BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF MECHANICAL ENGINEERING
FAKULTA STROJNÍHO INŽENÝRSTVÍ

INSTITUTE OF AEROSPACE ENGINEERING
LETECKÝ ÚSTAV

NEW APPROACHES IN NUMERICAL AEROELASTICITY APPLIED IN AERODYNAMIC OPTIMIZATION OF ELASTIC WING

NEW APPROACHES IN NUMERICAL AEROELASTICITY APPLIED IN AERODYNAMIC OPTIMIZATION OF ELASTIC WING

DOCTORAL THESIS
DIZERTAČNÍ PRÁCE

AUTHOR Ing. Jan Navrátil
AUTOR PRÁCE

SUPERVISOR doc. Ing. Jiří Hlinka, Ph.D.
ŠKOLITEL

BRNO 2016

Abstract

The aeroelasticity is an essential discipline involved in the aircraft design, aiming to predict phenomena occurring due to interaction of aerodynamic, elastic and inertial forces. Those phenomena might often lead to catastrophic consequences, thus it must be proven that they do not occur between the speeds bounding the airplane flight envelope.

Current aircraft design leads to increased flexibility of the airframe as a result of modern materials application or aerodynamically efficient slender wings. The airframe flexibility influences the aerodynamic performance and it might significantly impact the aeroelastic effects, which can be more easily excited by rigid body motions than in case of stiffer structures. The potential aeroelastic phenomena can occur in large range of speeds involving transonic regime, where the non-linear flow effects significantly influence the flutter speed. Common aeroelastic analysis tools are mostly based on the linear theories for aerodynamic predictions, thus they fails to predict mentioned non-linear effect.

The objective of the thesis is, therefore, to design, implement and test an aeroelastic computational tool employing the aerodynamic prediction solver which is able to predict non-linear flow. In the thesis, the main focus is directed to the static aeroelastic simulations.

The methods involved in numerical static aeroelastic simulation are presented in the thesis. The implementation of the computational aeroelastic tool was described and the convergence of the coupled solver was investigated. The tool functionality was validated in the test cases involving different types of the aerodynamic and structural models. The tool was applied also in the aerodynamic shape optimization of an elastic wing. The results and computational cost were compared to the rigid wing optimization.

Last chapter presents the author's contribution to the research oriented on the assessment of time synchronization scheme for the CFD-CSM coupled problem. The test case used here is a transonic flow around the Benchmark Super-Critical Wing at flutter condition. Results were compared to the experimental data provided by NASA.

Keywords

aeroelasticity, computational, static, equivalent structure, aerodynamics, optimization, elastic, wing, flutter, transonic

Abstrakt

Aeroelasticita je nezbytná vědní disciplína zahrnuta do návrhu letounů. Zaměřuje se na předpovídání jevů, které vznikají vlivem interakce aerodynamických, elastických a setrvačných sil. Tyto jevy často vedou ke katastrofickým následkům, proto musí být prokázáno, že nevzniknou v rozsahu rychlostí ohraničujících letovou obálku.

Aplikace moderních materiálů při konstrukci draku, spolu se snahou navrhnout aerodynamicky efektivní tvar křídel, vede ke zvyšování poddajnosti letounů. To má za následek změnu aerodynamických vlastností a také k výraznějšímu vliv na aeroelastické jevy, které mohou být vyvolány snadněji vlivem pohybů tuhého tělesa než v případě tužších konstrukcí. Aeroelastické jevy mohou vznikat v širokém rozsahu rychlostí zahrnujícím i transsonickou oblast. V této oblasti je ovlivněna zejména rychlost, při níž dochází k třepetání, a to vlivem nelineárních jevů v proudu. Běžné nástroje, které jsou založeny na lineárních teoriích, nejsou schopny tyto nelineární jevy popsat.

Cílem práce je proto navrhnout, implementovat a otestovat nástroj pro výpočetní (numerickou) simulaci aeroelasticity. Nástroj má využívat řešič proudového pole, který je schopen předpovědět nelineární jevy. V práci je kladen důraz na simulaci statické aeroelasticity.

V práci jsou popsány metody, které je nutno zahrnout do numerické simulace statické aeroelasticity. Dále je popsán vlastní nástroj a je provedeno zhodnocení konvergence statických aeroelastických výpočtů. Funkčnost nástroje byla ověřena na příkladech, kdy byly použity různé aerodynamické a strukturální modely. Nástroj byl také aplikován při aerodynamické tvarové optimalizaci poddajného křídla. Výsledky optimalizace a její výpočetní náročnost byly porovnány s případem optimalizace tuhého křídla.

Na závěr je v práci prezentován příspěvek autora do výzkumu zaměřeného na zhodnocení vlivu časové synchronizace mezi CFD a CSM řešiči. Použitý testovací případ je transsonické obtékání křídla na začátku třepetání (flutteru). Výsledky byly srovnány s experimentálními daty poskytnutými NASA.

Klíčová slova

aeroelasticita, výpočetní, statický, podobná konstrukce, aerodynamika, optimalizace, elastický, křídlo, třepetní, flutter, transonický

NAVRÁTIL, J. *New Approaches in Numerical Aeroelasticity Applied in Aerodynamic Optimization of Elastic Wing.* Brno: Brno University of Technology, Faculty of Mechanical Engineering, 2016. 93 p. Supervised by doc. Ing. Jiří Hlinka, Ph.D.

Declaration

I declare that the presented thesis is the result of my own work under the guidance of my supervisor and specialist supervisors and I cited all literature and electronic sources what I used during the research.

. . . .

. . . .
Ing. Jan Navrátil

Acknowledgement

I would like to express my gratitude to all who supported me during writing this thesis. First, I would like to thank my supervisor doc. Ing. Jiří Hlinka, Ph.D. for the opportunity of doctoral study. I would like to thank to the specialist supervisors Ing. Robert Popela, Ph.D. and Olivier Amoignon, Ph.D. for their guidance and valuable advices. My special thanks belongs to Adam Jirásek, Ph.D. for his cooperation and advices in research on flutter simulation. Last but not least, I would like to thank my family for their support.

Contents

1 Introduction **1**

1.1 Overview . . . 1

1.2 The Objective of the Thesis . . . 2

1.3 The Novelty of the Thesis . . . 2

2 Static Aeroelastic Computations **3**

2.1 Introduction . . . 3

2.2 Fluid-Structure Interaction . . . 3

2.2.1 Principle of coupling . . . 3

2.2.2 Transformation methods . . . 4

2.2.3 Chosen coupling method . . . 7

2.3 Aerodynamics Modeling . . . 8

2.3.1 Governing equations . . . 8

2.3.2 Flow solver . . . 9

2.4 Structural Modeling . . . 9

2.4.1 Matlab based finite element solver . . . 9

2.5 Mesh Deformation . . . 14

2.5.1 Spring analogy method . . . 14

2.5.2 Laplace smoothing method . . . 14

2.5.3 Radial basis functions interpolation method . . . 14

2.6 Design of Computational Aeroelasticity Tool . . . 15

2.7 Summary . . . 20

3 Equivalent Beam Model **21**

3.1 Introduction . . . 21

3.2 Inverse Design . . . 21

3.3 Cases Descriptions . . . 23

3.4 Results . . . 26

3.4.1 Single loading cases . . . 26

3.4.2 Multiple loading cases . . . 30

3.5 Summary . . . 33

4 Static Aeroelasticity: Validation of the Computational Aeroelasticity Tool **34**

4.1 Introduction . . . 34

4.2 Test Cases . . . 34

4.2.1 Geometry . . . 34

4.2.2 Structural models . . . 35

4.2.3 Aerodynamic model . . . 37

4.2.4 Aeroelastic interface . . . 37

4.2.5 Mesh deformation . . . 37

4.2.6 Flow conditions . . . 39

4.2.7 Test cases summary 39
4.3 Results 40
4.3.1 Case A: Wing-only geometry, wing-box structural model, $M = 0.85$ 40
4.3.2 Cases B: Wing-only geometry, beam stick and wing-box structural models . . . 44
4.3.3 Case C: Wing-fuselage geometry, wing-box structural model, $M = 0.85$ 47
4.4 Summary 50

5 Aerodynamic Shape Optimization of Elastic Wing 51
5.1 Introduction 51
5.2 Principle of Aerodynamic Shape Optimization 52
5.3 Tools 52
5.4 Test Cases 54
5.4.1 Common description 54
5.4.2 Initial design 55
5.5 Results 56
5.5.1 Rigid wing optimization 56
5.5.2 Elastic wing optimization 59
5.6 Summary 62

6 Numerical Study of Benchmark Supercritical Wing at Flutter Condition 63
6.1 Introduction 63
6.2 CFD-CSM Solver 63
6.2.1 CFD code Edge 63
6.2.2 Structural solver 64
6.2.3 Coupling scheme 64
6.2.4 Mesh deformation 64
6.3 Test Case 65
6.3.1 Experimental setup 65
6.3.2 Computational setup 65
6.4 Results 66
6.4.1 Time step convergence study 66
6.4.2 Influence of number of exchanges during each time step on flutter solution . . . 67
6.4.3 Estimate of flutter dynamic pressure 68
6.4.4 Estimate of the flutter boundary 70
6.5 Summary 71

7 Conclusions 72
7.1 Outcome of the Thesis 72
7.2 Conclusion 72
7.3 Perspectives 73

List of Symbols 80

List of Acronyms 81

Appendices 82

A Matlab Code of Structural Solver 83

B Computational Aeroelasticity Tool - Convergence Tests 91

Chapter 1

Introduction

1.1 Overview

The aeroelasticity is a discipline studying interaction of fluid, elastic and inertial forces, thus connecting together the fields of aerodynamics, elasticity and dynamics. The phenomena occurring due to interaction of the mentioned forces result in the performance changing or structural damaging effects. Therefore, the aeroelasticity concerns the aircraft designers since the earliest years of the aviation. The experimental research in this field started during development of the first airplane. That time the Wright brothers investigated the effect of the wing warping, applied to their Flyer biplane to provide the roll control, and also adverse effect of a propeller blade torsional deformation on the trust. Perhaps the first experience with the engined aircraft failure due to aeroelastic phenomenon was the flight of Samuel P. Langley's monoplane. His attempt of the first flight failed due to the wing twist off caused by insufficient wing torsional stiffness. The phenomenon is known as the torsional divergence. During following years, the airplane designers encountered other aeroelastic phenomena such as the elevator flutter caused by insufficient torsional stiffness of the fuselage and the tail combined with unsuitable design of the control surfaces drive. The return to the monoplane design and increased speed of aircrafts brought the need to solve other aeroelastic issues - aileron effectiveness loss or reversal, wing flutter or wing lift redistribution. With reaching the supersonic and hypersonic speed new aeroelastic phenomena connected with a shock wave oscillation occurred, such as control surface buzz or panel flutter [1].

In the early years of the aviation the solution of aeroelastic problems was sought by a trial and error. Thus, problems were solved in late stage of the design process and often led to accidents during flight tests. In the course of the aeronautical engineering development, the theoretical investigations and research were conducted to understand the aeroelastic phenomena. The theories solving unsteady aerodynamics were established by Wagner [2] and Küssner [3]. The developments in the unsteady aerodynamics allowed to create theories predicting the wing flutter in subsonic speeds by Küssner [4] and later by Theodorsen [5]. The Theodorsen's work created a basis for the strip theory, which has been further developed [6]. Nowadays, commonly applied method for aerodynamic prediction in aeroelastic computations is based on the doublet-lattice method, [7, 8]. This essentially linear method is capable of relatively accurate prediction in the subsonic conditions with no flow separation but fails in transonic flow or flow with extensive separation. Therefore, it has been common practice to correct results of the linear aerodynamic methods by the wind tunnel measurements. The advances in the development of computational methods based on the finite volumes started an extensive research of the aeroelastic solvers employing CFD simulations potentially reducing the number of tunnel or flight tests. The research in this field probably started by Bendiksen [9], who has been followed by Lee-Rausch [10], Alonso [11], Thomson [12], Feng [13] and many others, focusing on both static and dynamic aeroelastic simulations. The recently established activity in NASA [14, 15] focusing on validation of tools for high-fidelity flutter predictions in transonic speeds highlights the effort given to the research in this field.

1.2 The Objective of the Thesis

The trend in aircraft design is focused on the increase of aircraft overall efficiency. One of the aspects is the fuel consumption decrease, aiming to reduce the operational costs and emission of greenhouse gases. This can be obviously realized by the optimization of a propulsion system or by design of a lighter airframes and aerodynamically more efficient shapes, e.g. long slender wing. The combination of last two mentioned leads to increased flexibility of the airframe and consequently to the change of the aerodynamic characteristics, stability margin or control surface efficiency. In addition, compared to stiffer wing, the aeroelastic effect can become stronger and more easily excited by rigid body motions or an input of a flight control system. The potential aeroelastic phenomena can occur in large range of speeds involving transonic regime, where the non-linear flow effects significantly influence the flutter speed. Moreover, the complex flow around the wing, due to its shape and interaction with other components, such as nacelles, pylons, an engine flow and a fuselage, makes the aerodynamic design and analysis the challenging task.

All mentioned aspects ask for a simulation tool which is able to:

- cover full range of the flight envelope - increasing the design efficiency [16],
- include the aeroelastic effects in early design stage - minimizing the need for often costly redesign in later stages,
- resolve the non-linear aerodynamics - allowing the design optimization of the complex geometries.

The mentioned requirements disqualify common aeroelastic tools employing unsteady aerodynamic solvers based on linear methods such as a strip theory or a doublet-lattice method. The progress in Computational Fluid Dynamic (CFD) in the areas of numerical scheme stability, code effectiveness and turbulence modeling, created a reliable tool for unsteady non-linear aerodynamic predictions. Therefore, the CFD is the convenient method for application in high-fidelity aeroelastic simulation tool covering all mentioned requirements.

The thesis objective is to create a tool for simulation and design optimization of static aeroelastic models of the aircraft. The tool will be applicable in:

- computation of aerodynamic characteristics of an elastic airplane,
- aerodynamic design optimization of elastic wing (aircraft) giving the result closer to reality,
- conceptual design based on aero-structural optimization - design of aircraft shape together with airframe structure aiming for improvement of both aerodynamic and structural efficiencies.

1.3 The Novelty of the Thesis

The research in the field of aeroelastic simulation and design is important and concerns large variety of computational tools ranging from CFD capable to accept large deformation of boundaries, employing techniques such as mesh deformation [17, 18, 19, 20], chimera grids [21], immersed or embedded boundary conditions [22, 23, 24], through efficient and accurate time integration schemes of coupled fluid-structure equations [25, 26, 27], spatial coupling [28, 29] or recently the adjoint of coupled fluid-structure equations of high-fidelity solvers in multidisciplinary optimization [30, 31].

The thesis is developing the activity at Institute of Aerospace Engineering focused on aeroelastic simulation and design using modern techniques like the fluid-structure interaction and adjoint method [32].

Chapter 2

Static Aeroelastic Computations

2.1 Introduction

The research in field of high fidelity aeroelastic computation has been conducted since past few years. The advances in Computational Fluid Dynamics, creating the standard tool for non-linear aerodynamic predictions, commenced the interest in coupling high fidelity flow solvers with the already matured Finite Element Method. The research focuses on wide range of computational methods involved in the aeroelastic simulations, as it was mentioned previously in the thesis, ranging from mesh deformation methods [17, 18, 19, 20] through effective spatial and temporal coupling of the essentially different domains [28, 29, 25, 26, 27].

In this chapter, methods involved in static aeroelastic computation are described. The emphasis is given to fluid-structure interaction methods with focus on the description of the basic principles and methods involved. The summary of the aerodynamic and structural modeling follows and is complemented by the information about the solvers applied in the thesis. The end of the chapter is dedicated to the implementation of the static aeroelastic simulation tool applicable for the analysis and design optimization.

2.2 Fluid-Structure Interaction

Two main approaches for fluid-structure interaction are distinguishable - monolithic and partitioned. The monolithic approach, often referred as the strongly coupled, combines the fluid and the structural state equations together and treats them as a single system of equations governing both problems guaranteeing a conservation of properties at the fluid-structure interface. The interaction between domains is treated synchronously. The advantage is the robustness of the approach. The implementation requires the special solver, thus the existing well-established fluid and structure solver cannot be used.

