathbf{u}_{tgt}$, that the optimiser will try to retrieve by minimizing the objective function in (1.2), are shown in Fig. 8.3. The wall is discretized into a total of 32 regions, numbered from bottom to top (Fig. 8.3a). The vector of design variables, $\boldsymbol{\alpha}$, multiplies the reference Young's modulus (E_{ref}) for each test case,

$$\mathbf{E} = \boldsymbol{\alpha} \cdot E_{ref} = [\alpha_1, \dots, \alpha_{32}] \cdot E_{ref}. \quad (8.1)$$

We define as a target geometry a shape $\mathbf{u}_{tgt}$ with a deformation of a $0.3h$ at the top of the wall. For replicability purposes, the exact target geometry will be shared online in the aeroelastics group's website¹. The values of the objective functions $J(\mathbf{u})$ for the four possible combinations of the test cases described in Table 8.1 and Table 8.2, are presented in Table 8.3 for the initial value of $\boldsymbol{\alpha}_0 = [1.0, \dots, 1.0]$. In Fig. 8.4, the deformed geometry of the four cases is shown and compared with the target geometry $\mathbf{u}_{tgt}$.

We employ the Python framework described in chapter 7 in order to set up the optimisation

¹<http://www.imperial.ac.uk/aeroelastics>

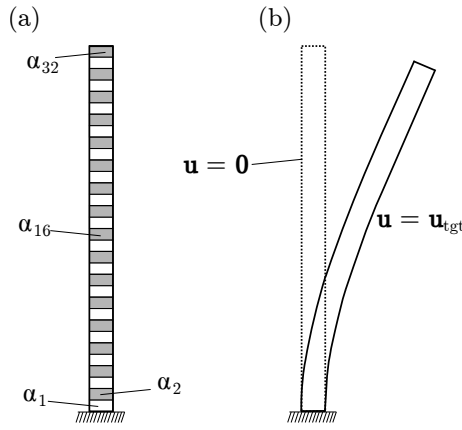
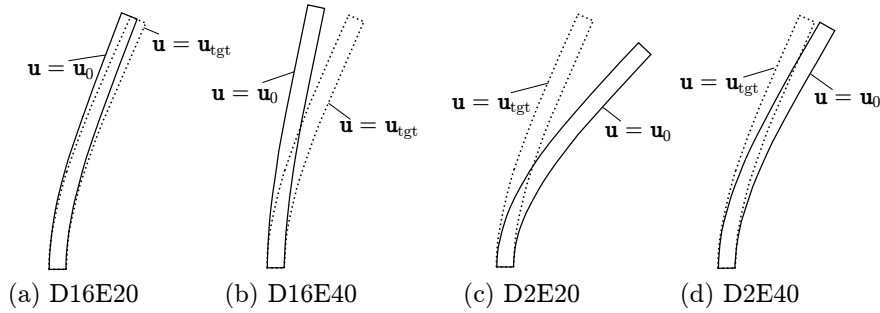


Figure 8.3: Design variables (a) and target geometry (b).

Table 8.3: Values of the objective function J for α_0

Test	D2E20	D2E40	D16E20	D16E40
$J(\mathbf{u})$	4.344E-01	4.701E-02	8.871E-03	1.645E-01

Figure 8.4: Deformed geometries for α_0

problem. This environment will selectively call the primal and adjoint solvers available in SU2. The values of the objective function and the gradient will be used by the SLSQP optimiser available in SciPy [213] to update the design variables within given bounds, and drive the optimisation.

We will use two different objective functions for the optimal design. For the first one, that we will call *unconstrained* for convenience, we do not incorporate any limitation in the values of the design variables apart from an upper and lower boundary. Therefore, the optimiser is allowed to modify the values of α within the range $[0.1, 10]$, in order to minimize $J_{uncons}(\mathbf{u})$ (8.2). This means physically that there is an unlimited availability of material,

$$J_{uncons}(\mathbf{u}) = (\mathbf{u} - \mathbf{u}_{tgt})^T \mathbf{C}(\mathbf{u} - \mathbf{u}_{tgt}). \quad (8.2)$$

It can be observed from Fig. 8.4 that the deformation for α_0 is larger than the target deformation for the cases with a smaller domain, D2. This is because the loads are larger in this

case, and therefore, the total stiffness of the structure is expected to be larger at the optimum than for $\boldsymbol{\alpha}_0$. The situation is opposite for the D16 cases. The domain is larger, which induces smaller fluid loads. The initial deformation is for these cases smaller than $\mathbf{u}_{tgt}$, and a reduction in the total stiffness is expected from a preliminary analysis.

For the second case, we aim to enforce that the total stiffness in the structural domain remains constant. This property mathematically resembles that the total amount of material in the wall remains unchanged, and therefore intends to distribute the material in an optimised way. In order to do so, we define the total variation of the design variables, $\Delta\boldsymbol{\alpha}_{ref}$, as

$$\Delta\boldsymbol{\alpha}_{ref} = \frac{\sum_{i=1}^{n_\alpha} \alpha_i}{n_\alpha} - 1.0, \quad (8.3)$$

and we incorporate it as a penalty term to the objective function, which we will call *constrained* or $J_{cons}(\mathbf{u}, \boldsymbol{\alpha})$, as

$$J_{cons}(\mathbf{u}, \boldsymbol{\alpha}) = (\mathbf{u} - \mathbf{u}_{tgt})^T \mathbf{C}(\mathbf{u} - \mathbf{u}_{tgt}) + k (\Delta\boldsymbol{\alpha}_{ref})^2, \quad (8.4)$$

where k is a penalty parameter that has been set in these calculations to $k = 10^3$.

Therefore, the optimiser will attempt to optimally redistribute the stiffness within the structural domain in order to minimise the deformation with respect to the target value $\mathbf{u}_{tgt}$. This problem may indeed not have a solution below a certain amount of total stiffness.

The results for all cases will be presented next. In section 8.2.1, the larger domain will be studied, where the distribution of stiffness for the *constrained* and the *unconstrained* yields similar results. In section 8.2.2, the *constrained* case fails to achieve $\mathbf{u}_{tgt}$. The reasons for this will be discussed.

8.2.1 Wide channel (D16)

First, we present the results for the larger domain, D16. In this case, as the fluid loads are smaller, the baseline design is stiffer than the target geometry. The optimisation history, and the optimised results for the constrained and unconstrained objective functions are presented in Fig. 8.5 for the test case D16E20. It can be observed that the simulation is successful in both cases, and leads to a very similar distribution of stiffness using either objective function. The

penalty term added to $J_{cons}(\mathbf{u})$ enforces the total stiffness to remain constant, which in turn leads to a larger stiffness at the tip of the cantilever in order to compensate for the reductions required towards the bottom.

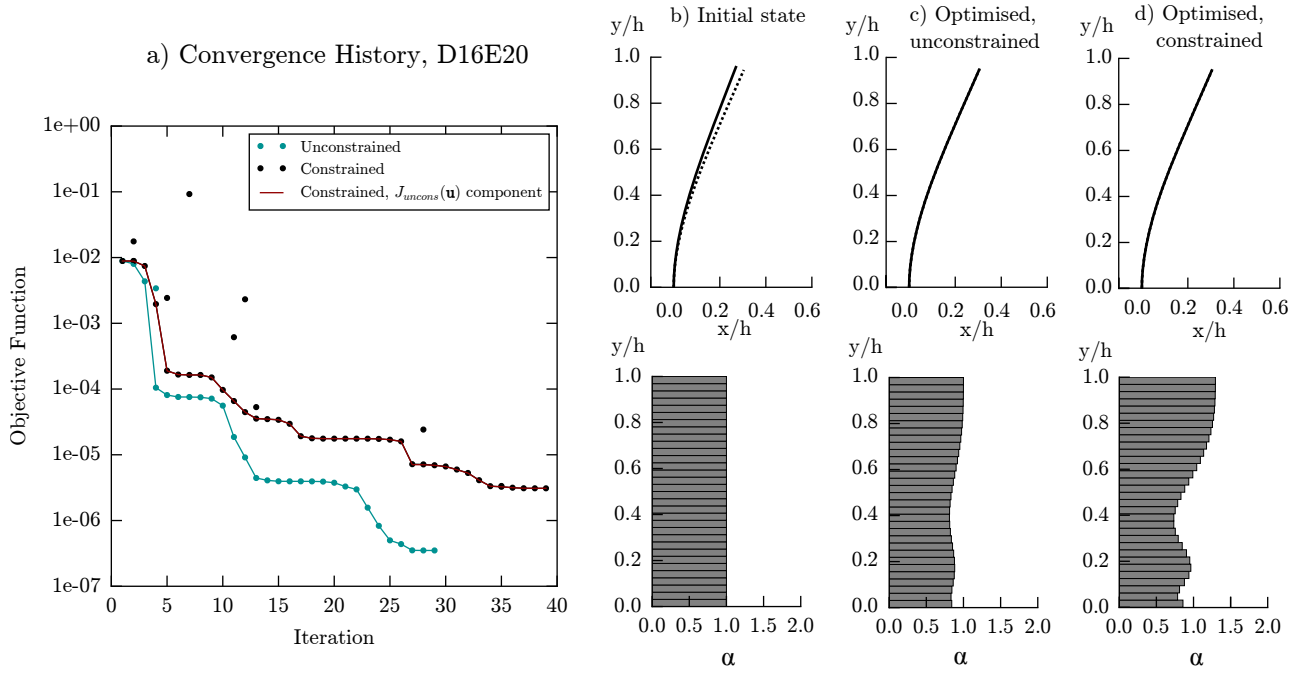


Figure 8.5: Optimisation history (a), initial state (b), and optimised geometries (top) and distribution of design variables (bottom) for the unconstrained (c) and constrained (d) test case D16E20

From Tab. 8.4, it becomes clear that $J_{cons}(\mathbf{u}, \boldsymbol{\alpha})$ successfully maintains the value of $\Delta\boldsymbol{\alpha}_{ref}$ low, as compared to over a 10% reduction in the total stiffness of the wall in the unconstrained case. This in turn increases the number of explorations on the design space that the optimiser requires to converge to an optimum, resulting in a larger number of objective function evaluations. The optimiser is unable to actuate over $\boldsymbol{\alpha}$ in an overly aggressive manner, as this normally will lead to the second parameter in (8.4) to be largely increased.

The performance of the method can be compared with the most common techniques to compute gradients. For 32 design variables, one single full optimisation loop involving the computation of the objective function and the gradient would require at least 33 evaluations of the primal solver using forward or backward differences, direct methods or the complex-step method, and up to 65 using central differences. The cost of solving the full optimisation problem using adjoint methods is, for this case, comparable to running just a couple of optimisation iterations using other techniques.

For D16E40, the optimisation process seems to fall into a local minimum, where still some differences between $\mathbf{u}_{final}$ and $\mathbf{u}_{tgt}$ may be observed, although they are minimal (Fig. 8.6). This is a common issue in gradient-based optimisation. For the unconstrained case, a different value

Table 8.4: Results for the D16E20 case.

Objective Function	Obj. Function Evaluations	Gradient Evaluations	$\min J(\mathbf{u})$	$\Delta\alpha_{ref}$
$J_{uncons}(\mathbf{u})$	30	29	3.50E-07	-10.33%
$J_{cons}(\mathbf{u}, \boldsymbol{\alpha})$	48	39	3.09E-06	+0.00%

of the initial value of the design variables $\boldsymbol{\alpha}_0$ would have led to a better solution, and indeed, starting the optimisation with $\boldsymbol{\alpha}_0 = [0.5, \dots, 0.5]$ leads to the solution of D16E20. However, this is no longer true for the constrained value, where $\boldsymbol{\alpha}_0 = [0.5, \dots, 0.5]$ would be an unacceptable distribution of the design variables that would notably increase the value of $J_{cons}(\mathbf{u}, \boldsymbol{\alpha})$.