The partitioned formulation approach allows to combine the complex domains described by different approaches and can differ in size by order of magnitude. The difference in size of the systems is common in the real applications when usually a flow domain is much larger than a structural domain. The necessary information obtained by arbitrary flow and structural solvers is exchanged on the defined interface according to chosen coupling scheme which should satisfy several criteria - conservation of energy and loads, accuracy and efficiency.

2.2.1 Principle of coupling

The principle of the fluid-structure coupling is based on the conservation of the virtual work satisfying the conservation of energy. The virtual work performed by the aerodynamic load must be equal to the virtual work of the structural forces:

$$\delta W = \mathbf{F}_s^T \delta \mathbf{u}_s = \mathbf{F}_f^T \delta \mathbf{u}_f \qquad (2.1)$$

where $\mathbf{F}_s$ is vector of forces acting on structural nodes, $\mathbf{F}_f$ is vector of forces at fluid nodes, $\mathbf{u}_s$ and $\mathbf{u}_f$ are structural and fluid nodes displacement vectors, respectively.

The fluid-structure coupling is often expressed by introduction of a coupling matrix $\mathbf{H}$ giving a relation between the displacement vectors of the fluid and the structure meshes:

$$\mathbf{u}_f = \mathbf{H}\mathbf{u}_s \tag{2.2}$$

The combination of equations 2.1 and 2.2 gives a relation:

$$\mathbf{F}_s = \mathbf{H}^T\mathbf{F}_f \tag{2.3}$$

Thus, if matrix $\mathbf{H}$ satisfies the mentioned conservation criteria, it can be used for the transformation of the structural displacements to the fluid mesh and once it is transposed for the transformation of the aerodynamic load to the forces acting on the structural nodes.

2.2.2 Transformation methods

During the research on fluid-structure interaction, relatively large number of coupling methods has been defined and applied. Reviews of methods were published, among others, by Hounjet and Meijer [33], Smith et. al. [34] and Boer [35]. Many articles focusing on the particular coupling method were also published by Cerbral and Löhner [36], Beckert [37], Zwaan and Prananta [38], Wendland [29], Rendall and Allen [28, 39] and others.

The nearest neighbor interpolation

The nearest neighbor interpolation is a simple method of the information transfer from mesh S (structure) to mesh F (fluid). For the given point x_F in mesh F the closest point x_S in mesh S is found. Consequently, the value of variable in x_F is taken as the same as in point x_S. Thus, the coupling matrix $\mathbf{H}$ is a Boolean matrix.

Weighted residual methods

The initial assumption for the method is the conservation of displacements on the interface, which is given in continuous form:

$$u_s(x) = u_f(x) \text{ on } \Gamma \tag{2.4}$$

A weighted residual method is used for approximate solution of this equation. Thus, both sides of the equation are multiplied by a set of weighting functions ϕ_k and integrated over the interface. The quantities $u_s(x)$ and $u_f(x)$ are approximated by:

$$u_s(x) = \sum_{i=1}^{n_s} N_s^i(x)\mathbf{u}_{si} \tag{2.5}$$

$$u_f(x) = \sum_{j=1}^{n_f} N_f^j(x)\mathbf{u}_{fj} \tag{2.6}$$

where vectors $\mathbf{u}_{si}$ and $\mathbf{u}_{fj}$ contain the values of the u_s and u_f at the interface (n_s and n_f are the number of structure and flow unknowns at the interface), N_s and N_f are the basis functions for the structure and the flow.

The mentioned operations lead to equation:

$$\int_\Gamma \phi_k(x) \sum_{i=1}^{n_s} N_s^i(x)\mathbf{u}_{si}dx = \int_\Gamma \phi_k(x) \sum_{j=1}^{n_f} N_f^j(x)\mathbf{u}_{fj}dx. \tag{2.7}$$

The equation can be solved using the Galerkin method. There are two possible values of the weighting function ϕ: the basis function of the structure N_s or the flow N_f. So, the equation 2.7 becomes:

$$\sum_{i=1}^{n_s}\left[\int_\Gamma N_\alpha^k(x)N_s^i(x)dx\right]\mathbf{u}_{s_i} = \sum_{j=1}^{n_f}\left[\int_\Gamma N_\alpha^k(x)N_f^j(x)dx\right]\mathbf{u}_{f_j} \tag{2.8}$$
$$k = 1,\ \cdots, n_\alpha$$

where $\alpha \in \{f, s\}$ and the integral on the left-hand side is denoted as $A_{\alpha s}^{ki}$ while the one on the right-hand side is $A_{\alpha f}^{kj}$. This equation can be written in the matrix form as

$$\mathbf{A}_{\alpha s}\mathbf{u}_s = \mathbf{A}_{\alpha f}\mathbf{u}_f, \tag{2.9}$$

Then the displacements in fluid are defined as

$$\mathbf{u}_f = \mathbf{A}_{ff}^{-1}\mathbf{A}_{fs}\mathbf{u}_s \tag{2.10}$$

Consequently, the transformation matrix is

$$\mathbf{H} = \mathbf{A}_{ff}^{-1}\mathbf{A}_{fs} \tag{2.11}$$

The last step is the selection of the interface for integration of integrals in Eq. 2.8. For the matrix $\mathbf{A}_{ff}$ it is the best to integrate over fluid domain Γ_f, because the N_f^k and N_f^j are known. Also for integration in matrix $\mathbf{A}_{fs}$ the Γ_f is chosen, in order to satisfy the global conservation of the forces (details are in [35]). Because N_s is defined only on the discretized structure interface, a projection method must be employed, e.g. Gauss interpolation [36] or Intersection method [35].

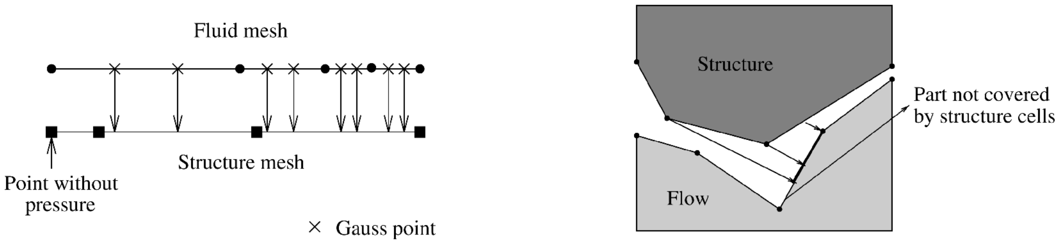


Figure 2.1: Issue in projection - Gauss interpolation (left) and Intersection method (source [35])

Essentially, Gauss interpolation uses Gauss integration for computation of integrals in $\mathbf{A}_{fs}$ with choice of integration points in fluid mesh cells and projection to the structure mesh. In certain cases, some structural points are not taken into account by projection and therefore they do not receive force information from fluid mesh, see Figure 2.1. To fix this, more Gauss points in the fluid cell must be used. The Gauss points are projected onto structural mesh employing search algorithm and orthogonal projection with minimum distance criterion.

The intersection method uses projection of elements from one mesh to element of other mesh. The relative area of projection gives proportion of the values in certain element which is taken into account. In order to satisfy the conservation of loads the integral is taken over flow interface. Thus, the structure elements are projected on the flow mesh. In certain situation, i.e. when convex/concave surfaces are present (see Fig. 2.1), some parts of fluid cells may not be covered by projection. Consequently, the pressure force of the flow interface is not completely taken into account and the global conservation of forces is not fulfilled. The method requires same projection algorithm as the previous one.

Method of finite interpolation element

The finite interpolation elements are special type of finite elements with no stiffness, mass or any material properties. They relate displacements (rotation) of the element node with those in any

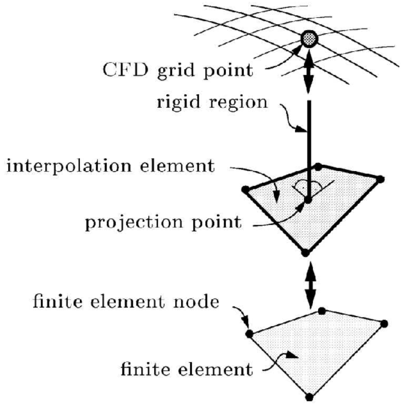


Figure 2.2: Principle of coupling by finite interpolation elements (source [37])

element point employing the shape functions. Usually, this kind of elements is used as a connection of compatible substructures in finite element analysis.

The principle of the coupling is illustrated in Figure 2.2. A node on the fluid grid is projected on the finite interpolation element. The space between the element and the fluid node is rigid, therefore the aerodynamic load is transformed in terms of the equilibrium of forces and moments. Those forces and moments can be interpolated to the structure nodes by using of shape function of the finite interpolation element. Vice versa, the displacements and rotation of the structure nodes are interpolated back to the projected points by the same shape functions.

The relation between the displacement vector of the projected point $\mathbf{u}_{\hat{s}}$ and the displacement vector of the structural node $\mathbf{u}_s$ can be formulated as

$$\mathbf{u}_{\hat{s}} = \hat{\mathbf{H}}_\mathbf{s}\mathbf{u}_s, \tag{2.12}$$

the matrix $\hat{\mathbf{H}}_\mathbf{s}$ contains n_s diagonal matrices $\mathbf{H}_{s_i}$ (the size is 6x6 in case of six degrees of freedom), which are composed of values resulting from the evaluation of the shape functions of the finite interpolation element. The relation between displacements of fluid node and its projection point is

$$\mathbf{u}_f = \mathbf{D}_{\hat{s}}\mathbf{u}_{\hat{s}}, \tag{2.13}$$

the matrix $\mathbf{D}_{\hat{s}}$ provides the rigid transformation of the displacements and rotations.

In order to formulate the transformation matrix $\mathbf{H}$, the equations 2.12 and 2.13 are combined together. Then

$$\mathbf{H} = \mathbf{D}_{\hat{s}}\hat{\mathbf{H}}_\mathbf{s} \tag{2.14}$$

The projection of the fluid nodes onto interpolation element can be solved by the orthogonal projection.

Radial basis functions

Radial basis functions [40], [19] are flexible and well-established tool for multivariate interpolation. The displacements at the fluid and the structure are approximated by an interpolant which has the form:

$$s(\mathbf{x}) = \sum_{j=1}^{N} \gamma_j \phi(\|\mathbf{x} - \mathbf{x}_j\|) + h(\mathbf{x}) \tag{2.15}$$

where ϕ is a given basis function, coefficients $\gamma_j \in \mathbb{R}$, the $\mathbf{x}_j$ are centers with known values (structural points), $h(\mathbf{x})$ are first degree polynomials and $\|\cdot\|$ denotes Euclidean norm. In many cases it is convenient to scale the basis function with a shape parameter ϵ, then the basic function is replaced by $\phi_\epsilon(r) = \phi(\epsilon r)$. Common radial basis function are given in Table 2.1.

Table 2.1: Common radial basis functions

Radial basis function	$\phi(r)$
Spline type	$\|r\|^n$, n odd
Thin plate spline	$\|r\|^n \log \|r\|$, n even
Multiquadric	$\sqrt{1+r^2}$
Inverse multiquadric	$(\sqrt{1+r^2})^{-1}$
Inverse quadric	$(1+r^2)^{-1}$
Gaussian	e^{-r^2}
Multi-quadric biharmonic splines	$\sqrt{r^2+a^2}$

In [[40]], it is shown that the coupling matrix $\mathbf{H}$ is:

$$\mathbf{H} = \mathbf{A}_{fs}\mathbf{C}_{ss}^{-1} \tag{2.16}$$

Both matrices at the right-hand side come from RBF approach. The square interpolation matrix $\mathbf{C}_{ss}$ of size $N_s \times N_s$ (N_s is a number of structural nodes) consists, among others, of values $\phi(\|\mathbf{x}_{s_i} - \mathbf{x}_{s_j}\|)$, while the radial basis function is evaluated only on structural nodes. The matrix $\mathbf{A}_{fs}$ (of size $N_f \times N_s$) depends on both fluid and structural nodes.

The interpolation by radial basis function is dependent on the radius of the support r, which can be varied by the shape parameter ϵ. A large support radius gives good approximation, but if the radius is too large it leads to singular matrices. On the other hand, a small support radius yields less accurate interpolation. In many cases, it would be useful to vary the radius from structure node to structure node but according to the theory it is not possible. In practice, the support radius should guarantee enough points are covered, on the other hand, points far away should have no influence. Therefore, the support radius for fluid-structure interaction should be chosen as large as the maximum distance of all centers from their nearest neighbors in both meshes.

Other important choice is a choice of the radial basis function itself. According to [35], the most robust, cost effective and accurate are:

- Multi-quadric biharmonic splines

$$\phi(r) = \sqrt{r^2+a^2} \tag{2.17}$$

- Thin-plate spline

$$\phi(r) = |r|^n \log |r|, n\text{even} \tag{2.18}$$

The parameter a controls the shape of the basis function.

The unique solution of the interpolation can be guaranteed only when at least four structural points are not in plane [29]. This is not often fulfilled, e.g. if the simple beam structural model is used. In this case, it is possible to solve the linear system but the solution is not unique.

2.2.3 Chosen coupling method

The comparison done by Boer [35] implies all methods fulfill the conservation of the virtual work and pressure forces over the interface, except of special case of concave/convex surfaces combined with large difference between size of the fluid and the structure grids. In this case, the intersection method does not conserve pressure forces. The nearest neighbor interpolation method is burdened by the error due to interpolation step which is larger than the discretization error.

For further application in the thesis, the radial basis function method was chosen. The reasons are the accuracy and no need of complicated and computationally expensive search and projection algorithm as in case of the weighted residual or the finite interpolation elements methods.

2.3 Aerodynamics Modeling

The thesis objective is to create tool for high fidelity aeroelastic simulations and multidisciplinary design optimization. The tool should be applicable for flight speeds ranging over full flight envelope, potentially involving transonic regime or conditions with strong flow separation. Moreover, the aircraft design involves flow solution around complex geometries, such as wing-fuselage configuration with nacelles or pylons. Those requirements disqualify the linear aerodynamic prediction tools such as doublet-lattice method. Since the Computational Fluid Dynamic (CFD) solvers fulfill mentioned requirements for the flow solution, the CFD solver will be used for aerodynamic prediction in the aeroelastic computational tool.

2.3.1 Governing equations

Viscous flow

The flow of compressible viscous fluid is described by equations expressing the conservation laws - conservation of the fluid mass, conservation of the momentum and conservation of the energy. The derivation of the particular equations can be found in numerous textbooks of the fluid mechanics. Versteeg [41], focused on finite volume method, gave the Navier-Stokes equations governing the time-dependent three-dimensional fluid flow and heat transfer of the compressible viscous fluid in form:

- Mass

$$\frac{\partial \rho}{\partial t} + \mathrm{div}(\rho \mathbf{u}) = 0 \tag{2.19}$$

- x-momentum

$$\frac{\partial \rho u}{\partial t} + \mathrm{div}(\rho u \mathbf{u}) = -\frac{\partial p}{\partial x} + \mathrm{div}(\mu \nabla u) + S_{Mx} \tag{2.20}$$

- y-momentum

$$\frac{\partial \rho v}{\partial t} + \mathrm{div}(\rho v \mathbf{u}) = -\frac{\partial p}{\partial y} + \mathrm{div}(\mu \nabla v) + S_{My} \tag{2.21}$$

- z-momentum

$$\frac{\partial \rho w}{\partial t} + \mathrm{div}(\rho w \mathbf{u}) = -\frac{\partial p}{\partial z} + \mathrm{div}(\mu \nabla w) + S_{Mz} \tag{2.22}$$

- Internal energy

$$\frac{\partial \rho i}{\partial t} + \mathrm{div}(\rho i \mathbf{u}) = -p div \mathbf{u} + \mathrm{div}(k \nabla T) + \Phi + S_i \tag{2.23}$$

- State equation

$$p = p(\rho, T) \text{ and } i = i(\rho, T) \tag{2.24}$$

where ρ is density of fluid, t is time, $\mathbf{u}$ is fluid velocity vector with components u, v, w, μ is dynamic viscosity, S_{Mi} is momentum source (i denotes x,y and z directions), Φ is dissipation function, T is temperature, p is pressure and i is internal energy.