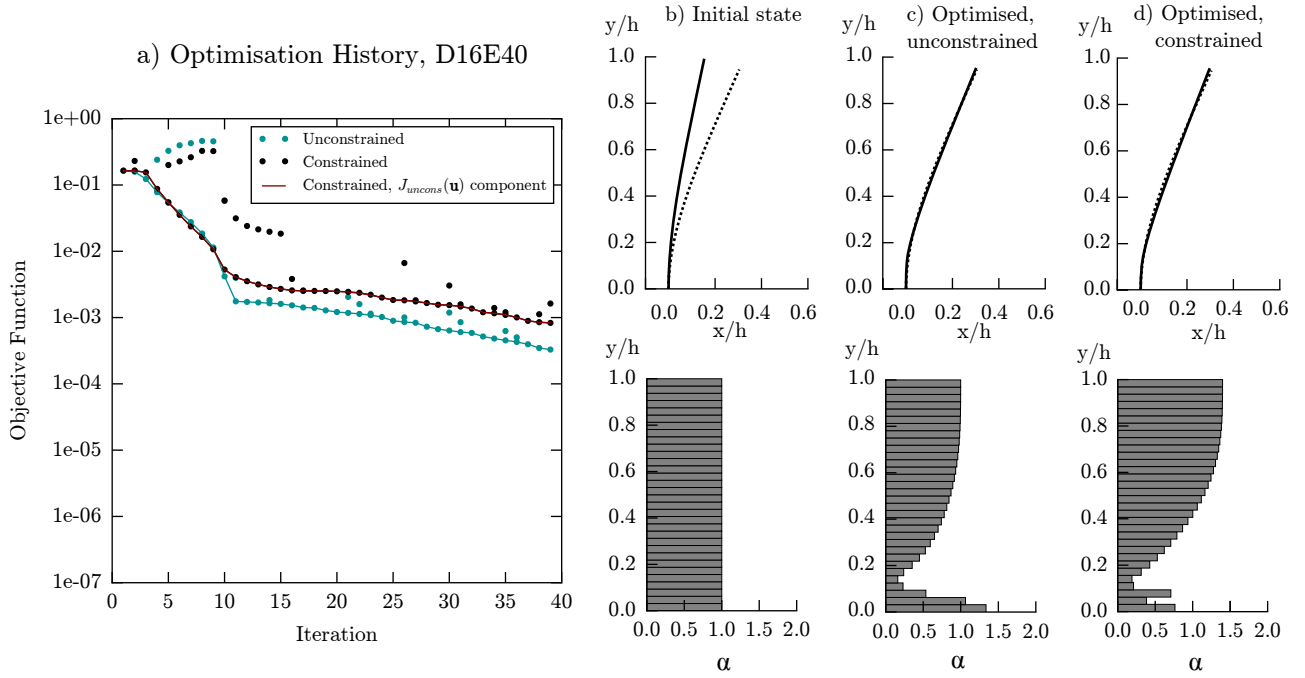


Figure 8.6: Optimisation history (a), initial state (b), and optimised geometries (top) and distribution of design variables (bottom) for the unconstrained (c) and constrained (d) test case D16E40

From Tab. 8.6, it is shown that the computational cost remains limited. We impose a stopping criterion of $\Delta J < 10^{-10}$. For a maximum limit of 39 gradient evaluations, the optimiser explores the global design space using between 56 and 62 objective function evaluations. Again, the application of the adjoint method limits greatly the number of evaluations of the primal solver in the optimisation process with respect to other methods to compute gradients.

Table 8.5: Results for the D16E40 case.

Objective Function	Obj. Function Evaluations	Gradient Evaluations	$\min J_{uncons}(\mathbf{u})$	$\min J(\mathbf{u})$	$\Delta\alpha_{ref}$
$J_{uncons}(\mathbf{u})$	56	39	3.29E-04	3.29E-04	-20.39%
$J_{cons}(\mathbf{u}, \boldsymbol{\alpha})$	62	39	8.29E-04	8.32E-04	-0.01%

8.2.2 Narrow channel (D2)

We now present the results for the smaller domain, D2. In this case, as the fluid loads are larger than for D16, which in turn results in a baseline design that is more flexible than the target geometry. The optimisation history, and the optimised results for the constrained and unconstrained objective functions are presented in Fig. 8.7 for the test case D2E20. For this case, it can be observed that the simulation is successful for the unconstrained case, but in turn the constrained case is unable to recover the target geometry.

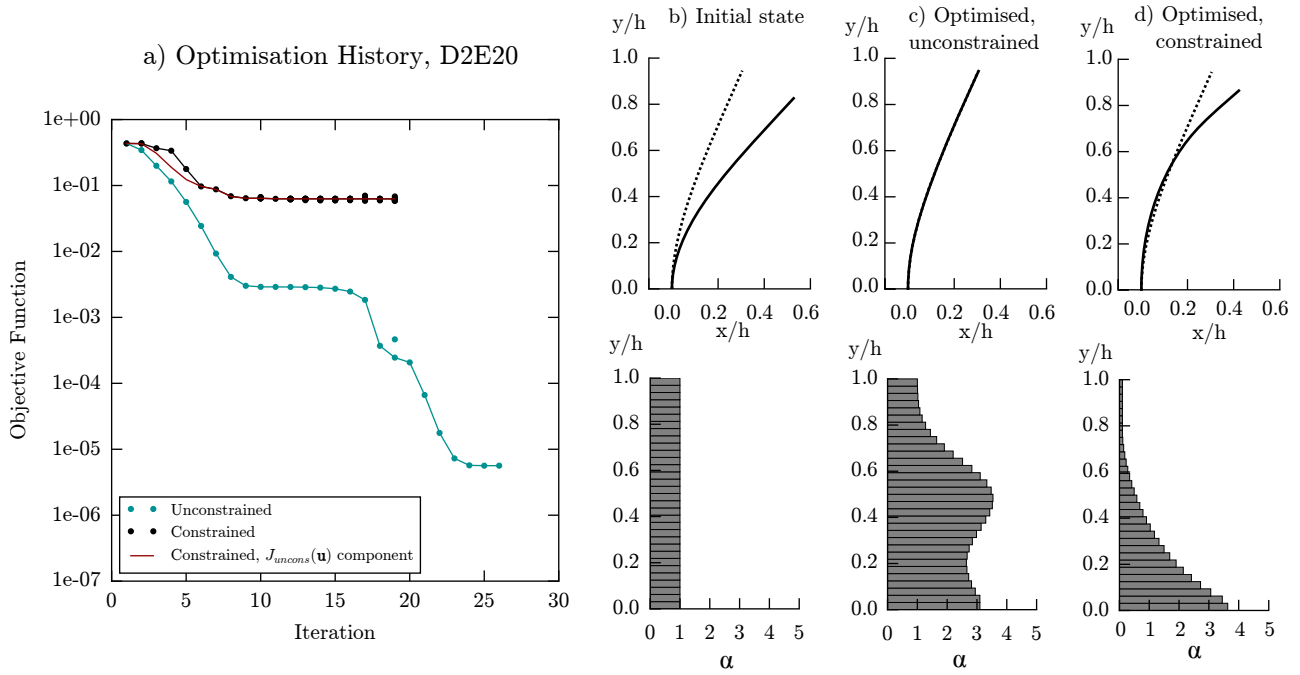


Figure 8.7: Optimisation history (a), initial state (b), and optimised geometries (top) and distribution of design variables (bottom) for the unconstrained (c) and constrained (d) test case D2E20

A summary of the results presented in Fig. 8.7 is shown in Tab. 8.6. The unconstrained case succeeds in finding an optimum with a small number of iterations, which results in very few evaluations of the primal and adjoint solvers. On the other hand, and although the optimisation stops for the constrained case after just 19 gradient evaluations, it required 111 evaluations of the objective function in order to explore a more complex design space. Even for this case, 111 evaluations of the objective function are just below 4 gradient evaluations using finite differences.

Four screenshots of the optimisation process for the unconstrained case D2E20 are shown in Fig. 8.8. Just 10 optimisation cycles are enough to noticeably stiffen the behaviour of the wall. At this first stage, the optimiser increased the values of α at the root of the cantilever,

Table 8.6: Results for the D2E20 case.

Objective Function	Obj. Function Evaluations	Gradient Evaluations	$\min J(\mathbf{u})$	$\Delta\alpha_{ref}$
$J_{uncons}(\mathbf{u})$	27	26	5.64E-06	+144.29%
$J_{cons}(\mathbf{u}, \boldsymbol{\alpha})$	111	19	6.27E-02	+0.01%

which successfully reduced the differences with respect to the target geometry, $\mathbf{u}_{tgt}$. This is because the sensitivities at the root are larger than at the tip, as shown for a similar problem in section 6.2.1, because of the larger moment generated by the fluid forces. However, some deviations in the curvature of the wall at approximately 40% of the height were still observed. This was tackled by the optimiser by reducing the stiffness at the root while increasing it at approximately 50% of the height, also exploiting the features of the incoming flow. The final solution very accurately resembles the chosen target geometry.

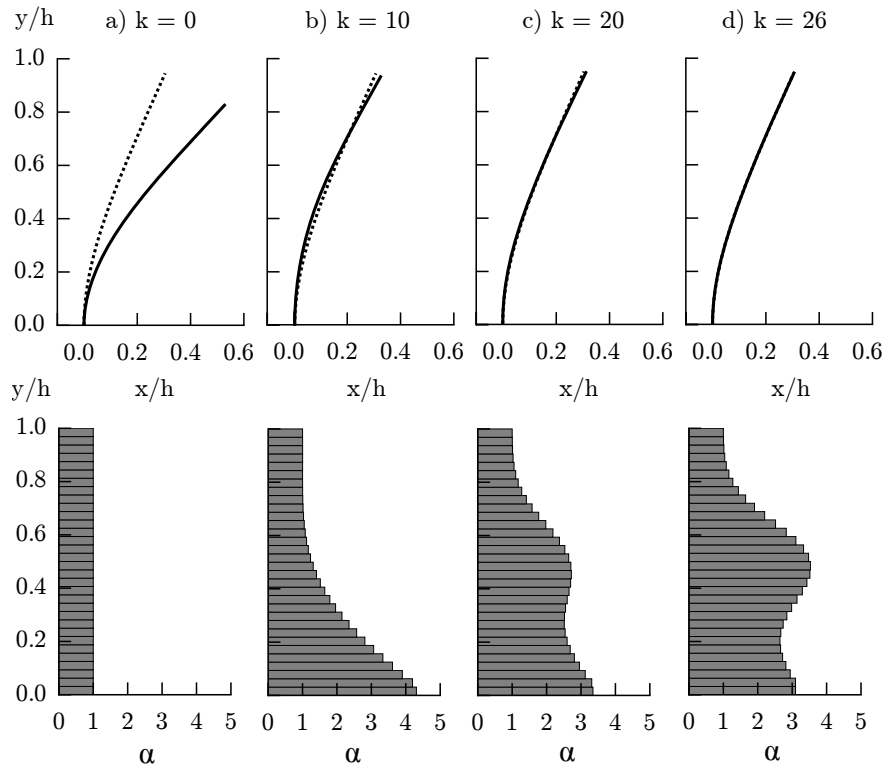


Figure 8.8: Test case D2E20, unconstrained. Deformed geometries (top) and value of the design variables (bottom) for the initial guess, two intermediate optimisation stages ($k=10$ and $k=20$), and the final design.

On the other hand, the optimisation process for the constrained case D2E20 is shown in Fig. 8.9. During the first stages of the optimisation process, the optimiser successfully redistributes the stiffness to reduce the deformation towards the root of the wall as in the previous case. However, as the total amount of stiffness is required to remain constant, stiffening the root requires making the tip of the cantilever more flexible. As a result, the optimiser is unable

to retrieve the target geometry in the area roughly above a 60% height of the wall.

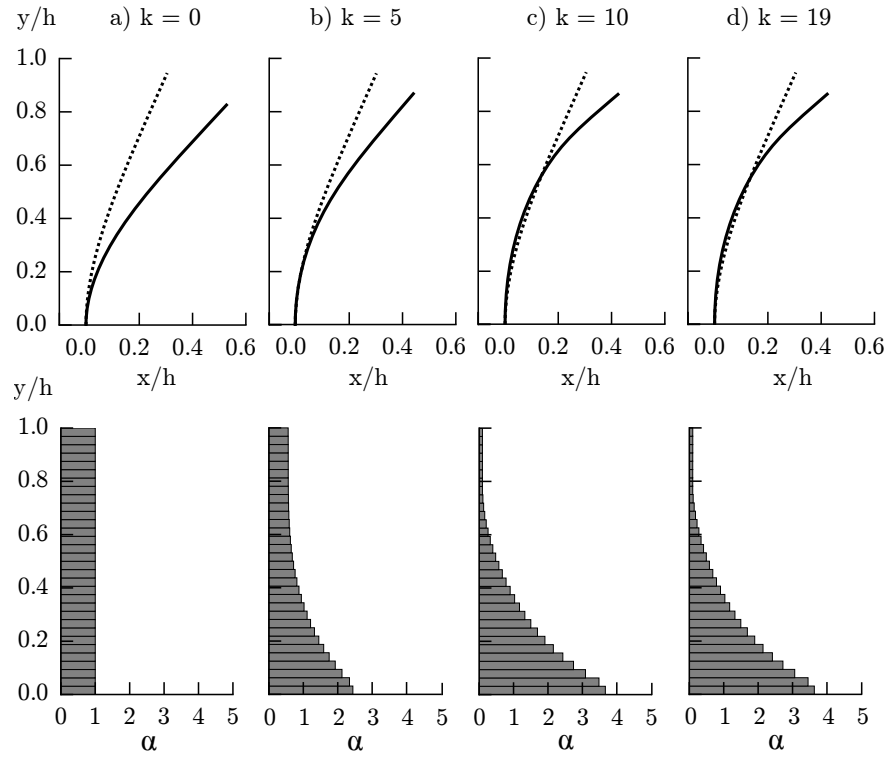


Figure 8.9: Test case D2E20, constrained. Deformed geometries (top) and value of the design variables (bottom) for the initial guess, two intermediate optimisation stages ($k=5$ and $k=10$), and the final design.

The previous result could be expected as the optimal solution for the unconstrained case required the total stiffness to be increased by more than 140%, as shown in Tab. 8.6. However, the optimised shape in Fig. 8.9 does not resemble the shape obtained for the unconstrained optimisation in Fig. 8.8. In order to test that the solution in Fig. 8.9 does not correspond to a local minimum, the unconstrained optimised distribution is rescaled in order to meet $\Delta\alpha_{ref} = 0.0$, and used as the initial guess in the constrained optimisation process.

The results for this new optimisation are presented in Fig. 8.10. It can be observed that the extra stiffening in the region between 40 and 60% of h leads to a slightly different distribution of stiffness. Although this new distribution yields a closer geometry in the region below $0.4h$, the value of the objective function is for this case $J = 0.067$, that is, a worse solution compared to $J = 0.063$ in Fig. 8.9. This is due to a poorer redistribution of the available resources: the tradeoff for a larger stiffness in the central region of the wall means that it remains too flexible at the root. Once allocated as much stiffness as possible to the root, there is no remainder left to stiffen the tip of the wall.

A similar behaviour is observed for the optimal solution obtained for the case D2E40 (Fig. 8.11). Again, the final solution for the unconstrained case matches the objective proposed, by defining

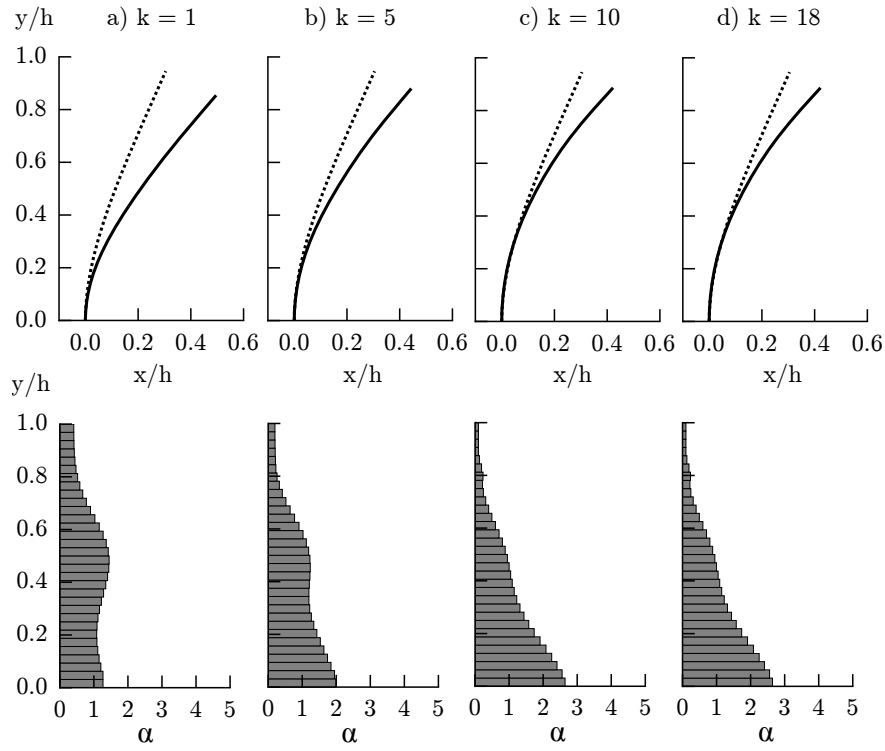


Figure 8.10: Optimised geometries (top) and distribution of design variables (bottom) for a constrained D2E20 test case with an initial guess based on the unconstrained result.

a distribution of stiffness with two stiffer regions, one towards the root and the other at $0.4h$. In fact, it very closely resembles the shape of the unconstrained solution for D2E20 scaled by $1/2$, which matches the intuitive behaviour, so this appears to be a global minimum. On the other hand, the constrained case fails to provide a distribution similar to the target geometry, $\mathbf{u}_{tgt}$.

The results from Fig. 8.11 are summarised in Tab. 8.7. As for D2E20, the unconstrained case finds a solution using a reduced number of evaluations of the solver, while the constrained requires a larger exploration of the design space.

Table 8.7: Results for the D2E40 case.

Objective Function	Obj. Function Evaluations	Gradient Evaluations	$\min J_{uncons}(\mathbf{u})$	$\min J(\mathbf{u})$	$\Delta\alpha_{ref}$
$J_{uncons}(\mathbf{u})$	38	36	8.05E-07	8.05E-07	+32.09%
$J_{cons}(\mathbf{u}, \alpha)$	121	39	1.45E-03	1.48E-03	+0.08%

We test again the constrained solution by restarting the process with an initial guess that corresponds to the optimised distribution of the unconstrained case rescaled to meet $\Delta\alpha_{ref} = 0.0$. This is shown in Fig. 8.12. Again in this case, the optimiser trades the stiffness in the middle region for the stiffness at the root, in order to stiffen up the wall and approximate its geometry to the target.

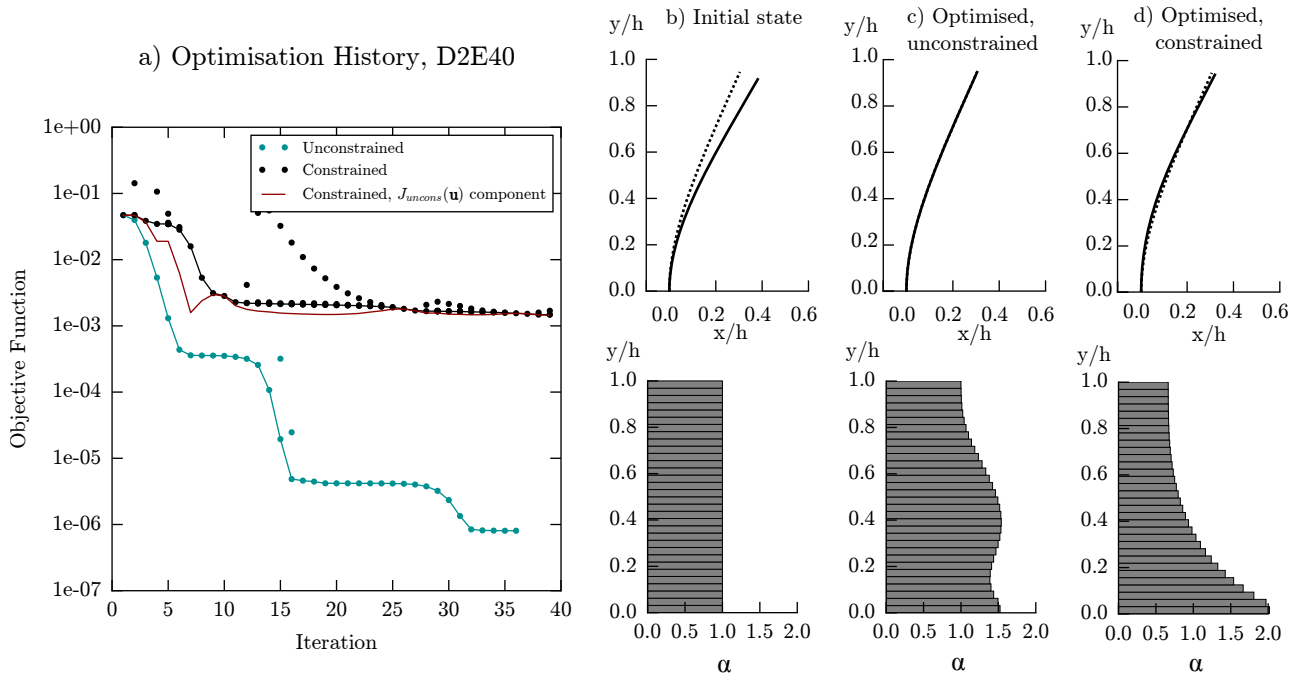


Figure 8.11: Optimisation history (a), initial state (b), and optimised geometries (top) and distribution of design variables (bottom) for the unconstrained (c) and constrained (d) test case D2E40

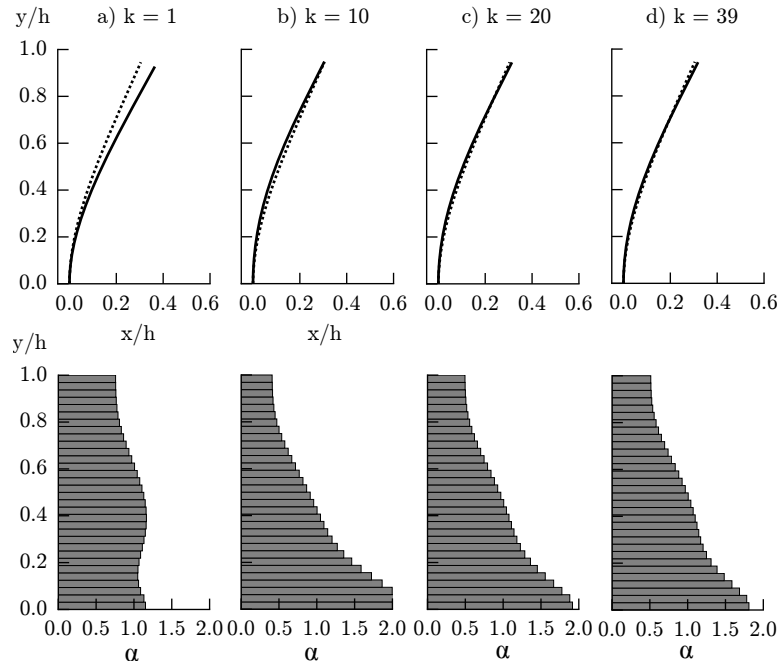


Figure 8.12: Optimised geometries (top) and distribution of design variables (bottom) for a constrained D2E40 test case with an initial guess based on the unconstrained result.

8.3 Optimal Actuation of Vertical Wall in Viscous Flow

In this final section, we tackle a problem with the same geometrical and fluid properties as in section 8.2. However, the mechanical properties of the cantilever are now modified, in order to incorporate electromechanical actuation to the wall. This introduces the additional complexity

of the electric material model, which needs to be accounted for by the ADDA method during the optimisation problem.

The wall is split into two regions. The upwind side remains unactuated, with a constant Young's Modulus $E_{ref} = 40$ kPa. The downwind side is now covered with a dielectric elastomer, divided into 32 actuation regions, as shown in Fig. 8.13a. The thickness of the actuator is $t_{act} = 0.05t$, while the unactuated region covers the remaining $t_{unact} = 0.95t$. The actuator is modelled using the ideal DE material model [196] and the method described in section 2, with the material relative permittivity is $\epsilon_r = 2.7$, the Young's Modulus $E_{DE} = 60$ kPa, and the Poisson ratio $\nu = 0.497$ [20].

The objective is now for the wall to recover its undeformed configuration, $\mathbf{u}_{tgt} = \mathbf{0}$. The actuation introduces a recovery moment in order to counteract the fluid loads (Fig. 8.13b). A new set of design variables is now defined such that it multiplies the modulus of the electric field in the reference configuration,

$$\tilde{\mathbf{E}}_{F,dist} = [\alpha_1, \dots, \alpha_{32}] \cdot \frac{V_{ref}}{t_{act}}, \quad (8.5)$$

thus obtaining a distributed recovery moment $M_{act}(y)$.

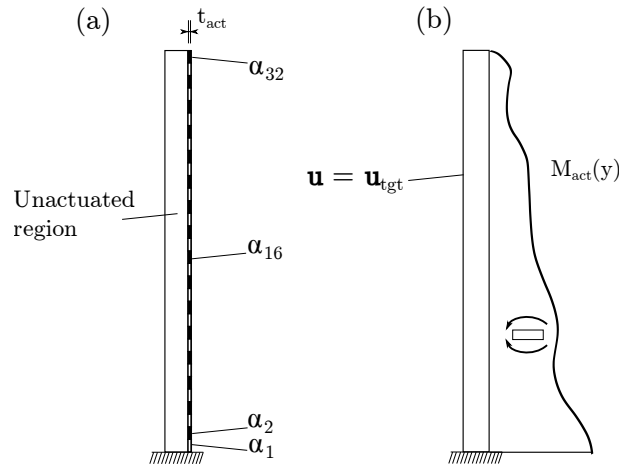


Figure 8.13: Design variables (a) and target geometry (b) with actuation mechanism.

The optimisation history is presented in Fig. 8.14, for an initial uniform actuation with $V_{ref} = 100$ V. The maximum voltage has been limited to 1 kV. We observe that, for the larger domain D16, the optimiser is able to successfully reduce the value of the objective function by almost four orders of magnitude. For the smaller domain, however, this is no longer the case, and the improvements on the objective function are much more modest, being reduced from a value of 0.806 to 0.296 as shown in Tab. 8.8. The optimisation has been rerun from an initial

guess of $\alpha_i = 5.0 \forall i$ in α_0 , which has been identified as D2**, yet the maximum actuation remains insufficient to recover the undeformed configuration of the wall.

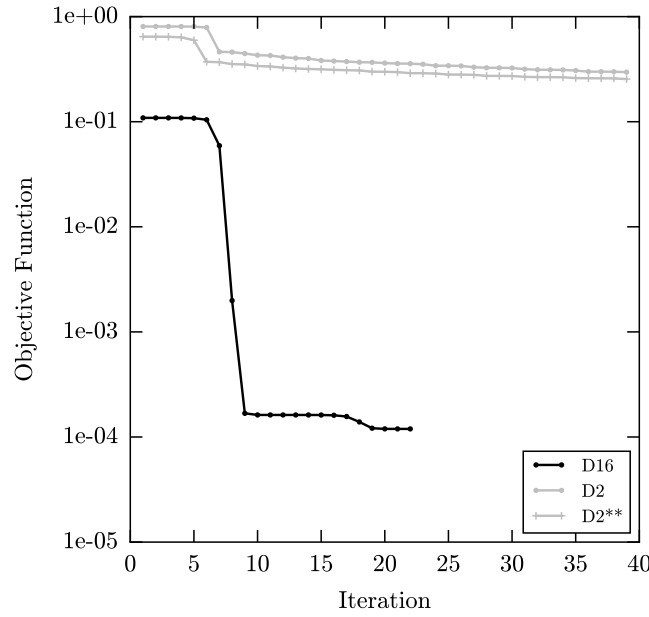


Figure 8.14: optimisation history for the actuated case.