Equations in this form fully describe laminar flow. In common engineering practice the solution of high Reynolds flow, involving the boundary layer transition from laminar to turbulent, is required. The usual practice is to solve Reynolds averaged Navier-Stokes equations with additional equations modeling either only turbulent boundary layer or a transition of laminar to turbulent boundary layer.

Inviscid flow

In certain cases, the viscous effect can be neglected. It is typical for aeroelastic applications when the flow is assumed fully attached with small perturbations [42]. Therefore, the viscous and thermal conductivity terms in Navier-Stokes equations are dropped and the system of equations reduces to the Euler equations governing the inviscid flow:

- Mass

$$\frac{\partial \rho}{\partial t} + \mathrm{div}(\rho \mathbf{u}) = 0 \tag{2.25}$$

- x-momentum

$$\frac{\partial \rho u}{\partial t} + \mathrm{div}(\rho u \mathbf{u}) = -\frac{\partial p}{\partial x} + S_{Mx} \tag{2.26}$$

- y-momentum

$$\frac{\partial \rho v}{\partial t} + \mathrm{div}(\rho v \mathbf{u}) = -\frac{\partial p}{\partial y} + S_{My} \tag{2.27}$$

- z-momentum

$$\frac{\partial \rho w}{\partial t} + \mathrm{div}(\rho w \mathbf{u}) = -\frac{\partial p}{\partial z} + S_{Mz} \tag{2.28}$$

- State equation

$$p = p(\rho, T) \tag{2.29}$$

2.3.2 Flow solver

The flow solver used in the thesis is CFD code Edge [43]. It is finite volume solver for unstructured grids which can solve 2D and 3D Euler and RANS equations, as well as the adjoint of the Euler and NS (frozen viscosity) equations [32]. The time integration uses the fourth order Runge-Kutta scheme. It employs local-time-stepping, local low-speed preconditioning, multi-grid and dual-time-stepping for steady-state and time-dependent problems. For the unsteady cases, the employed numerical scheme is a dual-time-stepping scheme[44]. The data structure of the code is edge-based. The solver can be run in parallel on a number of processors to efficiently solve large flow cases.

2.4 Structural Modeling

The simulation and prediction of aeroelastic effects require an appropriate structural model. The model should be able to describe behaviour of loaded structure in sufficient extent appropriate to its application. Slender structures, which are typical for airplane wings, can be represented by a beam stick model. The application of this model is also allowed by high chord-wise rigidity of the wing due to ribs. Usually, the beam is placed to the position of a wing elastic axis, which is a line of points where a bending loading does not produce torsional deformation and vice versa. Therefore, decoupling of bending and twisting is allowed and Euler-Bernoulli beam elements can be applied.

2.4.1 Matlab based finite element solver

In the frame of the thesis, the finite element solver was programmed in Matlab environment. It is linear elasticity solver working in two modes:

- Mode 1: Beam finite elements mode
- Mode 2: Prescribed stiffness and mass matrices mode

The purpose of this solver is to overcome cumbersome communication with commercial FEM packages via input files and to allow future development focused on direct communication between solvers. The important parts of finite element preprocessor and solver code in Matlab are given in Appendix A.

The first mode can solve static deformation and modal analysis of the model consisting of beam elements (either Euler-Bernoulli or Timoshenko). Its advantage is capability of direct input of a model which is applicable for model parameterization purposes. Input for this mode is:

- nodes - identification number and coordinations
- beam elements - connections, relevant structural characteristics
- rigid elements - connecting node of the beam element with free node - useful for visualization of torsion and for fluid-structure coupling
- material properties - isotropic material for entire model
- constraints definition
- loading definition - external point forces and inertial loading

The second mode is capable of solving model consisting of arbitrary structural finite elements but model size is limited by size of memory Matlab can allocate. The required input is:

- stiffness and mass matrices
- order of degrees of freedoms in matrices
- constraints definition
- loading definition - external point forces and inertial loading

Solution of linear static elasticity by finite elements

Solution of a linear static elastic problem using finite element formulation leads to system of linear algebraic equations in the form:

$$\mathbf{K}\mathbf{u} = \mathbf{F}, \tag{2.30}$$

where $\mathbf{K}$ is a stiffness matrix, $\mathbf{u}$ is a vector of nodal displacements and rotations and $\mathbf{F}$ is vector of corresponding nodal forces and moments. The stiffness matrix consists of components representing a stiffness of applied finite elements.

Implementation of the Prescribed stiffness and mass matrices mode

An arbitrary finite element model can be solved in this mode. The stiffness and mass matrices must be preprocessed in an external finite element preprocessor (in frame of the thesis, the MSC Patran/Nastran was used). Then matrices together with definition of degree of freedom order are loaded to the solver. According to definition of constrained degrees of freedom, corresponding rows and column in the stiffness matrix and components of the load vector are removed. Finally, the system of equations is solved which gives vector of displacements and rotations of unconstrained nodal degrees of freedom.

Implementation of the Beam finite elements mode

This mode serves as preprocessor and solver for a finite element model consisting of beam elements with 6 degrees of freedom, namely: axial displacement, traverse displacements in y_L and z_L directions and rotations about x_L, y_L and z_L axes, see Figure 2.3. The implementation is based on theory presented in [45]. In preprocessing step local stiffness and mass matrices of beam elements are created in a local coordinate system and transformed to a global coordinate system. Consequently, the global assembly stiffness and mass matrices are created according to element connections in the model. Then constrained degrees of freedom are removed from stiffness matrix and forces vector and the system of equations, Equation 2.30, is solved.

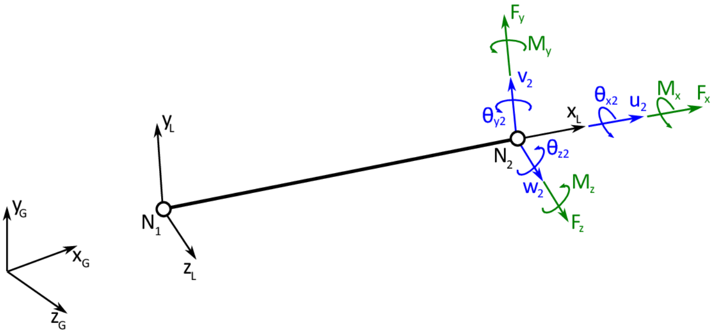


Figure 2.3: Beam element located in global coordinates with defined local coordinates and corresponding local displacements (blue), forces and moments (green)

The linear beam theory applies for small (or even infinitesimal) deflections and as result of a bending loading it gives transverse displacements only. Therefore, large deflection due to bending causes a change of the beam length. In fluid-structure interaction procedure, it would lead to change of corresponding aerodynamic surface (e.g. wing surface). This property could be fixed by introduction of fictitious displacement in direction of beam element x axis (see Figure 2.4). For not too large angles of element rotation θ due to transverse displacement v the fictitious displacement is:

$$u_f \backsimeq -\frac{1}{2l}(v_1 - v_2)^2,\ [45] \tag{2.31}$$

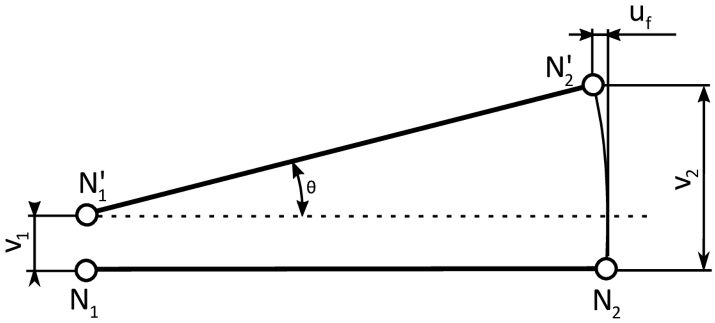


Figure 2.4: Fictitious displacement due to bending transverse displacement

Implementation of inertial forces loading

In both solver modes, the loading by inertial forces is implemented. Inertial forces are calculated according to Newton's second law of motion with an assumption that only inertial forces act on a structure. Thus, the inertial forces can be calculated from the equation:

$$\mathbf{F_i} = \mathbf{M\ddot{u}}, \tag{2.32}$$

where $\mathbf{M}$, $\mathbf{F_i}$ and $\mathbf{\ddot{u}}$ are mass matrix, vector of nodal inertial forces and vector of nodal accelerations, respectively.

Validation of the solver by MSC Patran/Nastran

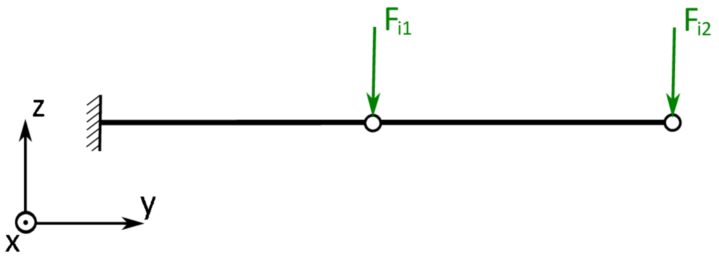


Figure 2.5: Validation of solver - beam loading by gravity forces

The solver was validated by comparison with results given by Patran/Nastran. Two models were created and loaded by gravity loading. The first one was cantilevered beam with constant cross-sectional characteristics discretized by two beam elements, see Figure 2.5. The beam cross-sectional and material characteristics are listed in the Table 2.2.

The second model is wing-box model described in Section 4.2.2.

The resultant deformations closely agree with Nastran results in all cases - deformation of beam model solved by both modes of Matlab based solver (Figure 2.6 and Table 2.3) and wing-box model deformation solved by Mode 2 (Figure 2.7 and Table 2.4). Differences in deformations are of the order of hundredths of percent.

Parameter	Value
Length, L	2000 m
Area, A	0.0312 m^2
Second moment of area about x-axis, I_x	0.002 m^4
Second moment of area about z-axis,I_z	0.002 m^4
Torsional constant, J	0.00036 m^4
Young's modulus, E	7.31e10 Pa
Poisson constant, ν	0.33
Material density, ρ	2850 $kg.m^{-3}$

Table 2.2: Flow conditions

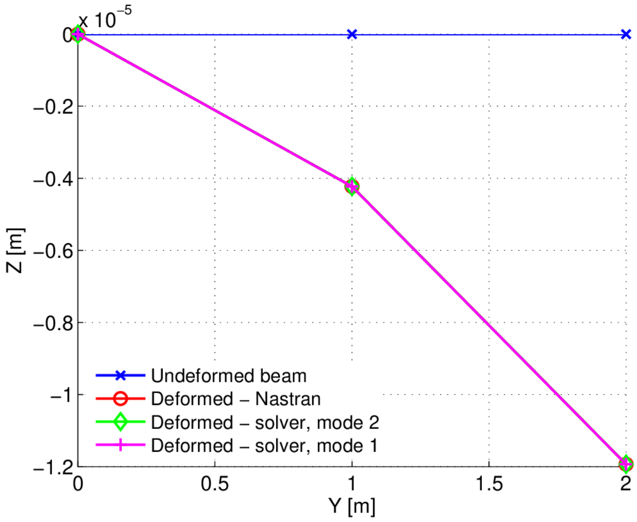


Figure 2.6: Deformation of the beam finite element model, gravity loading, comparison of Nastran and Matlab solvers

Node	Nastran	Solver - Mode 2	Solver - Mode 1
1	0.0	0.0	0.0
2	-0.0423	-0.0423	-0.0423
3	-0.1193	-0.1193	-0.1193

Table 2.3: Vertical displacements of the beam model loaded by gravity forces, comparison of Nastran and Matlab solvers

	Nastran	Solver - Mode 2	Difference
x-displ	0.0080809	0.0080807	0.00%
y-displ	0.0366014	0.0365988	0.01%
z-displ	0.3145667	0.3145469	0.01%

Table 2.4: Absolute values of wing-box model tip displacement due to gravity loading, comparison of Nastran and Matlab solvers.

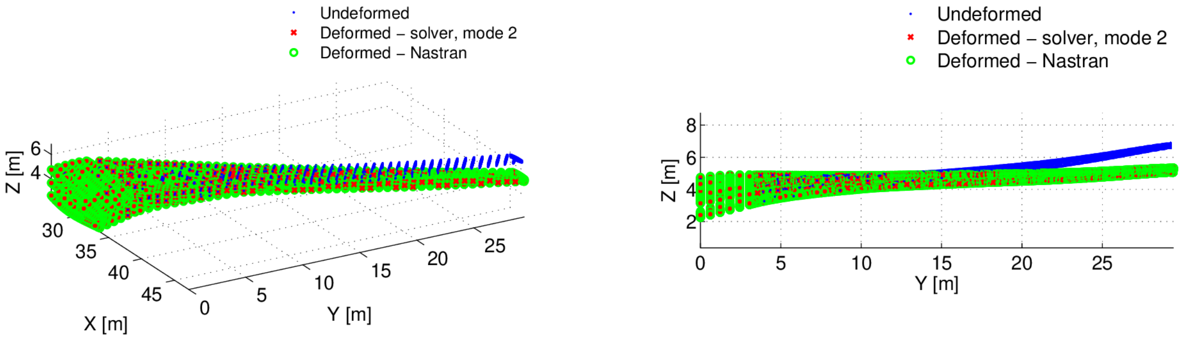


Figure 2.7: Deformation of the wing-box finite element model, gravity loading, comparison of Nastran and Matlab solvers (points represent nodes of structural model, deformation is 5x magnified)

2.5 Mesh Deformation

The motion of deformable surface of an aeroelastic model must be captured by a fluid computational grid prior to calculation of new flow solution. Two methods of moving geometry treatment exist, remeshing and grid deformation. The first mentioned allows to capture arbitrarily large geometry deformation, but at high computational cost connected with recalculation of entire volume mesh. Moreover, risk of physical conservation loss exists due to possibility of large changes in the grid which may lead to reduced local computational accuracy.

Therefore, development of mesh deformation techniques, such as spring analogy, Laplace smoothing and radial basis functions interpolation, has began in recent years. Their advantage is conservation of mesh topology, i.e. number of elements, nodal connectivity and generally lower computational cost compared to the remeshing. Commonly, mesh deformation methods suffer by high risk of inverted elements occurrence as result of large geometry deformation.

2.5.1 Spring analogy method

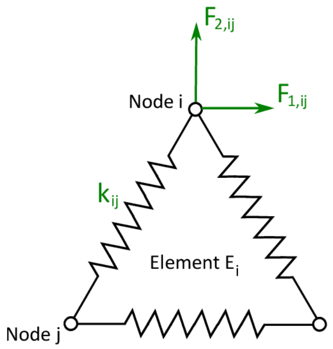


Figure 2.8: Fictitious springs connecting nodes of a 2D element

The method, originally proposed by Batina [46], is based on treatment of computational mesh as a network of springs connecting mesh nodes as it is illustrated for one two-dimensional element in Figure 2.8. Deformation of the boundary is propagated to the volume mesh on the basis of static equilibrium between fictitious forces which are proportional to nodal displacements.

Since, the method treats only element edges, not element as the whole, there is high possibility of element inversion. Thus, several attempts were made in order to improve the technique by introduction of additional fictitious springs, e.g. torsional spring controlling angles between two connected edges [47] or springs connecting tetrahedron node with other point on the element [48].

2.5.2 Laplace smoothing method

The mesh deformation method originate in Laplace smoothing which was originally employed for improvement of the computational grids [49]. Essentially, the propagation of boundary deformation into the interior is based on iterative movement of the mesh nodes towards the center of the polygon (2D) or polyhedron (3D) created by adjacent nodes. The method is prone to produce inverted cells, therefore it often fails to propagate large boundary movement. The effort was spent in order to overcome this issue [50, 51].