The analysis of Fig. 8.15 provides a good hint as what is the cause for this behaviour. The load induced by the flow over the cantilever is larger in the smaller domain D2 and introduces a larger displacement into the cantilever, as compared to the larger domain D16. This has already been discussed in this chapter. For the domain D16, the optimiser successfully distributes the amount of actuation required for the cantilever to recover the undeformed geometry. In good agreement with the intuitive solution, the actuation needs to be stronger towards the root, in order to counteract the larger bending moment. A similar approach is taken by the optimiser for the case of D2; however, the limitation of maximum voltage admissible to 1 kV constrains the amount of the actuation in the root.

Table 8.8: Results for the actuated case. (** indicates that the simulation started from the initial guess $\alpha_i = 5.0 \forall i$ in α_0)

Test Case	$J(\mathbf{u})$ Evals	$\nabla J(\mathbf{u})$ Evals	$J(\alpha_0)$	$J(\alpha_{opt})$
D2	39	39	8.06E-01	2.96E-01
D2**	39	39	6.45E-01	2.55E-01
D16	23	22	1.08E-01	1.19E-04

The results presented in this chapter lead to some of the following concluding remarks. Gradient-based optimisation methods are able to tackle problems of optimal design and actuation of very flexible structures with a reduced number of optimisation cycles; however, it is of

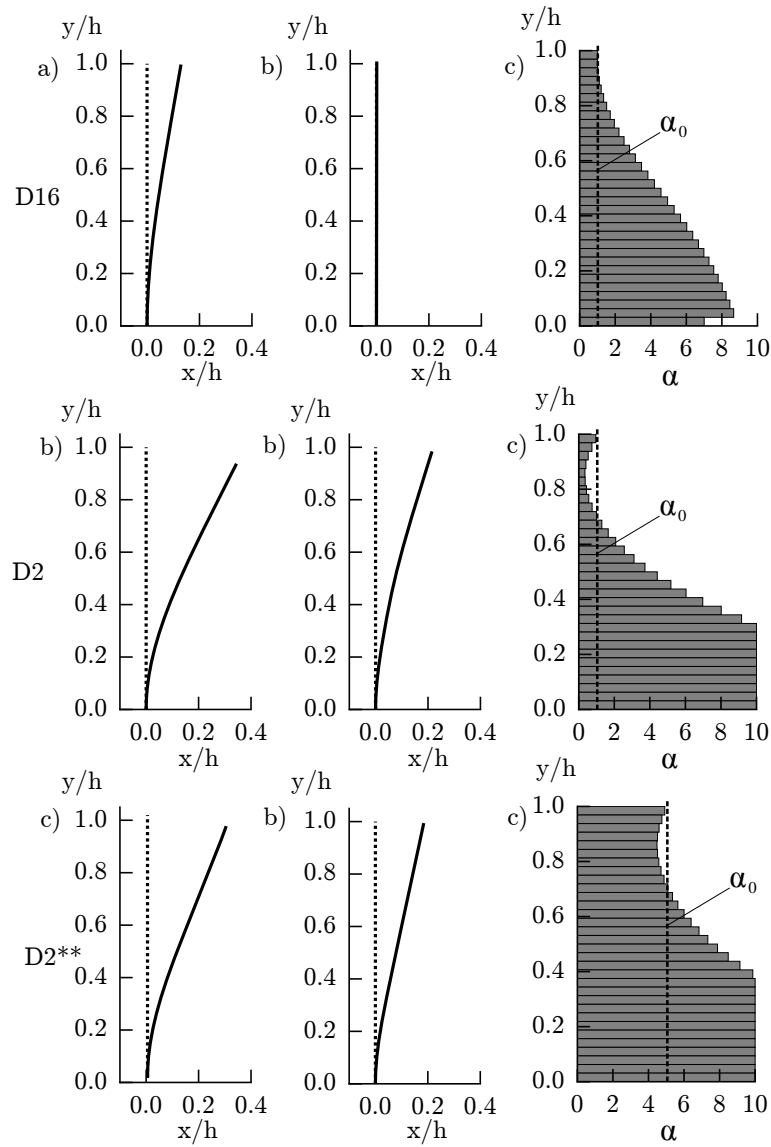


Figure 8.15: Actuated test case. Initial geometry (a), optimised geometry (b), and value of the design variables for the optimised geometry (c), for the cases with domains D16 (upper) and D2 (lower).

capital importance to use efficient techniques to compute the gradients, as the cost of the most extended methods grows fast with the number of design variables. The ADDA method has shown an excellence performance in the computation of gradients in complex, coupled problems. Although the geometrical definition of the problem has been simplified in this chapter, no simplifications were imposed in the complexity of the physics, which involve viscous effects, large deformations and complex material behaviour. The cost per evaluation of the gradient remains comparable to the cost of one evaluation of the objective function, independently of the number of design variables or the complexity of the underlying problem. This feature of the method will be greatly advantageous in the future study of industry-oriented optimisation problems.

Chapter 9

Conclusions

This final chapter covers the main achievements and outcomes of the works carried out in this thesis. Section 9.1 presents an overview of this dissertation and summarises the main accomplishments obtained during this work. Next, in section 9.2, the main research outcomes of the project are reviewed, and responses are provided to the research questions that were posed in section 1.3. Finally, further research that can be derived from this thesis is proposed in section 9.3.

9.1 Summary of Thesis Main Achievements

This dissertation has presented a coupled adjoint method for the calculation of gradients in high-fidelity, fully non-linear, computational Fluid-Structure Interaction problems. The method is based on the consistent application of the reverse mode of Algorithmic Differentiation to the fixed-point iterators of the sub-problems, building up a partitioned approach for the adjoint solution which closely resembles the primal solution method and is amenable to a Block Gauss-Seidel iterative scheme. To the author's knowledge this had not been achieved before using co-simulation, and it has the potential to facilitate complex industrial-quality design with no limitation on the governing equations in each discipline.

Chapter 1 covered the current state of the art of optimal design in the context of coupled FSI problems. A thorough review of the literature shows that the advances in this discipline over the last decades have led to a position in which these techniques can be successfully applied to very complex, industry-relevant problems. A significant research effort is currently being

placed in the development of efficient optimisation methodologies to find improved solutions to complex, multidisciplinary problems. In this context, the adjoint method has already been the focus of extensive research, given its major advantages in the computation of gradients and the complexity of its implementation in coupled problems.

The equations that govern fully non-linear, coupled Fluid-Structure Interaction problems are reviewed in chapter 2. This chapter presents the problem and an overview of the methods that have been used in this project to tackle it. The CFD solver chosen has been SU2, an open-source software suite. A non-linear, finite-element based structural solver has been implemented natively within SU2 in this project. The code structure has been discussed, and the implementation is verified in section 5.1. Then, the coupling conditions between fluid and structure are reviewed, and some changes are made into the solver structure to accommodate multi-physics into SU2. Finally, a partitioned solver is implemented for Fluid-Structure Interaction problems, which is later verified in section 5.2.

In chapter 3, the focus is put into the calculation of the gradients for the coupled problem. The strategy that has been followed for the gradient problem is similar to the one for the primal problem. The AD-based adjoint solver available in SU2 for the computation of gradients in aerodynamic shape optimisation problems was adopted and extended in this work for its application in standalone, fully non-linear structural problems. Its applicability to complex material behaviour such as electromechanical actuation was discussed in chapter 4. The structural adjoint was verified in section 6.1, and its applicability for gradient calculation in optimisation problems was shown through different examples in chapter 7. Next, in section 3.5 the extensibility of the AD-based strategy for fully-coupled, multidisciplinary FSI problems was studied. A novel approach developed in this work was presented, whose sensitivities were demonstrated via comparison to other methods in section 6.2. This new methodology was finally applied to coupled optimisation problems in chapter 8.

Therefore, the main achievements of this project can be summarised as follows:

- A non-linear structural solver able to deal with large deformations and complex material behaviour has been implemented natively, and coupled with the fluid solver in SU2 using an efficient strongly-coupled approach.
- The coupled solver has been verified using two low-Reynolds FSI benchmark test cases, a membrane wing undergoing large deformations and a dynamic cantilever interacting with a bluff-body wake.

- A static and dynamic structural adjoint solver has been derived and implemented in SU2, using AD-based techniques based on fixed-point iterators. Its ability to optimally design flexible structures has been shown.
- Electrical actuation using dielectric elastomers has been incorporated into the structural primal and adjoint solver in SU2, and later employed for optimal actuation of cantilever bimorphs.
- The AD-based adjoint solver has been consistently extended to fully-coupled FSI problems, using a novel coupled AD-based Discrete Adjoint (ADDA) methodology that abstracts the coupled adjoint solution from the internal physics of the sub-problems. This allows for the calculation of gradients in problems undergoing strong non-linearities and governing equations of arbitrary complexity.
- The accuracy of the coupled gradients computed using ADDA has been extensively demonstrated comparing them to other methods, and their benefits pointed out using a canonical example of optimal design and actuation of a very flexible wall immersed in viscous flows.

9.2 Main Research Outcomes

In chapter 1, the main motivations when attempting this research were summarised. Recent applications of the adjoint method demonstrated its potential in optimal design of very complex, industrial-scale FSI problems. However, the range of application of coupled adjoints remained limited due to the intricacy of their implementation and their close link to the physics of the problem. In a context where the trend moves towards more and more complex, involved representations of the “nonlinear reality” which are coupled Fluid-Structure Interaction problems, the implementation of the adjoint was still lagging behind the physical problem, and thus limiting the incorporation of these techniques to the industrial design chain.

In this work, the implementation of the coupled adjoint was abstracted from the constitutive non-linearities that affect each of the sub-fields, namely, the fluid and the structural problems. In order to reach this goal, state-of-the-art fluid adjoint methods based on Algorithmic Differentiation were observed as the starting point. It was shown that this AD-based concept could be

successfully applied to complex structural problems undergoing material and geometrical nonlinearities. Then, this idea was consistently extended in this thesis to coupled, multi-physics problems, by means of a mathematical reformulation of the problem based on Lagrangians and the KKT conditions. The coupled adjoint problem results in a system of fixed-point equations, which is amenable to a partitioned solution using Block Gauss-Seidel methods. The underlying physics remain opaque to the adjoint solver, which is not affected by new implementations or improvements to the primal solution methods. This concept limits the development and maintenance cost of coupled adjoint solvers, and eases their implementation into potentially more complex, optimisation frameworks for design in industry-relevant environments. On the other hand, the method here presented has only been verified in the context of having both the fluid and the structural solvers available on a consistent framework. Its applicability in other multi-solver structures remains yet to be explored.

This methodology has been implemented during this project into SU2, an open-source platform which is currently being actively developed by a large number of expert, multidisciplinary researchers from institutions across the world. The author found this context to be an invaluable environment for his research, as the suite is constantly updated with novel techniques in several areas of relevance. Although due to time limitations, the implementation of the structural and coupled solvers carried out in this work has been limited in some aspects, such as the available elements or the interfacing with the fluid solver, that is not the case for the physics of the problem, which have been kept as general as possible. Further works are expected to keep improving the primal solvers in order to be able to make the most of the proposed, coupled adjoint methodology.

The previous considerations lead to the following summary of **research outcomes**, which link directly to the research questions considered in section 1.3:

1. *What are the options to develop an adjoint method that does not require the exact analytical derivation of the Jacobian of the coupled FSI problem? What is the associated development cost?* It has been found in this research that, by reformulating the sub-solvers of the coupled problem into fixed-point iterators, it is possible to obtain an implicit representation of the adjoint variables that is not directly dependent on the Jacobian of the sub-problems, but on the partial derivatives of the fixed-point solvers. This idea has been very recently and successfully applied by Albring *et al.* [120, 121] and Zhou *et al.* [123, 124] to flow and aeroacoustic problems respectively, but this is, to the best of

the author's knowledge, the first application to coupled FSI problems. This approach has a very minimal impact associated to the characteristics of the solvers. It only requires to be able to identify the *output* variables of the fixed-point operator, that is, $\mathbf{x}_{output} = \mathbf{G}(\mathbf{x})$, and the *input* variables, namely, the variables with respect to which we intend to differentiate the solver, $\partial \mathbf{G} / \partial \mathbf{x}_{input}$.

2. *What would there be the main features of a computational framework that could support and integrate such a methodology?* In this project it has been demonstrated that SU2 is an outstanding framework to provide support for multi-disciplinary projects intending not only to carry out analysis, but also design of complex problems. The structural solver integrated well into an originally fluid-only framework, and although some refurbishment of the code was necessary, coupled problems were also incorporated in a limited time frame. The state-of-the-art adjoint features available in the suite were smoothly extended for coupled problems, in proof that collaborative work has several positive effects in the solution of novel problems. Algorithmic Differentiation has proven to be an invaluable tool in the differentiation of the fixed-point solvers. In parallel with the work in this thesis, SU2 was being natively coupled to an AD tool, CoDiPack, by the members of the Chair of Scientific Computing at TU Kaiserslautern. This tool was available for use in any other solver implemented within the suite, and was swiftly incorporated in this work for the calculation of adjoints in structural and coupled FSI problems. It provides a very efficient methodology which remains opaque to the physics of the problem being differentiated.
3. *Is there any limitation in the derivation of the adjoint imposed by the complexity of the sub-problems?* Because of the features explained in the previous point, the derivation of the adjoint has become independent of the complexity of the sub-problems. It only required a primal solver that is convergent in a system that is contractive, in order to inherit its convergence features and to retrieve its solution to initialise the adjoint problem.
4. *How would this method integrate new developments, such as complex material behaviour, with a contained maintenance cost?* As a result of the previously defined properties, the coupled adjoint solver proposed in this thesis immediately accounts for arbitrary problem non-linearities, and any new incorporation to the primal solver, such as for example the electromechanical actuation proposed in chapter 4, is accounted for by the AD-tool and

the coupled adjoint framework. This is an extremely powerful feature, which permits to engage into further development of the solvers, being able to incorporate them to the design chain straight away.

5. *What are the advantages of coupled adjoint methods in the resulting system performance?*

The computation of the coupled adjoint permits to incorporate additional gradient information to the optimisation problem with a relatively low added cost. The method demonstrates its efficiency with respect to other techniques such as finite differences, which would require a much larger number of costly objective function evaluations. The accuracy of the gradients obtained improves the convergence of gradient-based optimisation algorithms and provides an effective tool for optimal design in coupled environments.

9.3 Recommendations for Future Work

In this dissertation, a novel technique for the calculation of adjoint-based sensitivities in coupled Fluid-Structure Interaction problems has been proposed. The accuracy of the gradients obtained using this method was demonstrated, and applied them to canonical examples to provide proof-of-concept in optimal design. This new technique has served to answer the research questions that we had targeted in this dissertation; however, further work remains before the framework proposed in this work can be applied to industry-relevant design optimisation problems. As a result, and to conclude this thesis, an overview of future works is presented in this section which should make it possible to achieve this goal.

First, the **implementation of the time-domain coupled ADDA method for FSI problems** should be completed. It has been demonstrated in sections 6.1.3 and 7.4 that the AD-based adjoint method is able to compute sensitivities and perform optimisation in structural-only dynamic problems. Zhou *et al.* [123,124] show the applicability of the technique in time-domain aeroacoustic problems. Therefore, it should be feasible to apply ADDA to coupled, time-domain problems, the interest of which is growing as they provide response to more realistic dynamics with respect to idealised, time-independent problems. Support for dynamic FSI primal problems in SU2 has been shown in sections 2.3 and 5.2.2, and the equations for the adjoint were proposed in section 3.5.1. Therefore, only the incorporation of the time-dependent terms remains.

Next, in this work a fully non-linear structural solver has been implemented, with special care taken to avoid any physics-based simplification. However, some **improvements in the structural solver** remain to be done before it can be applied to complex, industrial problems. First, the implementation has been limited to linear shape functions at this time. This is mainly caused by the original finite-volume nature of SU2, which so far has lacked support for higher-order elements. This issue has reduced the accuracy of the solver for problems with large bending behaviour due to shear locking, as shown in section 5.1.2. However, there are ongoing efforts to **incorporate higher-order elements** at the University of Twente for the fluid solver in SU2, which are expected to be rapidly extensible for the structural solver.

Further improvements in the structural solver would be to **incorporate variational principles** such as the three-field Hu-Washizu formulation [197] via the mean dilatation technique [147], to avoid volumetric locking in incompressible materials as explained in section 4.1. Although this problem has not been yet addressed, the code has been designed to be able to accommodate these techniques. Moreover, it would be interesting to **explore new types of advanced elements** with solid interface, as for example solid-shell elements [214, 215], which could provide an advantage in the solution of three-dimensionally discretised domains with a predominantly planar behaviour, such as for example aircraft wings.

Large, three-dimensional problems have not been addressed in this dissertation, although the ADDA methodology does not require any specific modification in order to accommodate 3D problems. However, some limitations in SU2 remain at the time of writing, which has so far prevented the solution of these problems. First, **the mesh deformation techniques need to be improved** in order to effectively deal with very large fluid meshes. The methods currently available in SU2 for the solution of linear systems require an extraordinary (and unjustified) computational effort to move the mesh. New techniques could be incorporated without any modification to the adjoint solver, apart from an explicit transposition of the new linear solvers to avoid the computational effort of recording the solution of the system. Moreover, there is a need for a more versatile definition of boundary conditions in the mesh problem. For example, one may want the inlet boundary to be fixed over time, or maybe to undergo fixed displacements, the symmetric boundaries need efficient algorithms for in-plane only displacements, etc.

Furthermore, the problems presented in this thesis have been limited to matching mesh discretisations, although the transfer routines between fields were designed with interpolation in

mind. In general, the information transfer between solvers has to meet strict criteria regarding conservativeness of momentum and energy. Nonetheless, at the same time it has to aim at obtaining an efficient solution from the numerical point of view [83]. In this sense, both accuracy and robustness must be achieved in the transfer scheme, which at the same time has to be efficient (maximum accuracy with minimum cost) and simple to implement [216, 217]. Current efforts are being placed into **incorporating interpolation methods** such as Radial Basis Functions (see, e.g., [218]). A recent Master's thesis carried out in our group by Ho [219] successfully explores these methods in the context of SU2. The coupled adjoint method has been designed in a way that these new extensions would be immediately incorporated to the sensitivity analysis.

Another potential improvement that would benefit the solution of large 3D problems is an **efficient domain decomposition independent for each subproblem**. The current implementation forces the parallelisation of the coupled problems to use the same processors for flow and structural solver. That means, very large fluid domains are decomposed using the same number of subdomains as normally smaller structural domains, which limits scalability at the time of writing. An efficient distribution of MPI communicators, for which the basic structure is already available in the code, would likely solve this problem and improve the parallel behaviour of the coupled code.

Finally, and in order to carry out industry-relevant optimal design, the **incorporation of complex constraints to the optimisation** is required. In this thesis, the optimisation problems solved are only constrained by the physical equilibrium equations, being therefore the feasibility of the coupled problem the only constraint in the problem definition. These problems were rewritten in an unconstrained form using the Lagrangian and the adjoint variables, and only a design-variable constraint was imposed in section 8.2 via a penalty term in the objective function. In order to incorporate a more generalistic approach to stress-based or aerodynamic-based constraints, a new adjoint problem need to be solved to compute the sensitivities for each constraint in the problem [91]. Therefore, the adjoint would remain an efficient technique as long as the number of constraints remains below the number of design variables. With this purpose, it is suggested to follow approaches such as the Kreisselmeier–Steinhauser (KS) function for constraint aggregation [91, 98, 220], which accumulates the constraints into a single function for an efficient adjoint computation.

Bibliography

- [1] T. Belytschko. Methods and programs for analysis of fluid-structure systems. *Nuclear Engineering and Design*, 42(1):41 – 52, 1977.
- [2] T. Belytschko. Fluid-structure interaction. *Computers & Structures*, 12(4):459 – 469, 1980.
- [3] K.-J. Bathe and W.F. Hahn. On transient analysis of fluid-structure systems. *Computers and Structures*, 10(1-2):383–391, 1979.
- [4] J. Donea, S. Giuliani, and J.P. Halleux. An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 33(1-3):689–723, 1982.
- [5] Y. Bazilevs, K. Takizawa, and T.E. Tezduyar. *Computational Fluid-Structure Interaction: Methods and Applications*. John Wiley & Sons, Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO10 8SQ, UK, 2013.
- [6] R. T. Biedron and E. M. Lee-Rausche. Rotor airloads prediction using unstructured meshes and loose cfd/csd coupling. In *Collection of Technical Papers - AIAA Applied Aerodynamics Conference*, 2008.
- [7] S. C. Miller, M. P. Rumpfkeil, and J. J. Joo. Fluid-structure interaction of a variable camber compliant wing. In *53rd AIAA Aerospace Sciences Meeting*, 2015.
- [8] S. J. Massey, B. K. Stanford, C. D. Wieseman, and J. Heeg. Aeroelastic analysis of a distributed electric propulsion wing. In *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2017*, 2017.

- [9] R. Heinrich, N. Kroll., J. Neumann, and B. Nagel. Fluid-structure coupling for aerodynamic analysis and design - a dlr perspective. In *46th AIAA Aerospace Sciences Meeting and Exhibit*, 2008.
- [10] S. Keye. Fluid-structure coupled analysis of a transport aircraft and flight-test validation. *Journal of Aircraft*, 48(2):381–390, 2011.
- [11] J. Brezillon, A. Ronzheimer, D. Haar, M. Abu-Zurayk, M. Lummer, W. Krüger, and F. J. Natterer. Development and application of multi-disciplinary optimization capabilities based on high-fidelity methods. In *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 2012*, 2012.
- [12] L. Cambier and J. . Veuillot. Status of the elsa cfd software for flow simulation and multidisciplinary applications. In *46th AIAA Aerospace Sciences Meeting and Exhibit*, 2008.
- [13] M. Errera, A. Dugeai, Ph. Girodroux-Lavigne, J.-D. Garaud, M. Poinot, S. Cerqueira, and G. Chaineray. Multi-Physics Coupling Approaches for Aerospace Numerical Simulations. *Aerospace Lab, Issue 2*, 2011.
- [14] L. Cambier, S. Heib, and S. Plot. The onera elsa cfd software: Input from research and feedback from industry. *Mechanics and Industry*, 14(3):159–174, 2013.
- [15] M. Saeedi, K. Bletzinger, and R. Wüchner. Investigation of prestress-dependent aerodynamic performance of a double membrane sailwing. *Journal of Aircraft*, 54(3):980–994, 2017.
- [16] T. T. Nguyen, D. Shyam Sundar, K. S. Yeo, and T. T. Lim. Modeling and analysis of insect-like flexible wings at low reynolds number. *Journal of Fluids and Structures*, 62:294–317, 2016.
- [17] S. Tobing, J. Young, and J. C. S. Lai. Effects of wing flexibility on bumblebee propulsion. *Journal of Fluids and Structures*, 68:141–157, 2017.
- [18] W. Liu, Q. Xiao, and Q. Zhu. Passive flexibility effect on oscillating foil energy harvester. *AIAA Journal*, 54(4):1172–1187, 2016.

- [19] B. K. S. Woods, I. Dayyani, and M. I. Friswell. Fluid/structure-interaction analysis of the fish-bone-active-camber morphing concept. *Journal of Aircraft*, 52(1):307–319, 2015.
- [20] S. Buoso and R. Palacios. Electro-aeromechanical modelling of actuated membrane wings. *Journal of Fluids and Structures*, 58:188–202, 2015.
- [21] S. Buoso and R. Palacios. Viscoelastic effects in the aeromechanics of actuated elastomeric membrane wings. *Journal of Fluids and Structures*, 63:40 – 56, 2016.
- [22] F. Nieto, J. S. Owen, D. M. Hargreaves, and S. Hernández. Bridge deck flutter derivatives: Efficient numerical evaluation exploiting their interdependence. *Journal of Wind Engineering and Industrial Aerodynamics*, 136:138–150, 2015.
- [23] N. Lee, H. Lee, C. Baek, and S. Lee. Aeroelastic of bridge deck flutter with modified implicit coupling method. *Journal of Wind Engineering and Industrial Aerodynamics*, 155:11–22, 2016.
- [24] M. Andre, M. Péntek, K. . Bletzinger, and R. Wüchner. Aeroelastic simulation of the wind-excited torsional vibration of a parabolic trough solar collector. *Journal of Wind Engineering and Industrial Aerodynamics*, 165:67–78, 2017.
- [25] Y. Bazilevs, A. Korobenko, X. Deng, and J. Yan. Fluid-structure interaction modeling for fatigue-damage prediction in full-scale wind-turbine blades. *Journal of Applied Mechanics, Transactions ASME*, 83(6), 2016.
- [26] A. Michalski, B. Gawenat, P. Gellenne, and E. Haug. Computational wind engineering of large umbrella structures. *Journal of Wind Engineering and Industrial Aerodynamics*, 144:96–107, 2015.
- [27] V. Nguyen and B. Gatzhammer. A fluid structure interactions partitioned approach for simulations of explosive impacts on deformable structures. *International Journal of Impact Engineering*, 80:65–75, 2015.
- [28] M. Song, J. Ma, and Y. Huang. Fluid-structure interaction analysis of ship-ship collisions. *Marine Structures*, 55:121–136, 2017.
- [29] H. Casquero, C. Bona-Casas, and H. Gomez. Nurbs-based numerical proxies for red blood cells and circulating tumor cells in microscale blood flow. *Computer Methods in Applied Mechanics and Engineering*, 316:646–667, 2017.