2.5.3 Radial basis functions interpolation method

This method proposed in [52, 19] applies the similar idea to the one used in the fluid-structure interaction where the movement of the control points defined on the structure grid is interpolated on the CFD surface mesh. In application for CFD grid deformation, the control points are defined on the movable boundary of the fluid mesh. The boundary movement is interpolated into the fluid volume mesh. Compared to the spring analogy and Laplace methods, the computational cost is low, once the interpolation matrix is created. Method can handle large deformations and is applicable for structured, unstructured and hybrid meshes, because it is independent of the mesh connectivity. From the method nature, the boundary nodes movement is not exactly recovered in the resulting deformed volume mesh. The method performance depends on the RBF type and on the choice of the support radius.

2.6 Design of Computational Aeroelasticity Tool

The implementation of the computational aeroelasticity tool is based on the partitioned (coupled-field) formulation of the fluid-structure interaction. Therefore, it is possible to couple arbitrary separate flow and structure solvers independently of each other. Thus, the best suiting solver for particular domain and application can be employed. Moreover, the solvers can be separately improved and maintained to comply the state-of-the-art level in the specific field.

The basic task of the tool is sharing appropriate information between solvers on defined interface. The formulation of the interface employing Radial Basis Functions (RBF) ensures ability to couple independently discretized domains differing in size by several orders of magnitude. At the same time, the formulation satisfies the conservation of energy and loads and it is accurate and efficient [35].

Basic principle

The principle of the computational aeroelasticity process is depicted in Figure 2.9. The initial step is definition of the CFD and CSM models and an appropriate interface connecting the models. In both domains, the surface nodes are picked to define the interface. It is an obvious choice because aerodynamic loading acts through surface pressure. The interface is described by a coupling matrix **H** which is constant during the aeroelastic computation. The main computational loop consists of sequential calls of the fluid and structure solvers and the relevant information transfer. In each iteration, the flow solution is calculated on the actual deformed shape and the forces on fluid surface mesh $\mathbf{F}_f$ are transformed to the forces on structural nodes $\mathbf{F}_s$ according the equation:

$$\mathbf{F}_s = \mathbf{H}^T \mathbf{F}_f \tag{2.33}$$

The force vector of the i-th surface node in the fluid mesh is calculated according to:

$$\mathbf{F}_{f_i} = p_i S_i \mathbf{n}_{n_i} \ , \tag{2.34}$$

where the p_i is the pressure in the i-th node, the S_i is the area of the control surface associated to the i-th boundary node and the $\mathbf{n}_{n_i}$ is the control surface normal vector.

Consequently, the forces are sent to the CSM solver and are applied on the initial structural model giving the vector of structural deformation $\mathbf{u}_s$ due to actual aerodynamic loading. The displacements of boundary nodes in the fluid domain are given by:

$$\mathbf{u}_f = \mathbf{H} \mathbf{u}_s \tag{2.35}$$

The CFD surface mesh deformation is propagated to the volume mesh employing appropriate algorithm capable to solve rather large surface mesh deformation. In the extreme case, if mesh deformation fails, a remeshing technique can be employed. At the end of the iteration, residual values are calculated and compared with convergence criteria. If the criteria are satisfied, the loop will end and a static equilibrium state of the aeroelastic model will be obtained. Otherwise, the loop will be repeated.

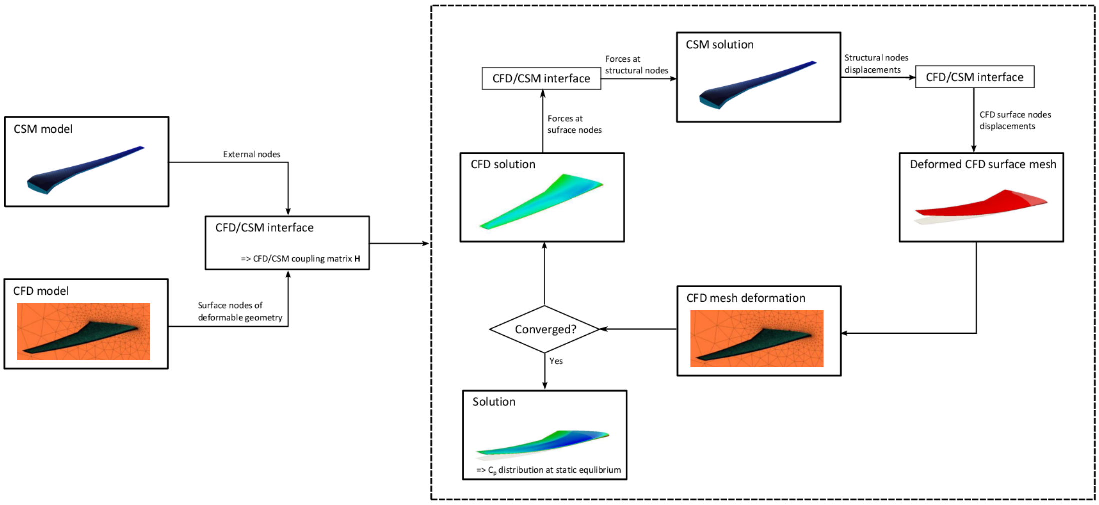

CSM model
External nodes
CFD/CSM interface
=> CFD/CSM coupling matrix **H**
CFD model
Surface nodes of deformable geometry
CFD/CSM interface
Forces at structural nodes
CSM solution
Structural nodes displacements
CFD/CSM interface
Forces at sufrace nodes
CFD surface nodes displacements
CFD solution
Deformed CFD surface mesh
CFD mesh deformation
Converged?
Yes
Solution
=> C_p distribution at static equlibrium

Figure 2.9: Description of fluid-structure interaction procedure

Implementation

From the perspective of implementation the acquisition of information and data transfer should be as fast as possible. There are two choices of data transfer, a file based using storage of data files on hard drive and online data exchange utilizing direct communication via random-access memory. Practical application of the first mentioned approach might be considered for the static aeroelasticity computation. In the case of dynamic aeroelasticity it would be very impractical due to time consuming I/O operations and large number of the data transfer during the solution of an unsteady problem.

The tool was implemented in the Matlab environment. Essentially, the tool is built on simpler principle employing data transfer using I/O operations via hard-copied files. The diagram describing the implementation is shown in Figure 2.10. The tool was designed to use arbitrary flow and structural solvers called by separate routines (cfd.m and fem.m in the diagram) whose function handles are fed to the main routine "run_fsi.m". The routine "cfd.m" returns the CFD forces for the input consisting of displacements of the CFD deformable surface. Thus, the routine must provide the CFD volume mesh deformation, solution of the flow field and calculation of the CFD surface forces. The routine "fem.m" provides structural nodes displacements for the input forces at the outer structural nodes.

The loop depicted in Figure 2.10 is repeated until the convergence criteria are met, i.e. when change between two consecutive normalized root mean square (RMS) values of CFD surface mesh nodes displacements is less than a prescribed value. The criterion must be satisfied for displacements in all directions: u_{f_x}, u_{f_y} and u_{f_z} and for one dimension is defined as (notation is without subscript defining direction):

$$\hat{u}_{f_j} - \hat{u}_{f_{j-1}} < t_s \tag{2.36}$$

where $\hat{u}_{f_j}$ in current j-th iteration is defined as:

$$\hat{u}_{f_j} = \frac{u_{f_{RMS,j}}}{u_{f_{RMS,1}}} \tag{2.37}$$

The $u_{f_{RMS,j}}$ and $u_{f_{RMS,1}}$ are RMS of fluid surface mesh displacements in given direction in j-th and the first coupling iteration, respectively, defined as follows:

$$u_{f_{RMS,j}} = \sqrt{\frac{1}{n}\sum_{i=1}^{N} u_{f_{i,j}}^2} \tag{2.38}$$

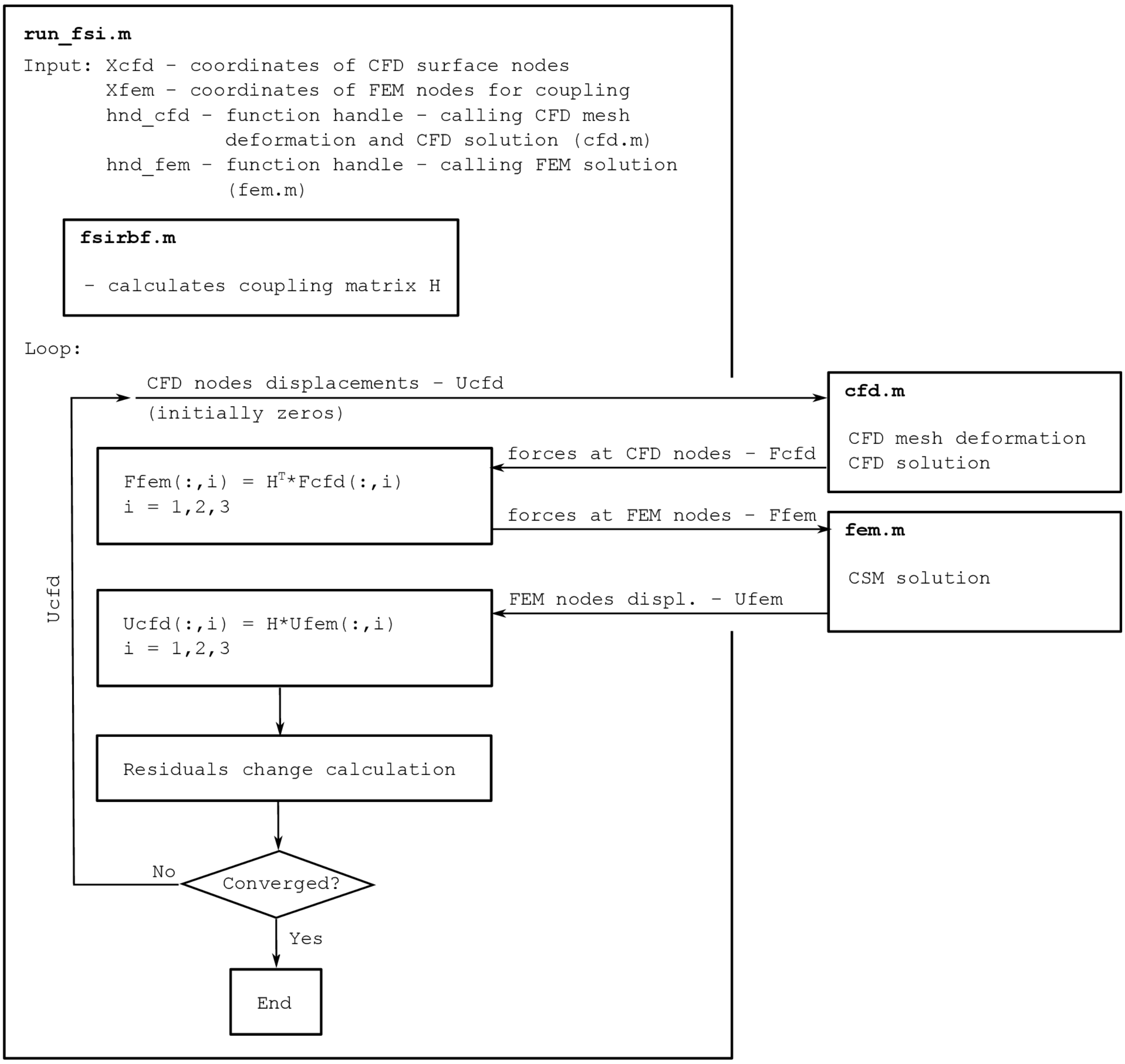

run_fsi.m
Input: Xcfd - coordinates of CFD surface nodes
Xfem - coordinates of FEM nodes for coupling
hnd_cfd - function handle - calling CFD mesh deformation and CFD solution (cfd.m)
hnd_fem - function handle - calling FEM solution (fem.m)
fsirbf.m
- calculates coupling matrix H
Loop:
CFD nodes displacements - Ucfd
(initially zeros)
cfd.m
CFD mesh deformation
CFD solution
forces at CFD nodes - Fcfd
Ffem(:,i) = H^T*Fcfd(:,i)
i = 1,2,3
forces at FEM nodes - Ffem
fem.m
CSM solution
FEM nodes displ. - Ufem
Ucfd
Ucfd(:,i) = H*Ufem(:,i)
i = 1,2,3
Residuals change calculation
No
Converged?
Yes
End

Figure 2.10: Diagram of the tool implementation in Matlab

Coupled problem convergence study

The influence of involved solvers settings on the convergence of the static aeroelastic solution was studied. The involved Matlab based linear CSM solver (described in Section 2.4.1) gives exact solution of the structural equations system for nearly no computational cost. Thus, the study was focused on the aerodynamic analysis (employing Edge CFD solver) which involves solution of non-linear equations with significantly larger number of unknowns than in case of the structural solver. In the Table 2.5 the considered settings of the CFD solver are listed. In general, well-converged CFD solution was required at the end of the coupled solution, thus the target residual reduction of the flow variables about 7 orders of magnitude was prescribed. Then, the influence of the flow residuals reduction in each coupling iteration on the coupled solution was observed in terms of computational cost and the resultant aerodynamic forces. The residual reduction was limited by prescribed total number of CFD iterations (referred as inner iterations) in each coupling iteration. The scheme designated as ”01” requires complete converged CFD solution before the data exchange on the aeroelastic interface. In other cases the CFD solution is limited by prescribed number of inner iterations. Therefore, partially converged flow solution can be expected mainly during first coupling iteration. After the coupled solution convergence criteria were satisfied, the reduction of the flow residuals was checked whether the prescribed value was reached. If not, the CFD solution was restarted to obtain the well-converged flow solution as it was required.

Scheme	01	02	03	04	05	06
Number of grids in multi-grid solution[a]	3	3	1	1	1	1
Max. number of full multi-grid cycles	500	500	-	-	-	-
Max. number of CFD iterations at each coupling iteration	5000	500	500	250	125	65
Order of residual decrease for converged solution	-7	-7	-7	-7	-7	-7

[a] if > 1 then full multi-grid solution was performed

Table 2.5: Flow solver settings

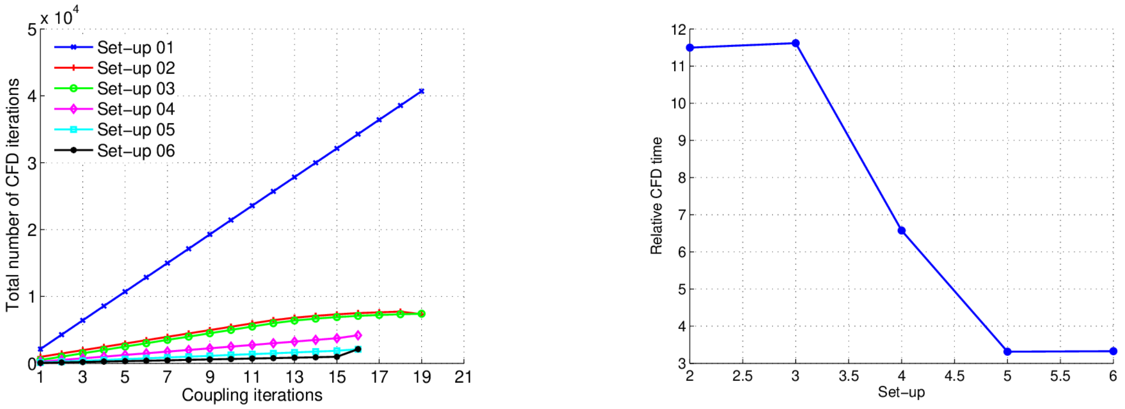


Figure 2.11: Computational cost of the aeroelastic solution - influence of the Edge CFD solver settings, Euler simulation at $M = 0.85$, $\alpha = 0°$

Scheme	01	02	03	04	05	06
C_L	0.273312	0.273304	0.273304	0.273306	0.273300	0.273281
C_D	0.004957	0.004957	0.004957	0.004956	0.004957	0.004956
C_m	-0.092672	-0.092664	-0.092664	-0.092660	-0.092662	-0.092657
Total CFD iterations[a]	40702	7338	7415	4196	2114	2124
Total coupling iterations	19	19	19	16	16	16

[a] in cases "01" and "02" the less expensive iteration on coarser CFD grids are included. The rigid CFD solution converged after 1249 iterations (including full multi-grid solution on coarser grids).

Table 2.6: Results of convergence study at $M = 0.85$ and $\alpha = 0°$

The reduction of the inner iterations number has consequence of the computational cost saving in terms of the total number of CFD iterations. (It should be noted that the really comparable are only cases from "03" to "06". The first and the second case employed full multi-grid solution using two coarser CFD grids, thus one iteration is cheaper on those grids than on the finest grid). The number of coupling iterations was reduced as well, this is beneficial from the perspective of time saving for the communication and the mesh deformation. In the this case, the lowest cost is suggested by the scheme "05" and is about three time higher than the cost of pure flow solution (see Figure 2.11). Figure 2.12 shows the convergence of the coupled solution. The plots on the right side of the figure show that the solution converged to the same result in terms of aerodynamic forces coefficients (the same shows the

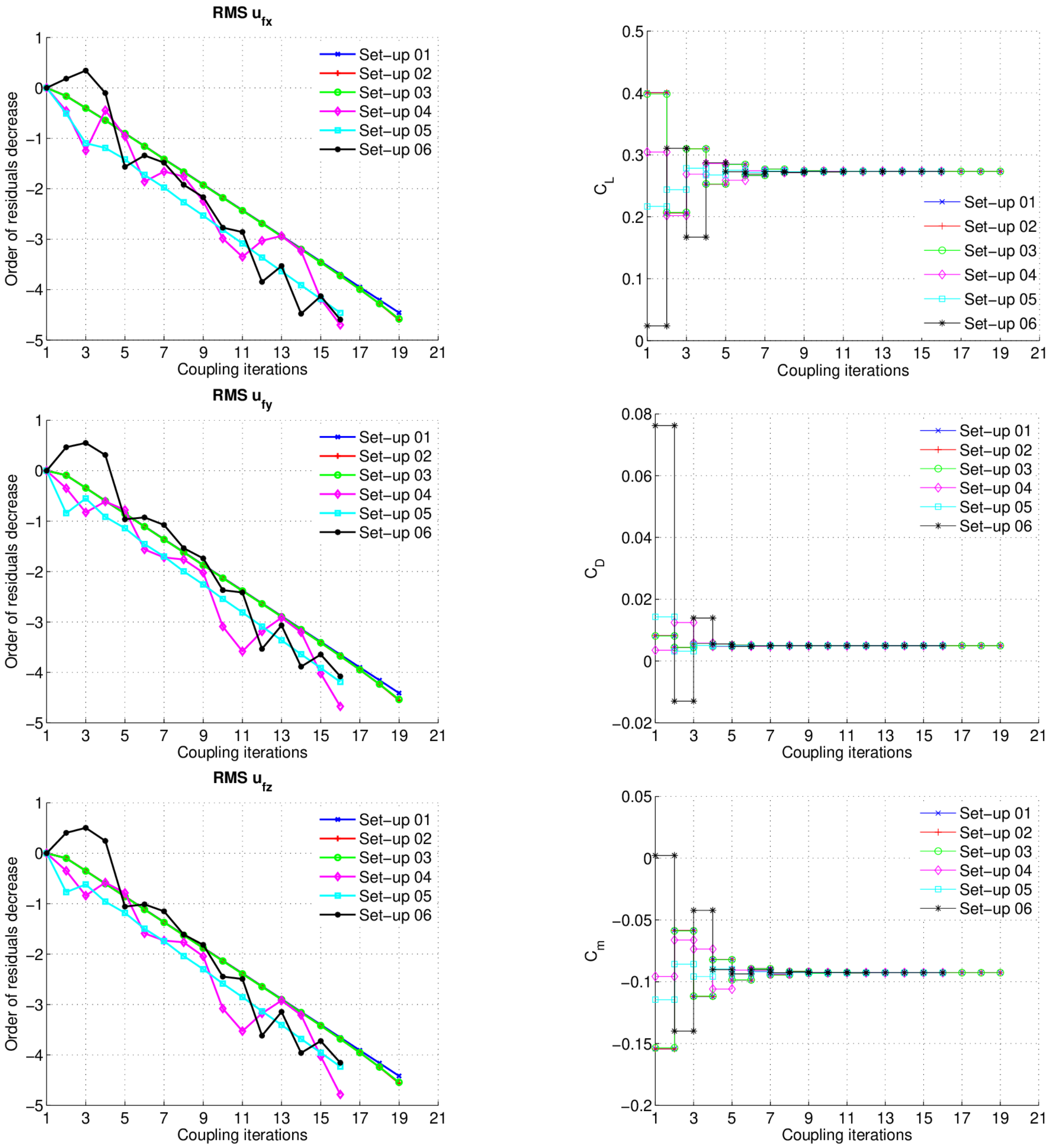


Figure 2.12: Convergence of the aeroelastic solution - influence of the Edge CFD solver settings, Euler simulation at $M = 0.85$, $\alpha = 0°$

Table 2.6).

The convergence was tested for other two flow conditions - at $M = 0.6$, $\alpha = 5°$ and at $M = 0.88$, $\alpha = 0°$. The results given in the Table 2.7 suggest that in this case the most efficient is scheme "06" from perspective of the needed CFD time. But the lowest number of the coupling iterations was achieved by scheme "03" (see the Table 2.7).

For the flow conditions at $M = 0.88$, $\alpha = 0°$, the same scheme was the most efficient from both perspectives (Table 2.8). The convergence plots for both flow conditions are given in the Appendix B.

Scheme	03	04	05	06
C_L	0.689944	0.689946	0.689879	0.689918
C_D	0.019790	0.019790	0.019787	0.019788
C_m	-0.119453	-0.119454	-0.119445	-0.119450
Total CFD iterations[a]	2653	1989	1813	1737
Total coupling iterations	7	8	9	19

[a] The rigid CFD solution converged after 1512 iterations (including full multi-grid solution on coarser grids).

Table 2.7: Results of convergence study at $M = 0.6$ and $\alpha = 5°$

Scheme	03	04	05	06
C_L	0.276955	0.276956	0.276956	0.276795
C_D	0.006963	0.006963	0.006963	0.006976
C_m	-0.099328	-0.099325	-0.099327	-0.099266
Total CFD iterations[a]	5411	7280	6227	5567
Total coupling iterations	11	26	37	45

[a] The rigid CFD solution converged after 1236 iterations (including full multi-grid solution on coarser grids).

Table 2.8: Results of convergence study at $M = 0.88$ and $\alpha = 0°$

2.7 Summary

Methods of definition of the fluid-structure interface, such as the nearest neighbour, weighted residuals, method of the finite interpolation elements and radial basis function, were presented in this chapter. The radial basis functions method has been chosen for the implementation in the computational aeroelasticity tool. The reasons for the choice of this method are accuracy, independence on the mesh connectivity and no need for computationally expensive search and projection algorithms.

The computational fluid dynamic method was applied for the aerodynamic predictions, because it can resolve non-linear features of the flow such as transonic shocks and the flow separation. Therefore, the tool is applicable for full range of a flight envelope of designed aircraft, which might improve the design efficiency.

The linear elastic solver was implemented in the Matlab environment. It is capable to predict deformation of the structure modelled by beam finite elements with direct input of beam stiffness parameters. The second option is to provide stiffness and mass matrices to the solver, thus matrices must be assembled using an external finite element preprocessor. Additionally, the influence of the inertial forces can be modelled by the solver. Inertial forces are calculated from the mass matrix according to Newton's second law of motion. The reason for the implementation of own structural solver was to overcome complicated communication with commercial solvers. Moreover, the solver might be implemented in the computational aeroelasticity tool using direct communication via the random-access memory, in potential further development.

The computational aeroelasticity tool was designed to employ arbitrary flow, structural and mesh deformation solvers. It is based on the simpler principle of communication using I/O operations via hard-copied files. The influence of various settings of the flow solver on the convergence of an aeroelastic solution was tested. The results suggest that optimal settings are case dependent.

Chapter 3

Equivalent Beam Model

3.1 Introduction

Simplified structural model still possesses its place in common aerospace engineering practice. It is widely used in both static and dynamic aeroelastic analyses and during the design of load control system [42]. It can be also applied in multidisciplinary design optimization aiming for estimation of favorable aerodynamic shape and structural characteristics combination during an aircraft conceptual design [53].

Different approaches for design of an equivalent structural model, either to real structure or to higher fidelity finite element, were studied. The Dunn [54] performed a study focused on matching of natural frequencies and mode shape given by ground vibration tests. He employed genetic algorithm in order to determine optimal wing stiffness and mass distribution of the model to match the experimental data. Algorithm updated physical parameters such as bending stiffness and mass. The problem of the solution uniqueness was addressed in the work. Resulting process was based on model complexity variation (number of parameters), and at the end, the model giving a good representation of experiment with minimum number of parameters was taken as unique solution. Similar approach was adopted by Trivailo et al. [55]. They studied different approaches, i.e. matching either dynamic or static response, or both of them. Genetic algorithm and artificial neural network methods were compared and uncertainty of solution was evaluated.

Other study focused on design of accurate beam finite element model is presented by Elsayed et al. [56]. In their work, method of stiffness estimation to match static deflection of more complex model (wing-box) was described and compared with other common approaches, such as analytical approach or empirical estimation of stiffness distribution. Presented method is based on sequential application of unit load on given segment of complex model. Subsequently, the resultant segment deformation gives the stiffness of segment.

Relatively lot of effort was given to the development of model simplification methods based on Guyan reduction [57]. The method works with stiffness and mass matrices directly - it does not give physical properties of reduced model. Thus, the model would not be applicable in the multidisciplinary design optimization.

In this chapter, method of equivalent beam derivation is presented. The goal is to create a simplified structural model giving similar static aeroelastic deformation as more complex structure. The model will be applied in the aerodynamic shape optimization of the elastic wing.

3.2 Inverse Design

The method purpose is to find simplified structural model, in this case a beam stick model, equivalent to higher fidelity model. Equivalence meant here is similarity in the static aeroelastic response of the coupled fluid-structure model. Therefore, deformation of given aerodynamic surface (e.g. a wing) under certain aerodynamic load must be, in the best case, same regardless of applied structural model.

The problem can be formulated as fitting of the beam model stiffness parameters in order to get known deformation.

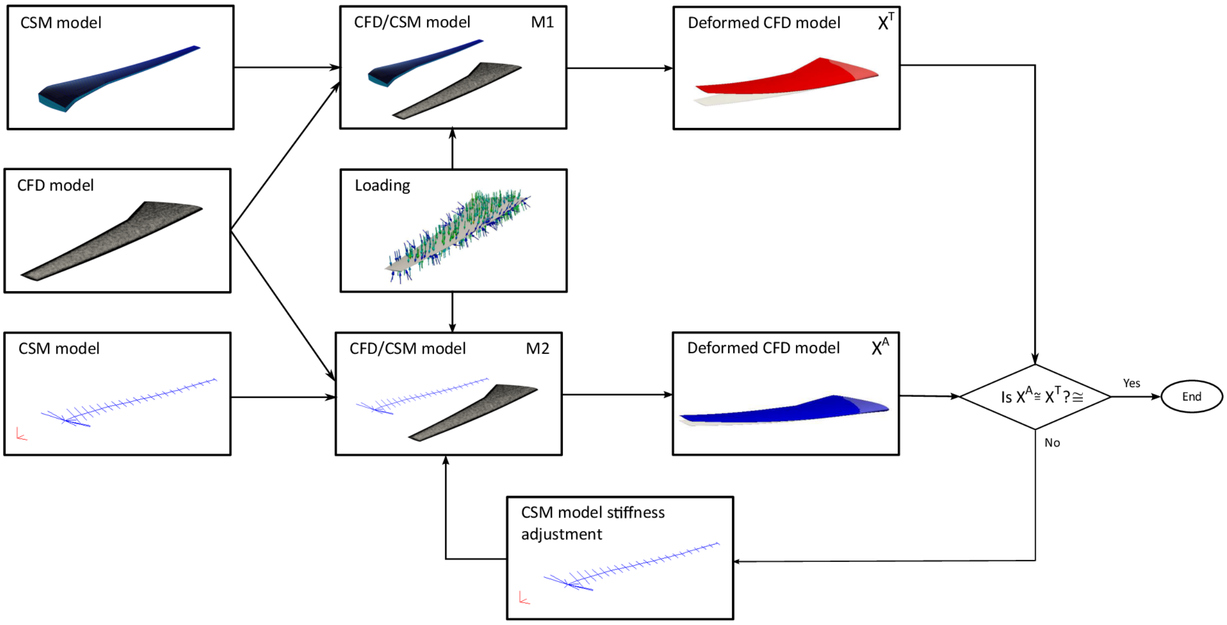


Figure 3.1: The equivalent beam derivation method

The inverse method is described in Figure 3.1. Initial step is a definition of aeroelastic models. The first model is a aerodynamic model coupled to a higher fidelity structural model M1 - a wing-box model or a full finite element model of the wing structure. The model M2 is the same aerodynamic model coupled to a simplified structural model - a beam stick model. The model geometry is defined at the beginning and remains unchanged during the process. Initial guess of structural stiffness distribution along beam must be provided (e.g. analytically obtained rough estimation or educated guess).

In the next step the same loading is defined for both aeroelastic models. The loading of the aeroelastic model M1 gives deformed wing shape, the target shape X^T. Finally, the fitting of aeroelastic model M2 begins by adjusting its stiffness until the loaded shape of the model X^A matches X^T. The adjustment is done through a gradient-based optimization method minimizing an objective function in form:

$$\min F = \frac{1}{N}\sum_{k=1}^{N}\sqrt{\sum_{j=1}^{3}\sum_{i=1}^{m}(X_{ij}^{A} - X_{ij}^{T})^2}, \tag{3.1}$$

where N is number of loading cases, m is total number of surface nodes in the aerodynamic model and $1 \leq j \leq 3$ represents x, y and z directions.

The NLPQLP package [58] was employed for the problem solution. The package was developed for the solution of the optimization problems with non-linear constraints. It is based on the sequential quadratic programming (SQP) algorithm, which is one of the most effective methods for such problems. More details can be found in the work of Boggs and Tolle [59]. The optimization algorithm belongs among the gradient-based methods searching for the local optima. Therefore, the global uniqueness of the result cannot be guaranteed. Nevertheless, desirable solution of the inverse method can be ensured by the proper definition of a feasible space and the initial guess. The feasible space was defined by constraint functions and bounds on variables.

3.3 Cases Descriptions

Models

Two types of coupled fluid-structure (aeroelastic) models were defined. In both models, the wing of the Common Research Model was used as an aerodynamic model (for more details about model origin see section 4.2.1). The employed Euler CFD grid is presented in the Figure 3.2.

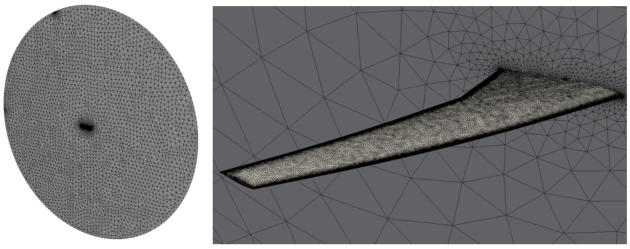


Figure 3.2: CFD model - overview (left) and detail (right)

The aeroelastic model employing the higher fidelity wing-box representation of the structure was considered as a target model. Specifically, the wing-box model labeled as v14 described in the section 4.2.2 was considered. In the other aeroelastic model, the structural model consisted of beam elements. Both structural models are shown in Figure 3.3.