- [30] F. Verdugo, C. J. Roth, L. Yoshihara, and W. A. Wall. Efficient solvers for coupled models in respiratory mechanics. *International Journal for Numerical Methods in Biomedical Engineering*, 33(2), 2017.
- [31] P. Tricerri, L. Dedè, S. Deparis, A. Quarteroni, A. M. Robertson, and A. Sequeira. Fluid-structure interaction simulations of cerebral arteries modeled by isotropic and anisotropic constitutive laws. *Computational Mechanics*, 55(3):479–498, 2015.
- [32] G. Luraghi, W. Wu, F. De Gaetano, J. F. Rodriguez Matas, G. D. Moggridge, M. Serrani, J. Stasiak, M. L. Costantino, and F. Migliavacca. Evaluation of an aortic valve prosthesis: Fluid-structure interaction or structural simulation? *Journal of Biomechanics*, 58:45–51, 2017.
- [33] A. Eken and M. Sahin. A parallel monolithic approach for fluid-structure interaction in a cerebral aneurysm. *Computers and Fluids*, 153:61–75, 2017.
- [34] M. C. H. Wu, D. Kamensky, C. Wang, A. J. Herrema, F. Xu, M. S. Pigazzini, A. Verma, A. L. Marsden, Y. Bazilevs, and M. . Hsu. Optimizing fluid–structure interaction systems with immersogeometric analysis and surrogate modeling: Application to a hydraulic arresting gear. *Computer Methods in Applied Mechanics and Engineering*, 316:668–693, 2017.
- [35] U. P. Farias Filho, A. R. E. Antunes, S. M. A. Bastos, and P. R. M. Lyra. Minimization of vortex induced vibrations using surrogate based optimization. *Structural and Multidisciplinary Optimization*, 52(4):717–735, 2015.
- [36] A.I.J. Forrester and A.J. Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1):50 – 79, 2009.
- [37] S. Koziel, D. Echeverría Ciaurri, and L. Leifsson. *Surrogate-Based Methods*, pages 33–59. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [38] T. P. Combes, A. S. Malik, G. Bramesfeld, and M. W. McQuilling. Efficient fluid-structure interaction method for conceptual design of flexible, fixed-wing micro-air-vehicle wings. *AIAA Journal*, 53(6):1442–1454, 2015.

- [39] M. Berci, V. V. Toropov, R. W. Hewson, and P. H. Gaskell. Multidisciplinary multifidelity optimisation of a flexible wing aerofoil with reference to a small uav. *Structural and Multidisciplinary Optimization*, 50(4):683–699, 2014.
- [40] D. W. Zingg, M. Nemec, and T. H. Pulliam. A comparative evaluation of genetic and gradient-based algorithms applied to aerodynamic optimization. *European Journal of Computational Mechanics*, 17(1-2):103–126, 2008.
- [41] J.R.R.A. Martins, P. Sturdza, and J.J. Alonso. The complex-step derivative approximation. *ACM Transactions on Mathematical Software*, 29(3):245–262, 2003.
- [42] K. Maute, M. Nikbay, and C. Farhat. Sensitivity analysis and design optimization of three-dimensional non-linear aeroelastic systems by the adjoint method. *International Journal for Numerical Methods in Engineering*, 56(6):911–933, 2003.
- [43] M. Barcelos and K. Maute. Aeroelastic design optimization for laminar and turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 197(19-20):1813–1832, 2008.
- [44] M. Zhu and M. H. Scott. Direct differentiation of the particle finite-element method for fluid-structure interaction. *Journal of Structural Engineering (United States)*, 142(3), 2016.
- [45] S. Nemat-Nasser and G. Herrmann. Adjoint systems in nonconservative problems of elastic stability. *AIAA Journal*, 4(12):2221–2222, 1966.
- [46] S.N. Prasad and G. Herrmann. The usefulness of adjoint systems in solving nonconservative stability problems of elastic continua. *International Journal of Solids and Structures*, 5(7):727 – 735, 1969.
- [47] J. Sobieszczanski-Sobieski. Sensitivity of complex, internally coupled systems. *AIAA journal*, 28(1):153–160, 1990.
- [48] K. Maute, M. Nikbay, and C. Farhat. Analytically based sensitivity analysis and optimization of nonlinear aeroelastic systems. In *8th Symposium on Multidisciplinary Analysis and Optimization*, Long Beach, CA, 6-8 September, 2000.

- [49] K. Maute, M. Nikbay, and C. Farhat. Coupled analytical sensitivity analysis and optimization of three-dimensional nonlinear aeroelastic systems. *AIAA Journal*, 39(11):2051–2061, 2001.
- [50] J.R.R.A. Martins, J.J. Alonso, and J.J. Reuther. Aero-structural wing design optimization using high-fidelity sensitivity analysis. In H. Honlinger, editor, *Proceedings - CEAS Conference on Multidisciplinary Aircraft Design Optimization*, pages 211–226, Cologne, Germany, 2001.
- [51] J.R.R.A. Martins, J.J. Alonso, and J.J. Reuther. Complete configuration aero-structural optimization using a coupled sensitivity analysis method. In *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization 2002*, Atlanta, GA, 4-6 September, 2002. American Institute of Aeronautics and Astronautics Inc.
- [52] M. Barcelos, H. Baviestrello, and K. Maute. A Schur-Newton-Krylov solver for steady-state aeroelastic analysis and design sensitivity analysis. *Computer Methods in Applied Mechanics and Engineering*, 195(17-18):2050–2069, 2006.
- [53] A. Fazzolari, N.R. Gauger, and J. Brezillon. Efficient aerodynamic shape optimization in MDO context. *Journal of Computational and Applied Mathematics*, 203(2 SPEC. ISS.):548–560, 2007.
- [54] M. Perego, A. Veneziani, and C. Vergara. A variational approach for estimating the compliance of the cardiovascular tissue: An inverse fluid-structure interaction problem. *SIAM Journal on Scientific Computing*, 33(3):1181–1211, 2011.
- [55] M. Abu-Zurayk and J. Brezillon. Development of the adjoint approach for aeroelastic wing optimization. *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, 121:59–66, 2013.
- [56] J. Degroote, M. Hojjat, E. Stavropoulou, R. Wüchner, and K.-U. Bletzinger. Partitioned solution of an unsteady adjoint for strongly coupled fluid-structure interactions and application to parameter identification of a one-dimensional problem. *Structural and Multidisciplinary Optimization*, 47(1):77–94, 2013.

- [57] G.K.W. Kenway, G.J. Kennedy, and J.R.R.A. Martins. Scalable parallel approach for high-fidelity steady-state aeroelastic analysis and adjoint derivative computations. *AIAA Journal*, 52(5):935–951, 2014.
- [58] M. Abu-Zurayk. *An Aeroelastic Coupled Adjoint Approach for Multi-Point Designs in Viscous Flows*. PhD thesis, RWTH Aachen University, 2016.
- [59] A. Mishra, D. Mavriplis, and J. Sitaraman. Time-dependent aeroelastic adjoint-based aerodynamic shape optimization of helicopter rotors in forward flight. *AIAA Journal*, 54(12):3813–3827, 2016.
- [60] F. Nobile, M. Pozzoli, and C. Vergara. Time accurate partitioned algorithms for the solution of fluid-structure interaction problems in haemodynamics. *Computers and Fluids*, 86:470–482, 2013.
- [61] C. Kassiotis, A. Ibrahimbegovic, R. Niekamp, and H.G. Matthies. Nonlinear fluid-structure interaction problem. part I: Implicit partitioned algorithm, nonlinear stability proof and validation examples. *Computational Mechanics*, 47(3):305–323, 2011.
- [62] M. Von Scheven and E. Ramm. Strong coupling schemes for interaction of thin-walled structures and incompressible flows. *International Journal for Numerical Methods in Engineering*, 87(1-5):214–231, 2011.
- [63] P. Le Tallec and J. Mouro. Fluid structure interaction with large structural displacements. *Computer Methods in Applied Mechanics and Engineering*, 190(24-25):3039–3067, 2001.
- [64] H. G. Matthies and J. Steindorf. Partitioned but strongly coupled iteration schemes for nonlinear fluid–structure interaction. *Computers & Structures*, 80(27–30):199–1999, 2002.
- [65] U. Küttler and W.A. Wall. Fixed-point fluid-structure interaction solvers with dynamic relaxation. *Computational Mechanics*, 43(1):61–72, 2008.
- [66] J. Degroote, K.-J. Bathe, and J. Vierendeels. Performance of a new partitioned procedure versus a monolithic procedure in fluid-structure interaction. *Computers and Structures*, 87(11-12):793–801, 2009.
- [67] C. Habchi, S. Russeil, D. Bougeard, J.-L. Harion, T. Lemenand, A. Ghanem, D.D. Valle, and H. Peerhossaini. Partitioned solver for strongly coupled fluid-structure interaction. *Computers and Fluids*, 71:306–319, 2013.

- [68] W. Dettmer and D. Perić. A computational framework for fluid-structure interaction: Finite element formulation and applications. *Computer Methods in Applied Mechanics and Engineering*, 195(41-43):5754–5779, 2006.
- [69] C. Farhat and M. Lesoinne. Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems. *Computer Methods in Applied Mechanics and Engineering*, 182(3-4):499–515, 2000.
- [70] C. Farhat, K.G. van der Zee, and P. Geuzaine. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 195(17-18):1973–2001, 2006.
- [71] S. Piperno and C. Farhat. Partitioned procedures for the transient solution of coupled aeroelastic problems – part II: energy transfer analysis and three-dimensional applications. *Computer Methods in Applied Mechanics and Engineering*, 190(24-25):3147–3170, 2001.
- [72] T. Richter. A monolithic geometric multigrid solver for fluid-structure interactions in ale formulation. *International Journal for Numerical Methods in Engineering*, 104(5):372–390, 2015.
- [73] U. Küttler, M. Gee, Ch. Förster, A. Comerford, and W.A. Wall. Coupling strategies for biomedical fluid-structure interaction problems. *International Journal for Numerical Methods in Biomedical Engineering*, 26(3-4):305–321, 2010.
- [74] M. Heil, A.L. Hazel, and J. Boyle. Solvers for large-displacement fluid-structure interaction problems: Segregated versus monolithic approaches. *Computational Mechanics*, 43(1):91–101, 2008.
- [75] P. Causin, J.F. Gerbeau, and F. Nobile. Added-mass effect in the design of partitioned algorithms for fluid-structure problems. *Computer Methods in Applied Mechanics and Engineering*, 194(42-44):4506–4527, 2005.
- [76] S. R. Idelsohn, F. Del Pin, R. Rossi, and E. Oñate. Fluid-structure interaction problems with strong added-mass effect. *International Journal for Numerical Methods in Engineering*, 80(10):1261–1294, 2009.

- [77] W.G. Dettmer and D. Perić. A new staggered scheme for fluid-structure interaction. *International Journal for Numerical Methods in Engineering*, 93(1):1–22, 2013.
- [78] M. A. Storti, N. M. Nigro, R. R. Paz, and L. D. Dalcín. Strong coupling strategy for fluid-structure interaction problems in supersonic regime via fixed point iteration. *Journal of Sound and Vibration*, 320(4-5):859–877, 2009.
- [79] M. M. Joosten, W. G. Dettmer, and D. Perić. Analysis of the block gauss-seidel solution procedure for a strongly coupled model problem with reference to fluid-structure interaction. *International Journal for Numerical Methods in Engineering*, 78(7):757–778, 2009.
- [80] J. Degroote, S. Annerel, and J. Vierendeels. Stability analysis of gauss-seidel iterations in a partitioned simulation of fluid-structure interaction. *Computers and Structures*, 88(5-6):263–271, 2010.
- [81] C. Farhat, M. Lesoinne, and N. Maman. Mixed explicit/implicit time integration of coupled aeroelastic problems: three-field formulation, geometric conservation and distributed solution. *International Journal for Numerical Methods in Fluids*, 21(10):807–835, 1995.
- [82] J. Donea, A. Huerta, J.-Ph. Ponthot, and A. Rodríguez-Ferran. *Encyclopedia of Computational Mechanics*, chapter Arbitrary Lagrangian-Eulerian Methods. John Wiley & Sons, Ltd, 2004.
- [83] C. Farhat, M. Lesoinne, and P. LeTallec. Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 157(1-2):95–114, 1998.
- [84] S. Deparis, M. Discacciati, G. Fourestey, and A. Quarteroni. Fluid-structure algorithms based on Steklov-Poincaré operators. *Computer Methods in Applied Mechanics and Engineering*, 195(41-43):5797–5812, 2006.
- [85] M.A. Fernández, J.-F. Gerbeau, and C. Grandmont. A projection semi-implicit scheme for the coupling of an elastic structure with an incompressible fluid. *International Journal for Numerical Methods in Engineering*, 69(4):794–821, 2007.

- [86] R. Löhner, E. Haug, A. Michalski, B. Muhammad, A. Drego, R. Nanjundaiah, and R. Zarfam. Recent advances in computational wind engineering and fluid-structure interaction. *Journal of Wind Engineering and Industrial Aerodynamics*, 144:14–23, 2015.
- [87] J. Murua, R. Palacios, and J.M.R. Graham. Assessment of wake-tail interference effects on the dynamics of flexible aircraft. *AIAA Journal*, 50(7):1575–1585, 2012.
- [88] H. Hesse, R. Palacios, and J. Murua. Consistent structural linearization in flexible aircraft dynamics with large rigid-body motion. *AIAA Journal*, 52(3):528–538, 2014.
- [89] R. J. S. Simpson and R. Palacios. Integrated flight dynamics and aeroelasticity of flexible aircraft with application to swept flying wings. In *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2015.
- [90] S. Hernández. *Métodos de diseño óptimo de estructuras*. CICCIP, 1990.
- [91] J.R.R.A. Martins, J.J. Alonso, and J.J. Reuther. A coupled-adjoint sensitivity analysis method for high-fidelity aero-structural design. *Optimization and Engineering*, 6(1):33–62, 2005.
- [92] R. P. Liem, C. A. Mader, and J. R. R. A. Martins. Surrogate models and mixtures of experts in aerodynamic performance prediction for aircraft mission analysis. *Aerospace Science and Technology*, 43:126–151, 2015.
- [93] S. Buoso. *High-fidelity modelling and feedback control of bio-inspired membrane wings*. PhD thesis, Imperial College London, 2016.
- [94] S. Khosravi and D. W. Zingg. Aerostructural perspective on winglets. *Journal of Aircraft*, 54(3):1121–1138, 2017.
- [95] Z. J. Zhang, S. Khosravi, and D. W. Zingg. High-fidelity aerostructural optimization with integrated geometry parameterization and mesh movement. *Structural and Multidisciplinary Optimization*, 55(4):1217–1235, 2017.
- [96] M. Hojjat, E. Stavropoulou, T. Gallinger, U. Israel, R. Wüchner, and Bletzinger K.-U. *Fluid Structure Interaction II*, volume 73, chapter Fluid-Structure Interaction in the Context of Shape Optimization and Computational Wind Engineering, pages 351–381. Springer Berlin Heidelberg, 2010.