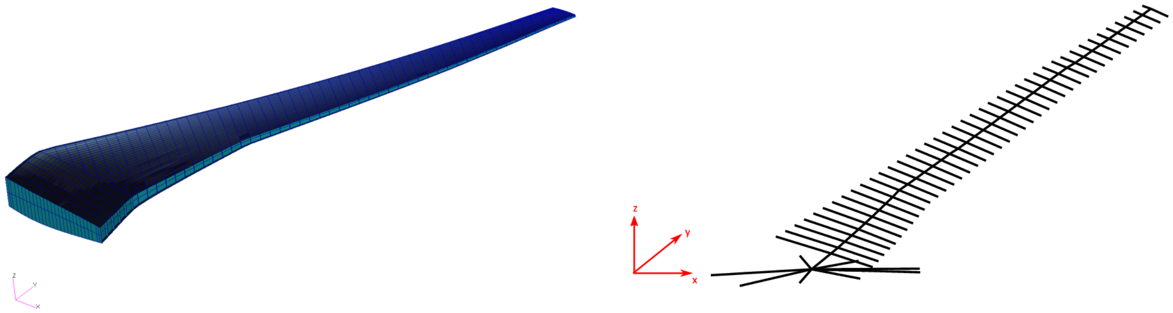


Figure 3.3: Wing box (left) and beam stick (right) finite element models

In all cases, the coupling between CFD and CSM model was done by Thin Plate Spline type of radial basis function, when the wing surface nodes were coupled to outside nodes of wing-box model or to all beam nodes.

Loading

The goal is, in both aeroelastic models, to obtain the same static wing deformation due to aerodynamic load, thus the aerodynamic forces acting on the rigid wing surface at given flight condition (Mach number, altitude, angle of attack, ...) might be chosen as the loading (labeled as Q_a). This loading results mostly in the displacement of the wing surface in vertical (z) direction due to bending, which is influenced by the bending stiffness parameter I_{z_b}. Thus, the loading might be insufficient to fit other parameters such as bending stiffness parameter I_{y_b} and torsional constant I_{x_b}. Therefore, other loading cases were considered:

- loading Q_z - positive unit forces in z direction in each wing surface node
- loading Q_x - positive unit forces in x direction in each wing surface node
- loading Q_t - positive unit forces in z direction in wing surface nodes ahead of considered elastic axis and negative unit forces in z direction in wing surface nodes aft of the elastic axis

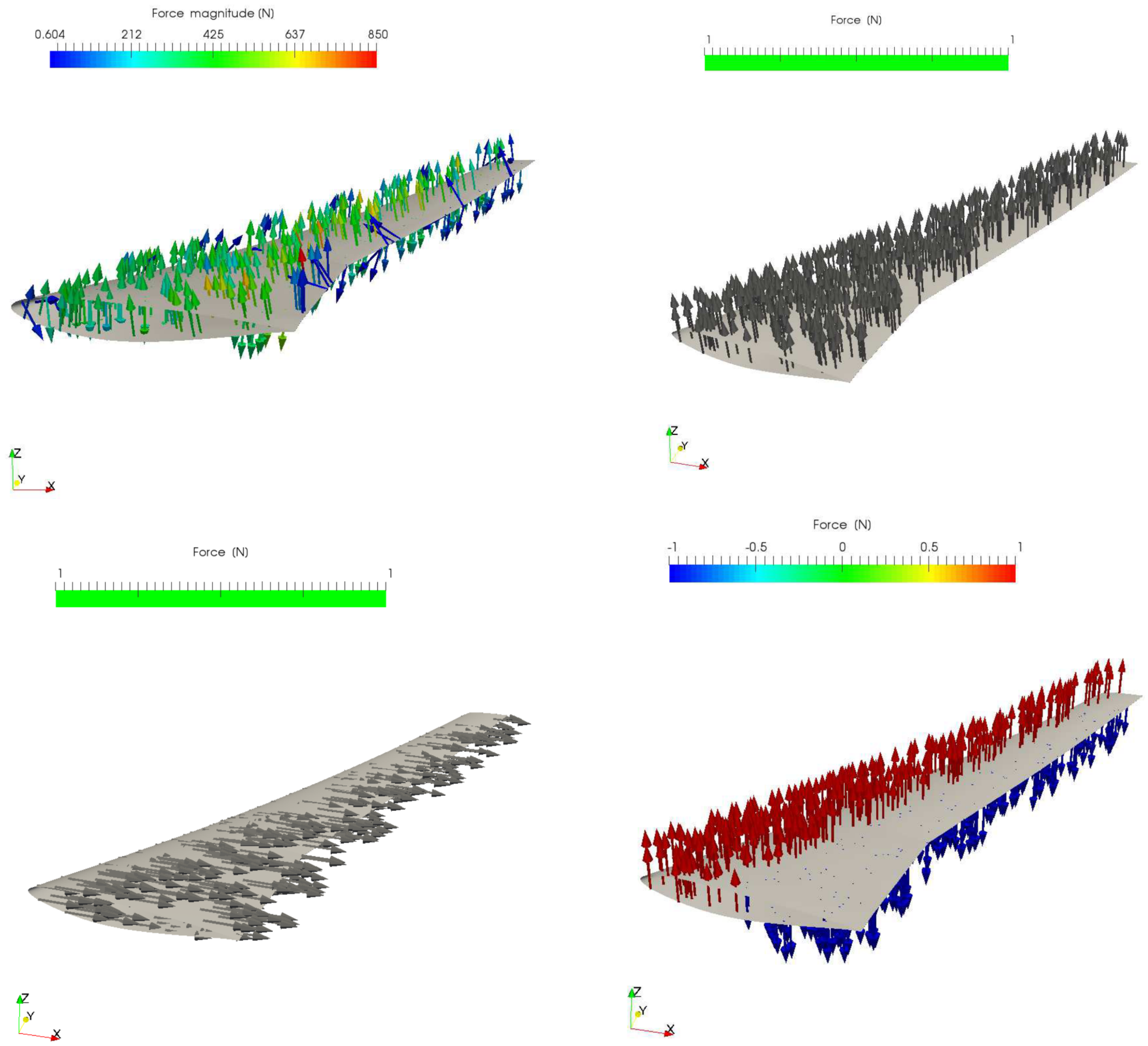


Figure 3.4: Considered loading cases - aerodynamic loading Q_a (upper left), unit forces loading Q_z (upper right), unit forces loading Q_x (lower left) and unit forces loading Q_t (lower right)

The loading cases are visualized in the Figure 3.4.

The considered coordinate system is shown in Figure 3.5.

Parameterization

The optimization based method requires suitable parameterization in order to find characteristics of equivalent structural model. Two types of parameterizations were tested. The first parameterization uses structural characteristics directly as parameters. Thus number of optimization variables grows with increasing number of beam elements. Physical feasibility of structural characteristic was imposed by bounds on parameters.

The second one uses constants of n-th degree polynomial as parameters, therefore number of variables depends on polynomial degree. Constraint functions and bounds on parameters were imposed in order to define space of feasible structural characteristic, which is close to reality such as:

1. negative gradients of polynomial functions - decreasing stiffness from root to tip
2. continuity of given characteristic at kink position

3. positive values of polynomial functions - strictly positive stiffness

Additionally, influence of beam stick model discretization was also tested. The test cases descriptions are given in Table 3.1. Analytically calculated stiffness distribution along beam was the initial guess for minimization problem. In cases C-H the linear polynomial constants were fitted to initial beam stiffness.

The beam stick model and the wing surface mesh were placed into coordinates xyz. Each beam element is placed into local coordinate system $x_b y_b z_b$, thus its cross-sectional characteristics are related to this coordinate system.

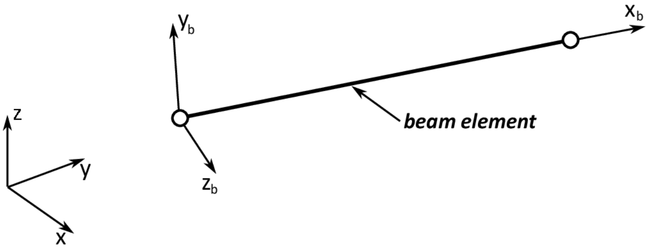


Figure 3.5: The coordinates system

Case	N_{elem}	N_{sec}[a]	N_{par}	Parameterized values [b]	Polynomial degree	Loading
A	21	-	84	A_b, I_{y_b}, I_{z_b}, I_{x_b}	-	Q_a
B	21	-	63	I_{y_b}, I_{z_b}, I_{x_b}	-	Q_a
C1	21	1	4	I_{z_b}, I_{x_b}	1st	Q_a
C2	21	1	4	I_{z_b}, I_{x_b}	1st	Q_z,Q_t
D1	21	2	8	I_{z_b}, I_{x_b}	1st	Q_a
D2	21	2	8	I_{z_b}, I_{x_b}	1st	Q_z,Q_t
E1	21	2	12	I_{y_b}, I_{z_b}, I_{x_b}	1st	Q_a
E2	21	2	12	I_{y_b}, I_{z_b}, I_{x_b}	1st	Q_x,Q_z,Q_t
F1	41	2	12	I_{y_b}, I_{z_b}, I_{x_b}	1st	Q_a
F2	41	2	12	I_{y_b}, I_{z_b}, I_{x_b}	1st	Q_x,Q_z,Q_t
G	21	2	18	I_{y_b}, I_{z_b}, I_{x_b}	2nd	Q_a
H	41	2	18	I_{y_b}, I_{z_b}, I_{x_b}	2nd	Q_a

[a] y is in interval for the case of - one section: $y_r \leq y \leq y_t$; two sections: $y_r \leq y \leq y_k$ and $y_k > y \leq y_t$, y_r, y_k and y_t is y-coordinate of the root, kink and tip section, respectively

[b] parameters are: cases A-B - structural characteristic; cases C-H - polynomial constants, index ^{k} is section number

Table 3.1: Cases descriptions

Stopping criteria for optimization algorithm are given in Table 3.2.

Stopping tolerance	No. of iterations	No. of function calls in line search
10^{-7}	1000	20

Table 3.2: Optimization stopping criteria

3.4 Results

3.4.1 Single loading cases

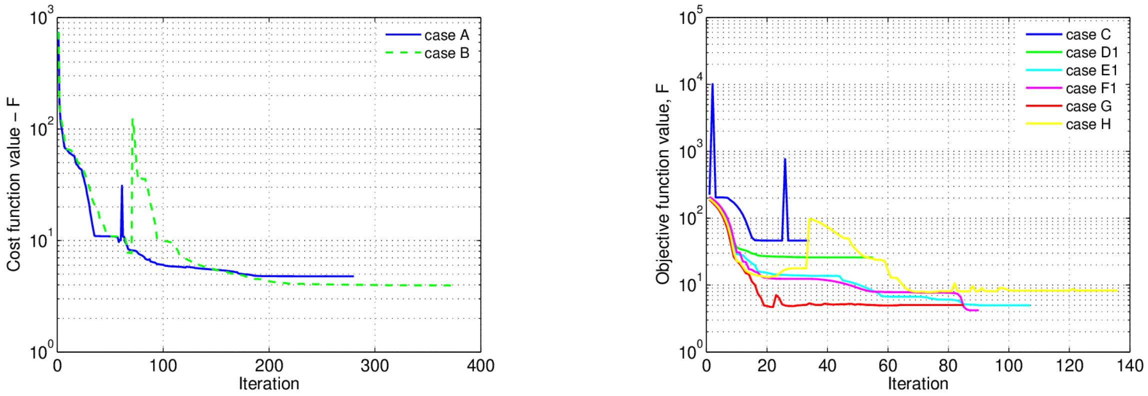


Figure 3.6: Solution history - direct parameterization cases (left), polynomial parameterization cases (right)

The solution history of the inverse design method is plotted in Figure 3.6. The Table 3.3 presents the objective function value at the end of minimization together with computational cost. The number of iterations, until the convergence is satisfied, is generally higher in cases of direct parameterization. The reason might be larger design space as the number of parameters is about 4 times higher. The inclusion of the bending stiffness in other plane has significant influence on the decrease of the objective function. The increase of the polynomial degree from the liner to the quadratic polynomial did not significantly improve the result but the cost of computation in terms of total objective function calls was roughly doubled. Finer beam model discretization resulted in minor improvement of the result in the cases of the linear polynomial parameters (cases E1 and F1), in the quadratic polynomial cases the result was worse.

Case	F_{min}	$N_{F_{eval}}$	$N_{gr_{eval}}$ [a]	Total $N_{F_{eval}}$ [b]
A	4.7711	869	280	24389
B	3.9447	1106	378	24920
C1	46.2494	135	34	271
D1	25.8930	122	55	562
E1	4.9786	263	107	1547
F1	4.1913	147	90	1227
G	5.0418	734	85	2264
H	8.2777	1003	136	3451

[a] gradients were calculated by finite differences, calculation cost in terms of CPU time is proportional to the number of optimization variables

[b] total number of objective function evaluation calls - objective values + gradients calculation

Table 3.3: Results of single loading cases - the objective function and the computational cost

Better insight on results is provided by plots of initial and resultant stiffness parameters given in Figures 3.7 and 3.8. The plots suggest that resultant bending stiffness about beam Z axis is relatively independent on the parameterization type and choice of additional parameters. In the cases A and B,

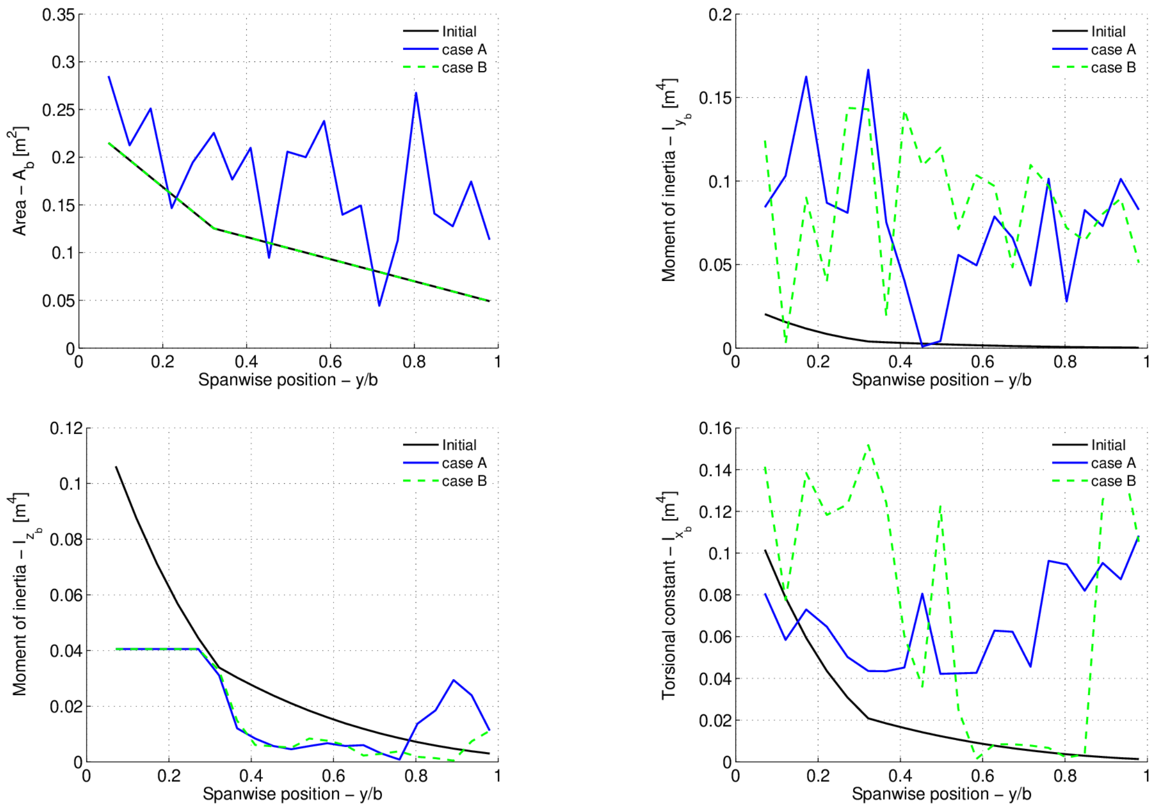


Figure 3.7: Structural characteristics of resultant beam stick models - direct parameterization cases

which employs direct parameterization, the resultant distribution of the bending stiffness parameter I_{z_b} is relatively smooth. The values of the parameter along span are comparable with the resultant values of the cases using polynomial parameterization. The values of the other parameters I_{y_b} and I_{x_b} are highly irregular (in cases A and B) and indeterminate, which is alos suggested by results of the cases C-H. The reason might be that the largest proportion of deformation is produced by the bending about beam Z axis. The torsion and other bending stiffnesses do not influence the deformation so significantly. Therefore, related parameters have less influence on the cost function minimization as it is illustrated in Figure 3.9. The gradients of cost function with respect to variables related to stiffness parameter I_{z_b} are about three order of magnitude larger than others. As results, less significant stiffness parameters are more scattered.

The Figure 3.10 compares the deformed elastic wing shapes as results of loading by given aerodynamic forces. The comparison is done for the wing-box structural model, initial beam stick model and beam stick models obtained by different setting of the inverse design method. The plots suggest that in all cases the beam model was updated in the way that after loading the wing deformation is more comparable to target wing deformation. The worst solution, among all cases, is given by Case C in which bending stiffness and torsional parameters, I_{z_b} and I_{x_b}, were parameterized along span by single linear polynomial for each parameter. The incorporation of abrupt stiffness change in the kink position produced improvement of results. Significant improvement of the results was achieved by inclusion of the bending stiffness parameter in other bending plane (cases E-G), as it is suggested by plots of the wing twist and span-wise sections.

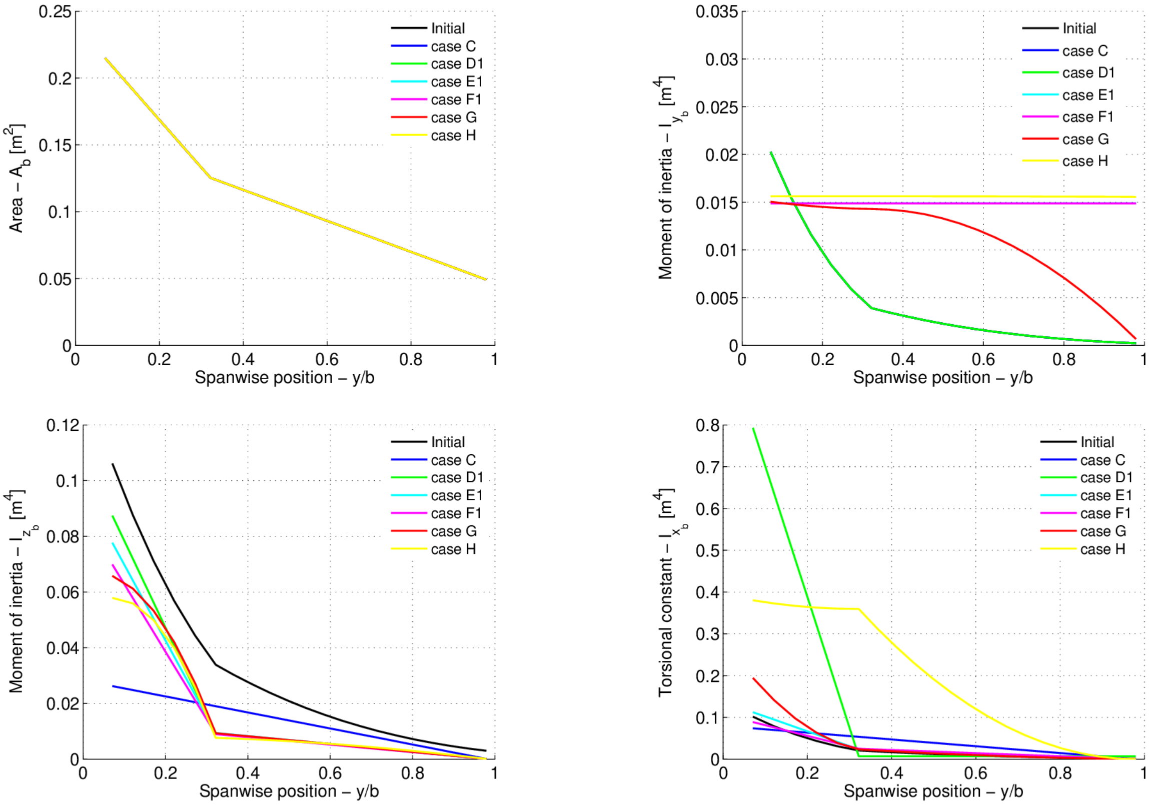


Figure 3.8: Structural characteristics of resultant beam stick models - single loading, polynomial parameterization cases

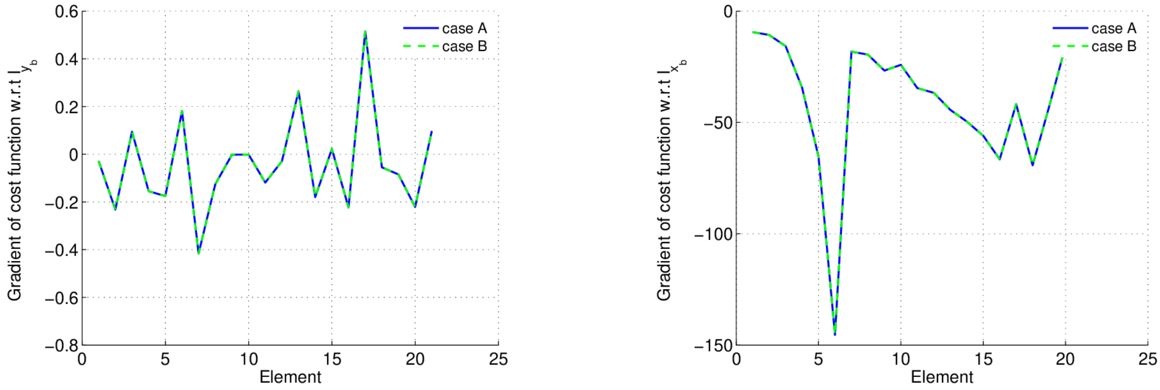


Figure 3.9: Gradients of cost function w.r.t. various design parameters at first iteration

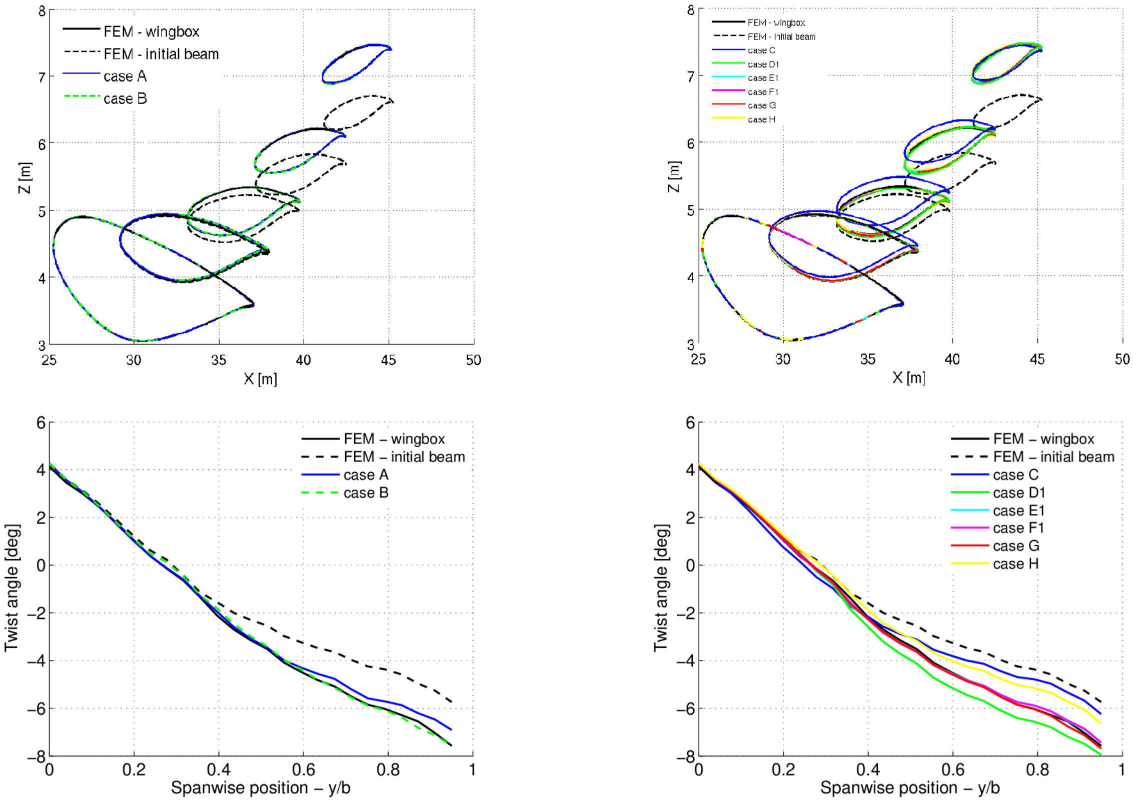


Figure 3.10: Comparison of target loaded wing shape with loaded wing shape using beam stick models - single loading direct (left) and polynomial (right) parameterization cases

3.4.2 Multiple loading cases

The inclusion of the three different load cases aims to solve the indeterminacy of the resultant values of the stiffness parameters I_{y_b} and I_{x_b}. Only the cases employing linear polynomial parameterization were considered. Direct parameterization and quadratic polynomial parameterization cases were not tested. The reason is that the first mentioned, besides their tremendous computational cost, provide essentially unrealistic results. The second parameterization cases did not provided any considerable improvement of the results and the cost was higher compared to cases employing linear polynomial parameterization.

Additionally, the effect of different initial guess of the design parameters was tested. The case F2 was taken as a reference. In that cases (labeled as F3 and F4), the initial stiffness parameters were constant along span and the values were equal to the ones of the initial beam at the kink position.

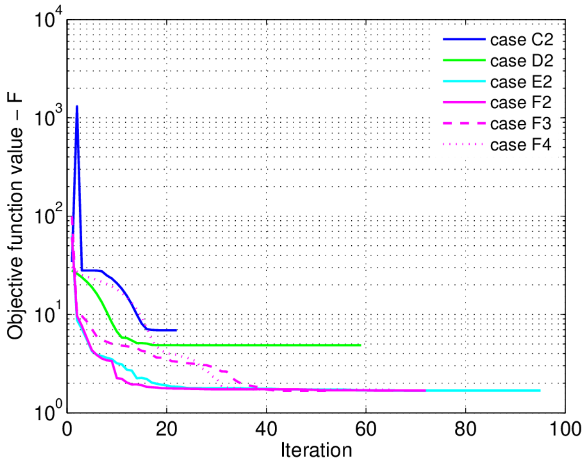


Figure 3.11: Solution history - multiple loading, polynomial parameterization cases

The solution history plotted in the Figure 3.11 suggests that the inclusion of the different load cases improved the results. Moreover, the objective function converged to lower values as the design space was loosened by incorporation of the abrupt change of the stiffness (at the wing kink position) and by consideration of the other bending stiffness parameter. The Table 3.4 shows the values of the objective function and the computational cost. The comparison of the cases with their counterparts, among the single loading cases, shows the overall improvement of the results for similar computational cost.

The resultant objective function values in the cases F3 and F4 with different initial guess are nearly the same as in the reference case F2. The computational cost is comparable as well.

The resultant design parameters values are plotted in Figure 3.12. The plots suggest that the final bending stiffness parameter I_{z_b} is independent on the setting of the inverse design method. Although the certain improvement is observable, the other two parameters are not still clearly defined.

The different initial guess of the design parameters led to same resulting values of I_{z_b} and I_{x_b} but results of I_{y_b} were slightly different.

The shapes of the loaded wing models, shown in Figure 3.13, are in good agreement with the reference wing shape in the cases E2 and F2. The different initial guess has negligible influence on the resulting shape which is indistinguishable from reference shape (case F2).

Case	F_{min}	$N_{F_{eval}}$	$N_{gr_{eval}}$ [a]	Total $N_{F_{eval}}$ [b]
C2	6.8967	31	22	119
D2	4.8647	80	59	552
E2	1.6803	189	95	1329
F2	1.6831	400	72	1264
F3	1.6729	176	53	812
F4	1.7228	182	65	962

[a] gradients were calculated by finite differences, calculation cost in terms of CPU time is proportional to number of optimization variables

[b] total number of objective function evaluation calls - objective values + gradients calculation

Table 3.4: Results of multiple loading cases - the objective function and the computational cost

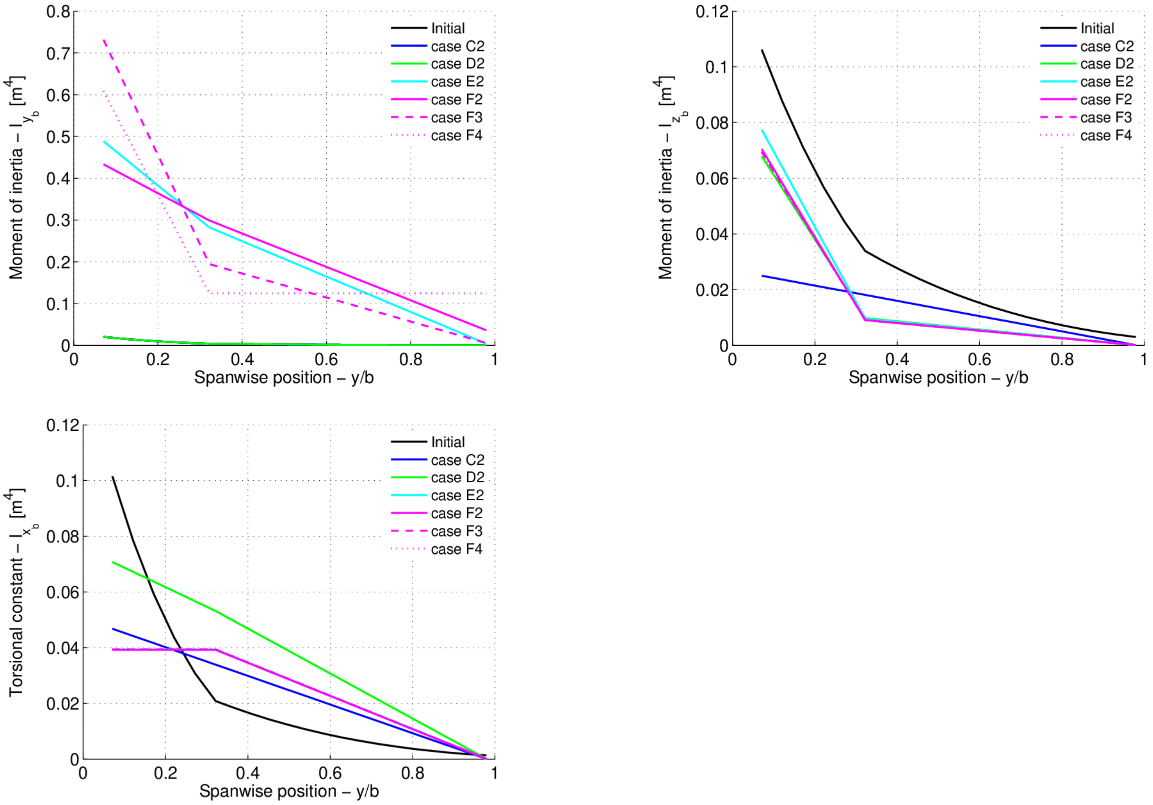


Figure 3.12: Structural characteristics of resultant beam stick models - multiple loading polynomial parameterization cases

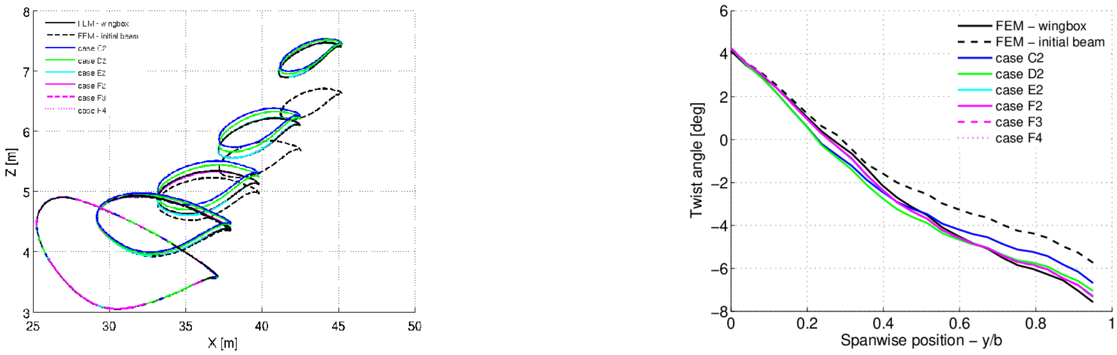


Figure 3.13: Comparison of target loaded wing with loaded wing using beam stick models - multiple loading polynomial parameterization cases

3.5 Summary

The method for design of simplified structural model was presented. The reference structural model was a wing-box model of a complex transonic wing with a kink and high sweep angle. The method is based on the minimization of an objective function which compares loaded wing shapes, the reference one with the wing shape of aeroelastic model employing current design of the beam structural model.