- [97] J.R.R.A. Martins and J.T. Hwang. Review and unification of methods for computing derivatives of multidisciplinary computational models. *AIAA Journal*, 51(11):2582–2599, 2013.
- [98] N. Garg, G. K. W. Kenway, J. R. R. A. Martins, and Y. L. Young. High-fidelity multipoint hydrostructural optimization of a 3-d hydrofoil. *Journal of Fluids and Structures*, 71:15–39, 2017.
- [99] S. Kreissl, G. Pingen, A. Evgrafov, and K. Maute. Topology optimization of flexible micro-fluidic devices. *Structural and Multidisciplinary Optimization*, 42(4):495–516, 2010.
- [100] C. S. Andreasen and O. Sigmund. Topology optimization of fluid-structure-interaction problems in poroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 258:55–62, 2013.
- [101] R. Picelli, W. M. Vicente, and R. Pavanello. Evolutionary topology optimization for structural compliance minimization considering design-dependent fsi loads. *Finite Elements in Analysis and Design*, 135:44–55, 2017.
- [102] N. Jenkins and K. Maute. An immersed boundary approach for shape and topology optimization of stationary fluid-structure interaction problems. *Structural and Multidisciplinary Optimization*, 54(5):1191–1208, 2016.
- [103] M. Raulli and K. Maute. Optimization of fully coupled electrostatic–fluid–structure interaction problems. *Computers & Structures*, 83(2–3):221 – 233, 2005.
- [104] P.W. Fick, E.H. van Brummelen, and K.G. van der Zee. On the adjoint-consistent formulation of interface conditions in goal-oriented error estimation and adaptivity for fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 199(49–52):3369–3385, 2010.
- [105] K.G. van der Zee, E.H. van Brummelen, I. Akkerman, and R. de Borst. Goal-oriented error estimation and adaptivity for fluid-structure interaction using exact linearized adjoints. *Computer Methods in Applied Mechanics and Engineering*, 200(37–40):2738–2757, 2011.
- [106] Y. Bazilevs, M.-C. Hsu, and M.T. Bement. Adjoint-based control of fluid-structure interaction for computational steering applications. In *13th Annual International Conference on Computational Science, ICCS*, volume 18, pages 1989–1998, Barcelona, 2013. Elsevier.

- [107] H. Chung and J. J. Alonso. Using gradients to construct cokriging approximation models for high-dimensional design optimization problems. In *40th AIAA Aerospace Sciences Meeting and Exhibit*, 2002.
- [108] J. Laurenceau, M. Meaux, M. Montagnac, and P. Sagaut. Comparison of gradient-based and gradient-enhanced response-surface-based optimizers. *AIAA Journal*, 48(5):981–994, 2010.
- [109] Z. Han, S. Görtz, and R. Zimmermann. Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function. *Aerospace Science and Technology*, 25(1):177–189, 2013.
- [110] D.G. Krige. A statistical approach to some basic mine valuation problems on the witwatersrand. *Journal of the Southern African Institute of Mining and Metallurgy*, 52(6):119–139, 1951.
- [111] J. Laurenceau and P. Sagaut. Building efficient response surfaces of aerodynamic functions with kriging and cokriging. *AIAA Journal*, 46(2):498–507, 2008.
- [112] J. Ollar, C. Mortished, R. Jones, J. Sienz, and V. Toropov. Gradient based hyperparameter optimisation for well conditioned kriging metamodels. *Structural and Multidisciplinary Optimization*, 55(6):2029–2044, 2017.
- [113] J.R.R.A. Martins, J.J. Alonso, and J.J. Reuther. High-fidelity aerostructural design optimization of a supersonic business jet. *Journal of Aircraft*, 41(3):523–530, 2004.
- [114] A. Griewank and G.F Corliss, editors. *Automatic Differentiation of Algorithms: Theory, Implementation, and Application*, volume Proceedings of the first SIAM Workshop on Automatic Differentiation, Breckenridge, CO, January 6-8, 1991. SIAM - Society for Industrial and Applied Mathematics.
- [115] A. Griewank and A. Walther. *Evaluating Derivatives*. Society for Industrial and Applied Mathematics, second edition, 2008.
- [116] A.C. Marta, C.A. Mader, J.R.R.A. Martins, E. Van Der Weide, and J.J. Alonso. A methodology for the development of discrete adjoint solvers using automatic differentiation tools. *International Journal of Computational Fluid Dynamics*, 21(9-10):307–327, 2007.

- [117] C.A. Mader, J.R.R.A. Martins, J.J. Alonso, and E.D. Van Weide. Adjoint: An approach for the rapid development of discrete adjoint solvers. *AIAA Journal*, 46(4):863–873, 2008.
- [118] C.A. Mader and J.R.R.A. Martins. Derivatives for time-spectral computational fluid dynamics using an automatic differentiation adjoint. *AIAA Journal*, 50(12):2809–2819, 2012.
- [119] I. Ozaki, F. Kimura, and M. Berz. Higher-order sensitivity analysis of finite element method by automatic differentiation. *Computational Mechanics*, 16(4):223–234, 1995.
- [120] T. Albring, M. Sagebaum, and N.R. Gauger. Development of a consistent discrete adjoint solver in an evolving aerodynamic design framework. In *16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 22-26 June*, Dallas, TX, 2015.
- [121] T. Albring, M. Sagebaum, and N.R. Gauger. Efficient aerodynamic design using the discrete adjoint method in su2. In *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2016.
- [122] R.J. Hogan. Fast reverse-mode automatic differentiation using expression templates in C++. *ACM Transactions on Mathematical Software*, 40(4), 2014.
- [123] B.Y. Zhou, T. Albring, N.R. Gauger, T. D. Economon, F. Palacios, and J. J. Alonso. A discrete adjoint framework for unsteady aerodynamic and aeroacoustic optimization. In *16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 22-26 June*, Dallas, TX, 2015.
- [124] B. Y. Zhou, T. Albring, N. R. Gauger, C. Ilario, T. Economon, and J. J. Alonso. Reduction of airframe noise components using a discrete adjoint approach. *AIAA 2017-3658*, 2017.
- [125] F. Palacios, M.R. Colonno, A.C. Aranake, A. Campos, S.R. Copeland, T.D. Economon, A.K. Lonkar, T.W. Lukaczyk, T.W.R. Taylor, and J.J. Alonso. Stanford University Unstructured (SU2): An open-source integrated computational environment for multi-physics simulation and design. In *AIAA 51st Aerospace Sciences Meeting, 7-10 January*, Grapevine, TX, 2013.
- [126] F. Palacios, T.D. Economon, A.C. Aranake, S.R. Copeland, A.K. Lonkar, T.W. Lukaczyk, D.E. Manosalvas, K.R. Naik, A. Santiago Padrón, B. Tracey, A. Variyar, and J.J. Alonso.

- Stanford university unstructured (SU2): Open-source analysis and design technology for turbulent flows. In *AIAA 52nd Aerospace Sciences Meeting, SciTech, 13-17 January*, National Harbor, MD, 2014.
- [127] T.D. Economon, F. Palacios, S.R. Copeland, T.W. Lukaczyk, and J.J. Alonso. SU2: An open-source suite for multiphysics simulation and design. *AIAA Journal*, 54(3):828–846, 2016.
- [128] T.D. Economon, F. Palacios, and J.J. Alonso. Optimal Shape Design for Open Rotor Blades. In *30th AIAA Applied Aerodynamics Conference, 25 - 28 June*, New Orleans, Louisiana, 2012.
- [129] F. Palacios, T. D. Economon, A. D. Wendorff, and J. J. Alonso. Large-scale aircraft design using su2. In *53rd AIAA Aerospace Sciences Meeting*, 2015.
- [130] T.D. Economon, F. Palacios, and J.J. Alonso. Unsteady continuous adjoint approach for aerodynamic design on dynamic meshes. *AIAA Journal*, 53(9):2437–2453, 2015.
- [131] T. D. Economon, D. Mudigere, G. Bansal, A. Heinecke, F. Palacios, J. Park, M. Smelyanskiy, J. J. Alonso, and P. Dubey. Performance optimizations for scalable implicit rans calculations with su2. *Computers and Fluids*, 129:146–158, 2016.
- [132] S. Vitale, M. Pini, P. Colonna, G. Gori, A.A. Guardone, T.D. Economon, F. Palacios, and J.J. Alonso. Extension of the su2 open source cfd code to the simulation of turbulent flows of fluids modelled with complex thermophysical laws. In *22nd AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics Inc, AIAA, 2015.
- [133] M. Pini, S. Vitale, P. Colonna, G. Gori, A. Guardone, T. Economon, J.J. Alonso, and F. Palacios. Su2: the open-source software for non-ideal compressible flows. *Journal of Physics: Conference Series*, 821(1):012013, 2017.
- [134] G. Gori, D. Vimercati, and A. Guardone. Non-ideal compressible-fluid effects in oblique shock waves. *Journal of Physics: Conference Series*, 821(1):012003, 2017.
- [135] E. S. Molina, C. Spode, R. G. A. Da Silva, D. E. Manosalvas-Kjono, S. Nimmagadda, T. D. Economon, J. J. Alonso, and M. Righi. Hybrid rans/les calculations in su2. In *23rd AIAA Computational Fluid Dynamics Conference, 2017*, 2017.

- [136] S. Nimmagadda, T. D. Economon, J. J. Alonso, and C. R. I. da Silva. Robust uniform time sampling approach for the harmonic balance method. In *46th AIAA Fluid Dynamics Conference*, 2016.
- [137] H. L. Kline, T. D. Economon, and J. J. Alonso. Multi-objective optimization of a hypersonic inlet using generalized outflow boundary conditions in the continuous adjoint method. In *54th AIAA Aerospace Sciences Meeting*, 2016.
- [138] A. Variyar, T. D. Economon, and J. J. Alonso. Multifidelity conceptual design and optimization of strut-braced wing aircraft using physics-based methods. In *54th AIAA Aerospace Sciences Meeting*, 2016.
- [139] B. C. Munguía, T. D. Economon, and J. J. Alonso. A discrete adjoint framework for low-boom supersonic aircraft shape optimization. In *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2017*, 2017.
- [140] R. Sanchez, R. Palacios, T.D. Economon, H.L. Kline, J.J. Alonso, and F. Palacios. Towards a Fluid-Structure Interaction solver for problems with large deformations within the open-source SU2 suite. In *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA SciTech*,, San Diego, CA, 4-8 Jan, 2016.
- [141] R. Sanchez, T. Albring, R. Palacios, Gauger N.R., T.D. Economon, and J.J. Alonso. Coupled Adjoint-Based Sensitivities in Large-Displacement Fluid-Structure Interaction using Algorithmic Differentiation. *International Journal for Numerical Methods in Engineering*, 2017.
- [142] R. Sanchez, R. Palacios, T.D. Economon, J.J. Alonso, T. Albring, and Gauger N.R. Optimal Actuation of Dielectric Membrane Wings using High-Fidelity Fluid-Structure Modelling. In *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA SciTech*,, Grapevine, TX. Jan 9-13, 2017.
- [143] R. Sanchez, H. Kline, D. Thomas, A. Variyar, M. Righi, T.D. Economon, J.J. Alonso, R. Palacios, G. Dimitriadis, and V. Terrapon. Assessment of the Fluid-Structure Interaction Capabilities for Aeronautical Applications of the Open-Source Solver SU2. In M. Papadrakakis, V. Papadopoulos, G. Stefanou, and V. Plevris, editors, *VII European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2016)*, Crete Island, Greece, 5-10 June, 2016.

- [144] R. Sanchez and R. Palacios. Computing Derivatives in Nonlinear Aeroelasticity using Algorithm Differentiation. In *International Forum on Aeroelasticity and Structural Dynamics (IFASD)*, Como, Italy, 25-28 June 2017.
- [145] A. J. Chorin. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics*, 135(2):118–125, 1997.
- [146] A. Jameson. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. In *AIAA 10th Computational Fluid Dynamics Conference, June 24-26*, Honolulu, HI, 1991.
- [147] J. Bonet and R. D. Wood. *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press, 2008.
- [148] K.-J. Bathe. *Finite Element Procedures in Engineering Analysis*. Civil engineering and engineering mechanics series. Prentice-Hall, 1982.
- [149] N. M. Newmark. A method of computation for structural dynamics. *Journal of the Engineering Mechanics Division*, 85(3):67–94, 1959.
- [150] M. Géradin and D. Rixen. *Mechanical Vibrations: Theory and Application to Structural Dynamics*. John Wiley & Sons Ltd, Chichester, UK., 2nd edition, 1997.
- [151] C. Farhat, M. Lesoinne, P. Stern, and S. Lantéri. High performance solution of three-dimensional nonlinear aeroelastic problems via parallel partitioned algorithms: Methodology and preliminary results. *Advances in Engineering Software*, 28(1):43–61, 1997.
- [152] R. Jaiman, P. Geubelle, E. Loth, and X. Jiao. Combined interface boundary condition method for unsteady fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 200(1-4):27–39, 2011.
- [153] H. G. Matthies and J. Steindorf. Partitioned strong coupling algorithms for fluid–structure interaction. *Computers & Structures*, 81(8–11):805–812, 2003.
- [154] J.. Vierendeels, L.. Lanoye, J. Degroote, and P. Verdonck. Implicit coupling of partitioned fluid-structure interaction problems with reduced order models. *Computers and Structures*, 85(11-14):970–976, 2007.

- [155] B. M. Irons and R. C. Tuck. A version of the Aitken accelerator for computer iteration. *International Journal for Numerical Methods in Engineering*, 1:275–277, 1969.
- [156] V. M. Korivi, A.C. Taylor, P.A. Newman, G.W. How, and H.E Jones. An incremental strategy for calculating consistent discrete CFD sensitivity derivatives. Technical report, National Aeronautics and Space Administration, Langley Research Center, 1992.
- [157] J.S. Arora. *Introduction to Design Optimization*. Academic Press, Boston, 3rd edition, 2012.
- [158] H.M. Adelman and R.T. Haftka. Sensitivity Analysis of Discrete Structural Systems. *AIAA journal*, 24(5):823–832, 1986.
- [159] M. Gunzburger. *Perspectives in Flow Control and Optimization*. Society for Industrial and Applied Mathematics, 2002.
- [160] L. Hascoet and V. Pascual. The tapenade automatic differentiation tool: Principles, model, and specification. *ACM Transactions on Mathematical Software*, 39(3), 2013.
- [161] T. Veldhuizen. Expression templates. *C++ Report*, 7(5):26–31, 1995.
- [162] C. Pflaum. Expression templates for partial differential equations. *Computing and Visualization in Science*, 4(1):1–8, 2001.
- [163] D.A. Di Pietro and A. Veneziani. Expression templates implementation of continuous and discontinuous galerkin methods. *Computing and Visualization in Science*, 12(8):421–436, 2009.
- [164] O. Pironneau. On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, 64(1):97–110, 1974.
- [165] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3(3):233–260, 1988.
- [166] J. Reuther, J. J. Alonso, M. J. Rimlinger, and A. Jameson. Aerodynamic shape optimization of supersonic aircraft configurations via an adjoint formulation on distributed memory parallel computers. *Computers and Fluids*, 28(4-5):675–700, 1999.

- [167] D. J. Mavriplis. Discrete adjoint-based approach for optimization problems on three-dimensional unstructured meshes. *AIAA Journal*, 45(4):740–750, 2007.
- [168] J. Brezillon and N.R. Gauger. 2d and 3d aerodynamic shape optimisation using the adjoint approach. *Aerospace Science and Technology*, 8(8):715 – 727, 2004.
- [169] S. Xu and S. Timme. Robust and efficient adjoint solver for complex flow conditions. *Computers and Fluids*, 148:26–38, 2017.
- [170] D. X. Wang and L. He. Adjoint aerodynamic design optimization for blades in multistage turbomachines-part i: Methodology and verification. *Journal of Turbomachinery*, 132(2), 2010.
- [171] M. Pini, G. Persico, D. Pasquale, and S. Rebay. Adjoint method for shape optimization in real-gas flow applications. *Journal of Engineering for Gas Turbines and Power*, 137(3), 2015.
- [172] M.B. Giles and N.A. Pierce. An introduction to the adjoint approach to design. *Flow, Turbulence and Combustion*, 65(3-4):393–415, 2000.
- [173] C. S. Kim, C. Kim, and O. H. Rho. Feasibility study of constant eddy-viscosity assumption in gradient-based design optimization. *Journal of Aircraft*, 40(6):1168–1176, 2003.
- [174] R.P. Dwight and J. Brezillon. Effect of approximations of the discrete adjoint on gradient-based optimization. *AIAA Journal*, 44(12):3022–3031, 2006.
- [175] M.P. Bendsøe. Optimal shape design as a material distribution problem. *Structural Optimization*, 1(4):193–202, 1989.
- [176] G.I.N. Rozvany. A critical review of established methods of structural topology optimization. *Structural and Multidisciplinary Optimization*, 37(3):217–237, 2009.
- [177] R.J. Yang and C.H. Chuang. Optimal topology design using linear programming. *Computers & Structures*, 52(2):265 – 275, 1994.
- [178] T. Buhl, C.B.W. Pedersen, and O. Sigmund. Stiffness design of geometrically nonlinear structures using topology optimization. *Structural and Multidisciplinary Optimization*, 19(2):93–104, 2000.

- [179] D. Jung and H.C. Gea. Topology optimization of nonlinear structures. *Finite Elements in Analysis and Design*, 40(11):1417–1427, 2004.
- [180] L. Li, G. Zhang, and K. Khandelwal. Topology optimization of energy absorbing structures with maximum damage constraint. *International Journal for Numerical Methods in Engineering*, 2017.
- [181] L. Xia, F. Fritzen, and P. Breitkopf. Evolutionary topology optimization of elastoplastic structures. *Structural and Multidisciplinary Optimization*, 55(2):569–581, Feb 2017.
- [182] C. Suchocki. A finite element implementation of knowles stored-energy function: Theory, coding and applications. *Archive of Mechanical Engineering*, 58(3):319–346, 2011.
- [183] C.C. Hsieh and J.S. Arora. Design sensitivity analysis and optimization of dynamic response. *Computer Methods in Applied Mechanics and Engineering*, 43(2):195 – 219, 1984.
- [184] J.S Jensen and B.S. Lazarov. Optimization of non-linear mass damper parameters for transient response. In B.R. Andrievsky and A.L. Fradkov, editors, *Proceedings of the Sixth EUROMECH Nonlinear Dynamics Conference*, 2008.
- [185] I. Kucuk, I.S. Sadek, E. Zeini, and S. Adali. Optimal vibration control of piezolaminated smart beams by the maximum principle. *Computers and Structures*, 89(9-10):744–749, 2011.
- [186] P.B. Nakshatrala and D.A. Tortorelli. Nonlinear structural design using multiscale topology optimization. part ii: Transient formulation. *Computer Methods in Applied Mechanics and Engineering*, 304:605–618, 2016.
- [187] D. A. Tortorelli, R. B. Habert, and S. C. . Lu. Adjoint sensitivity analysis for nonlinear dynamic thermoelastic systems. *AIAA Journal*, 29(2):253–263, 1991.
- [188] B. Stanford, P. Beran, and M. Kurdi. Adjoint sensitivities of time-periodic nonlinear structural dynamics via model reduction. *Computers and Structures*, 88(19-20):1110–1123, 2010.
- [189] K. Maute and M. Allen. Conceptual design of aeroelastic structures by topology optimization. *Structural and Multidisciplinary Optimization*, 27(1-2):27–42, 2004.

- [190] Z. Suo. Theory of dielectric elastomers. *Acta Mechanica Solida Sinica*, 23(6):549 – 578, 2010.
- [191] X. Zhao and Q. Wang. Harnessing large deformation and instabilities of soft dielectrics: Theory, experiment, and application. *Applied Physics Reviews*, 1(2), 2014.
- [192] R. Pelrine, R. Kornbluh, Q. Pei, and J. Joseph. High-speed electrically actuated elastomers with strain greater than 100%. *Science*, 287(5454):836–839, 2000.
- [193] H.S. Park, Q. Wang, X. Zhao, and P.A. Klein. Electromechanical instability on dielectric polymer surface: Modeling and experiment. *Computer Methods in Applied Mechanics and Engineering*, 260:40 – 49, 2013.
- [194] A. Ask, A. Menzel, and M. Ristinmaa. Electrostriction in electro-viscoelastic polymers. *Mechanics of Materials*, 50:9–21, 2012.
- [195] N. Goulbourne, E. Mockensturm, and M. Frecker. A nonlinear model for dielectric elastomer membranes. *Journal of Applied Mechanics, Transactions ASME*, 72(6):899–906, 2005.
- [196] X. Zhao and Z. Suo. Method to analyze programmable deformation of dielectric elastomer layers. *Applied Physics Letters*, 93(25), 2008.
- [197] J.C. Simo, R.L. Taylor, and K.S. Pister. Variational and projection methods for the volume constraint in finite deformation elasto-plasticity. *Computer Methods in Applied Mechanics and Engineering*, 51(1–3):177 – 208, 1985.
- [198] X. Zhao, W. Hong, and Z. Suo. Electromechanical hysteresis and coexistent states in dielectric elastomers. *Physical Review B - Condensed Matter and Materials Physics*, 76(13), 2007.
- [199] P. Steinmann, M. Hossain, and G. Possart. Hyperelastic models for rubber-like materials: Consistent tangent operators and suitability for Treloar’s data. *Archive of Applied Mechanics*, 82(9):1183–1217, 2012.
- [200] A.N. Gent. A new constitutive relation for rubber. *Rubber Chemistry and Technology*, 69(1):59–61, 1996.

- [201] E.M. Arruda and M.C. Boyce. A three-dimensional constitutive model for the large stretch behavior of rubber elastic materials. *Journal of the Mechanics and Physics of Solids*, 41(2):389–412, 1993.
- [202] Dassault System. SIMULIA ABAQUS FEA 6.13. *User Documentation*, 2013.
- [203] R.E. Gordnier. High fidelity computational simulation of a membrane wing airfoil. *Journal of Fluids and Structures*, 25(5):897–917, 2009.
- [204] W.A. Wall and E. Ramm. Fluid-Structure interaction based upon a stabilized (ALE) finite element method. In E. Oñate S.R. Idelsohn and E.N. Dvorkin (Eds.), editors, *Computational Mechanics. New Trends and Applications*. CIMNE, Barcelona, Spain, 1998.
- [205] C. Wood, A.J. Gil, O. Hassan, and J. Bonet. A partitioned coupling approach for dynamic fluid-structure interaction with applications to biological membranes. *International Journal for Numerical Methods in Fluids*, 57(5):555–581, 2008.
- [206] B. Froehle and P.-O. Persson. A high-order discontinuous Galerkin method for fluid-structure interaction with efficient implicit-explicit time stepping. *Journal of Computational Physics*, 272:97–104, 2014.
- [207] P. Rojratsirikul, Z. Wang, and I. Gursul. Unsteady fluid-structure interactions of membrane airfoils at low reynolds numbers. *Experiments in Fluids*, 46(5):859–872, 2009.
- [208] C. Geuzaine and J.-F. Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.
- [209] E. Simiu and R. H. Scanlan. *Wind effects on Structures: Fundamentals and Applications to Design*. John Wiley & Sons, 3rd edition, 1996.
- [210] J.Á. Jurado, R. Sanchez, and S. Hernández. Wind tunnel sectional tests for the identification of flutter derivatives and vortex shedding in long span bridges. In *7th Subrata Chakrabarti International Conference on Fluid Structure Interaction*., volume 129, pages 51–59, Gran Canaria, Spain, 10-12 April, 2013.
- [211] R. Sanchez, F. Nieto, K.C.S Kwok, and S. Hernandez. CFD analysis of the aerodynamic response of a twin-box deck considering different gap widths. In Apmtac, editor, *Congresso de Métodos Numéricos em Engenharia 2015*, Lisbon, Portugal, June-July 2015.

- [212] B. Hübner, E. Walhorn, and D. Dinkler. A monolithic approach to fluid-structure interaction using space-time finite elements. *Computer Methods in Applied Mechanics and Engineering*, 193(23-26):2087–2104, 2004.
- [213] E. Jones, T. Oliphant, P. Peterson, et al. Scipy: Open source scientific tools for python (*Online*), 2001–.
- [214] M. Schwarze and S. Reese. A reduced integration solid-shell finite element based on the EAS and the ANS concept - Geometrically linear problems. *International Journal for Numerical Methods in Engineering*, 80(10):1322–1355, 2009.
- [215] M. Schwarze and S. Reese. A reduced integration solid-shell finite element based on the EAS and the ANS concept - Large deformation problems. *International Journal for Numerical Methods in Engineering*, 85(3):289–329, 2011.
- [216] M.J. Smith, D.H. Hodges, and C.E.S. Cesnik. Evaluation of computational algorithms suitable for fluid-structure interactions. *Journal of Aircraft*, 37(2):282–294, 2000.
- [217] A. de Boer, A.H. van Zuijlen, and H. Bijl. Review of coupling methods for non-matching meshes. *Computer Methods in Applied Mechanics and Engineering*, 196(8):1515–1525, 2007.
- [218] A. Beckert and H. Wendland. Multivariate interpolation for fluid-structure-interaction problems using radial basis functions. *Aerospace Science and Technology*, 5(2):125–134, 2001.
- [219] J. Ho Mun Onn. Spatial Interpolation in Fluid-Structure Interaction with Radial Basis Functions. Master’s thesis, Imperial College London, 2017.
- [220] N. M. K. Poon and J. R. R. A. Martins. An adaptive approach to constraint aggregation using adjoint sensitivity analysis. *Structural and Multidisciplinary Optimization*, 34(1):61–73, 2007.