Different parameterization approaches were tested. The first one, used directly the stiffness parameters at each finite element as design parameters. Results show that the resulting distribution of the stiffness is irregular and does not agree with the distribution expected in reality. Moreover, the computational cost is tremendous. Therefore, the parameterization using polynomials in order to define the design values of the stiffness parameters was employed. The results suggest that linear polynomials might be sufficient to define the beam stiffness characteristics. The initial guess of the stiffness parameters was rough analytical estimation of the relevant characteristic.

Initial test employed single loading by aerodynamic forces but it was shown that it is not enough in order to clearly define all stiffness characteristics. Thus, other cases were tested employing three loads by unit forces in different directions. The results of the method were improved.

Further, the influence of different initial guess was tested. The case, which gave the best results in terms of the objective value, the computational cost and the resulting shape of the loaded wing, was taken as a reference. The results of all observed characteristics agreed with reference ones, except the parameter defining the wing in-plane bending stiffness.

The results suggest that the proposed inverse method might be applicable for design of simplified structural model of a complex wing geometry. There is still space for further development of the method, such as extension for design of beam stick model dynamically equivalent to higher fidelity structural model. In that case, an objective could be to minimize differences between natural frequencies of structures and mode shapes interpolated to a wing surface.

Chapter 4

Static Aeroelasticity: Validation of the Computational Aeroelasticity Tool

4.1 Introduction

The application of the Computational aeroelasticity tool is presented in the chapter. Several cases have been performed differing in applied structural and aerodynamic models. The main focus of the chapter is the validation of the tool functionality.

Two different structural models were applied - wing-box and beam stick models. The beam model is equivalent to the wing-box model in the sense of static deformation. The model was designed according to the inverse design method described in Chapter 3. The cases employing the beam stick model were focused on validation of the inverse design method. Other aim was the evaluation of the simplified structural model applicability for static aeroelastic simulation and aerodynamic design optimization of the flexible wing. In the cases employing the wing-box model, the airframe weight was considered. The comparison of the wing-box and beam structural cases was done without weight consideration.

The applied aerodynamic models are transonic wing-only and wing-fuselage geometries. In the first case the flexible wing is considered. The other case, using a rigid fuselage combined with a flexible wing, evaluates the tool ability to handle more complex conditions. The considered flow models were compressible inviscid and viscous solved by Euler and Reynold averaged Navier-Stokes (RANS) equations, respectively. In attached flow conditions, the less expensive Euler flow solution can give reliable estimation of the pressure distribution needed for the aeroelastic simulations. Therefore, the aeroelastic solutions employing the Euler flow and RANS flow were compared in order to evaluate the differences.

4.2 Test Cases

4.2.1 Geometry

The model of a common transonic transport aircraft, namely the NASA Common Research Model (CRM), has been applied in the tests. The model was originally intended for CFD validation studies [60], but it became standard model for other applications including aerodynamic shape optimization [61, 62], aero-structural optimization [63, 64, 65] and aeroelastic tailoring [66].

The model corresponds to the Boeing 777 airliner, the relevant specification of the CRM are listed in the Table 4.1. Since only symmetric flow conditions without a sideslip were considered, only half of airplane has been used in the test cases as it is shown in the model schematic drawing in Figure 4.1. The wing-only configuration was created by cut in the position about 3 meters from symmetry plane removing the fuselage. New symmetry plane was created in the wing root location.

Parameter	Value
Cruise Mach number	0.85
Cruise lift coefficient	0.5
Cruise altitude	11000 m
Wing span	59.1 m
Reference wing area	383.7 m^2
Reference wing area - wing-only geometry	332 m^2

Table 4.1: CRM specification

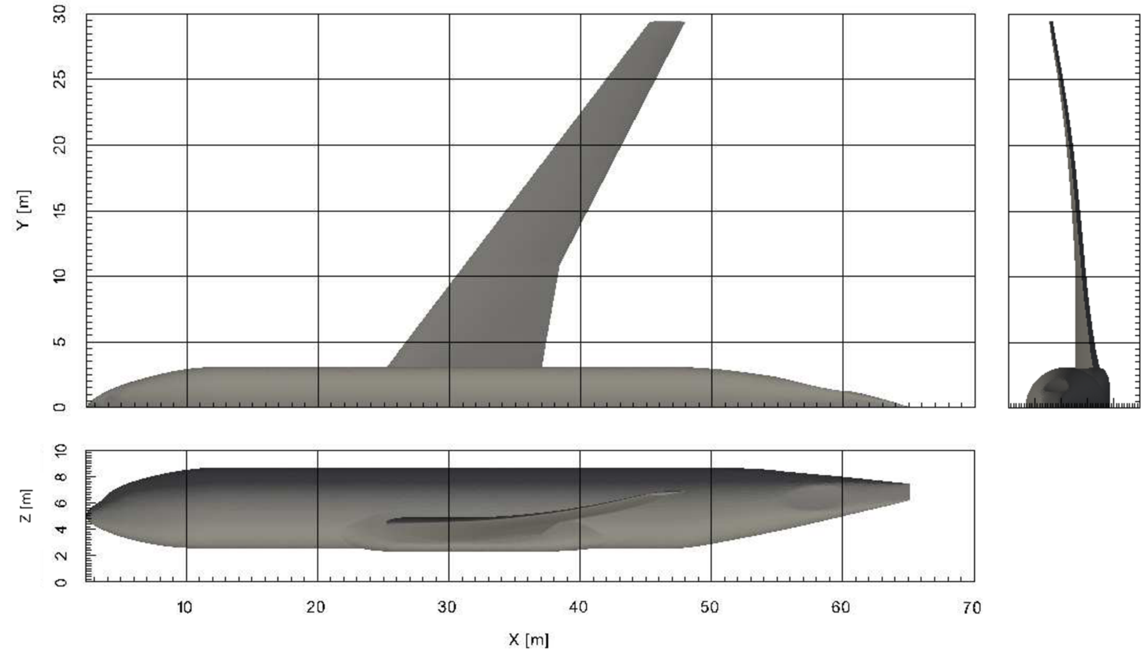


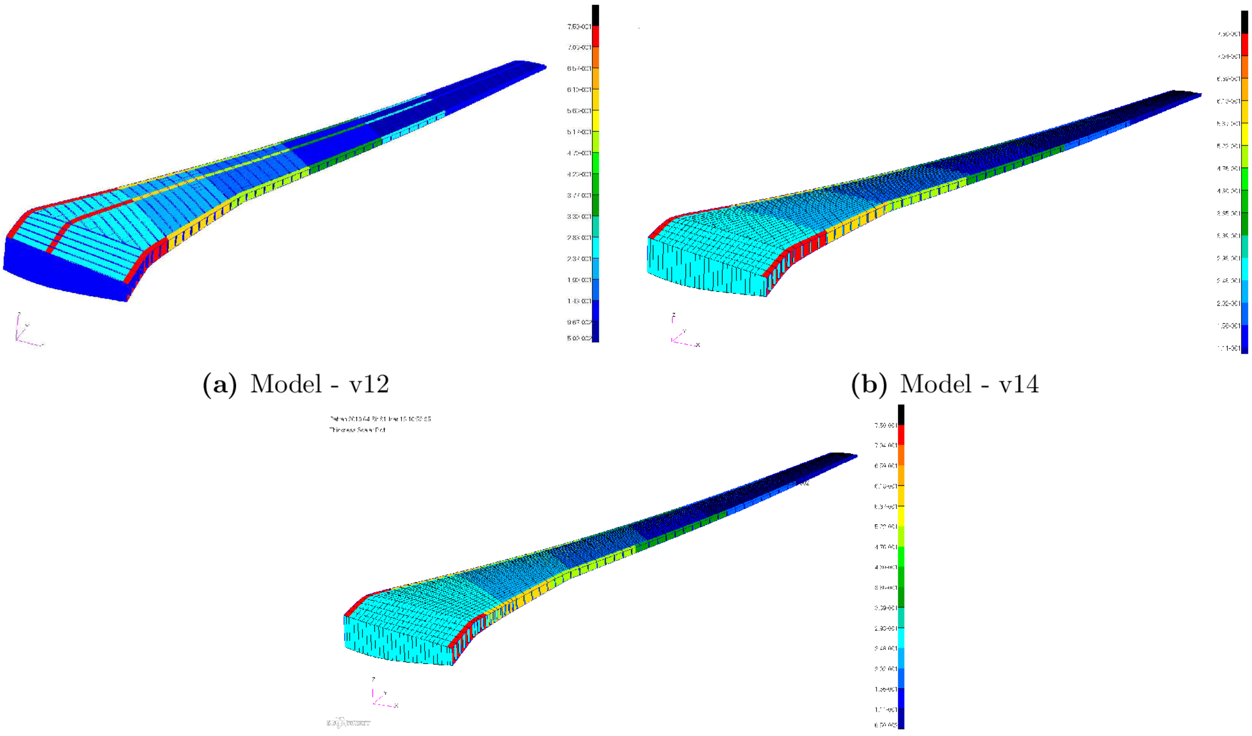
Figure 4.1: CRM wing-fuselage geometry

4.2.2 Structural models

Wing-box model

The finite element models of CRM wing structure are provided on the website [67]. The available structural models differ in the mesh density and element topology, see Figure 4.2. The highest fidelity model labeled as v12 is high detailed model of the wing structure consisting of explicit shell based geometry. It is three spars wing-box structure with multiple ribs and a skin reinforced by stringers. In other models called v14 and v15 the mid-chord spar was replaced by chord-spaced shear webs (in figurative sense, the mid-chord spar was eliminated in favor of multiple spars along chord wise direction). All the models are provided in coarse and refined versions. The models have been validated according the FAR25 regulations by static sizing check including gust (-2 to 3.75g), maneuver (2.5g) and taxi bump (-2.0g) loads and by flutter analysis for nominal cruise conditions.

The coarse model v14 was chosen for the evaluation of the computational aeroelasticity tool. It consists of 4622 nodes connected to 8502 quad or tri shell elements (see Figure 4.3). The wing main structure mass distribution was calculated from the finite element model. The masses of the engines, nacelles, control surfaces, flaps and the fuel were not taken into account.



(a) Model - v12

(b) Model - v14

(c) Model - v15

Figure 4.2: Available finite element models of CRM wing structure - colored by element thickness

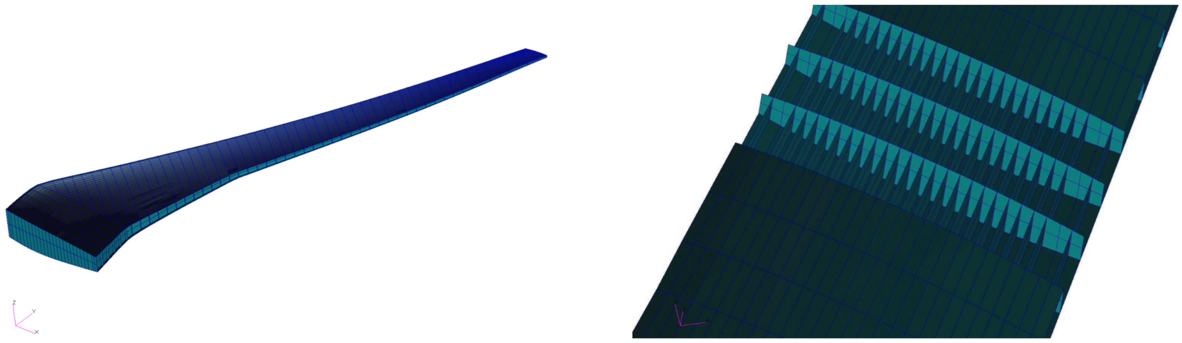


Figure 4.3: Wing-box finite element model of CRM wing structure applied in test cases, overview (left) and detail with partly hidden skin (right)

Beam stick model

A beam stick model of wing structure was designed according to method presented in the Chapter 3. The model is equivalent to the wing-box model meaning the static aeroelastic deformation of the wing is comparable in given operating conditions. The wing structure weight is neglected.

The beam stick model (Figure 4.4) consists of 22 nodes connected together by beam elements. Each node of the beam element is connected to two additional nodes by rigid elements. Those nodes are beneficial for coupling with aerodynamic surface, meaning the nodes allow reconstruction of rotations only by translational degrees of freedom. In the wing root, more additional rigid elements are required due to applied fluid-structure coupling interface, which is defined by radial basis functions method.

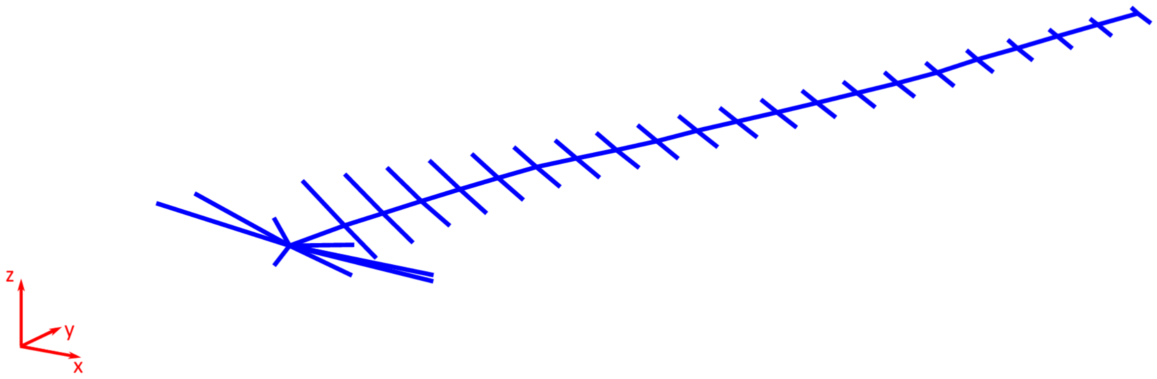


Figure 4.4: Beam stick finite element model of CRM wing structure applied in test cases

4.2.3 Aerodynamic model

CFD computational grid was created from a geometry provided on CRM website which is a wing-fuselage configuration of a common transport aircraft. The model was originally designed for evaluation of computational tools abilities in prediction of a drag, thus the shape of the wing is twisted and bended as it would be during the design cruise conditions. It was assumed the deformed shape of the wing is not in conflict with intended use of the model for verification of the aeroelastic tool.

The wing-only unstructured Euler mesh consists of 842837 nodes and about 4.5 millions tetrahedral elements, whereas hybrid unstructured RANS grid consist of 2644786 nodes and about 7.6 millions tetrahedral and prismatic elements. In case of the wing-fuselage geometry, the Euler and RANS meshes consist of 865982 nodes (4.7 millions elements) and 2831524 nodes (8.5 millions elements), respectively.

4.2.4 Aeroelastic interface

The fluid-structure interface was defined using radial basis function (RBF), particularly by Thin Plate Spline (TPS) function. The TPS is one of the most robust, cost effective and accurate RBFs for the fluid structure interaction (see section 2.2.2). The interface was defined between the wing surface nodes in aerodynamic grid and either the surface nodes of the wing-box structural model or all nodes in the beam finite element model.

4.2.5 Mesh deformation

The method applied for deformation of aerodynamic computational grid in test of the aeroelastic tool was RBF mesh deformation which is part of the CFD code Edge. The advantage of this method is ability to handle relatively large boundary deformations with low computational cost, but the surface deformation are not recovered exactly. However, the proper settings of the tool can ensure that the error in the surface shape is low (Figure 4.6). Thus, the influence of the deformation error on aeroelastic results was assumed negligible.

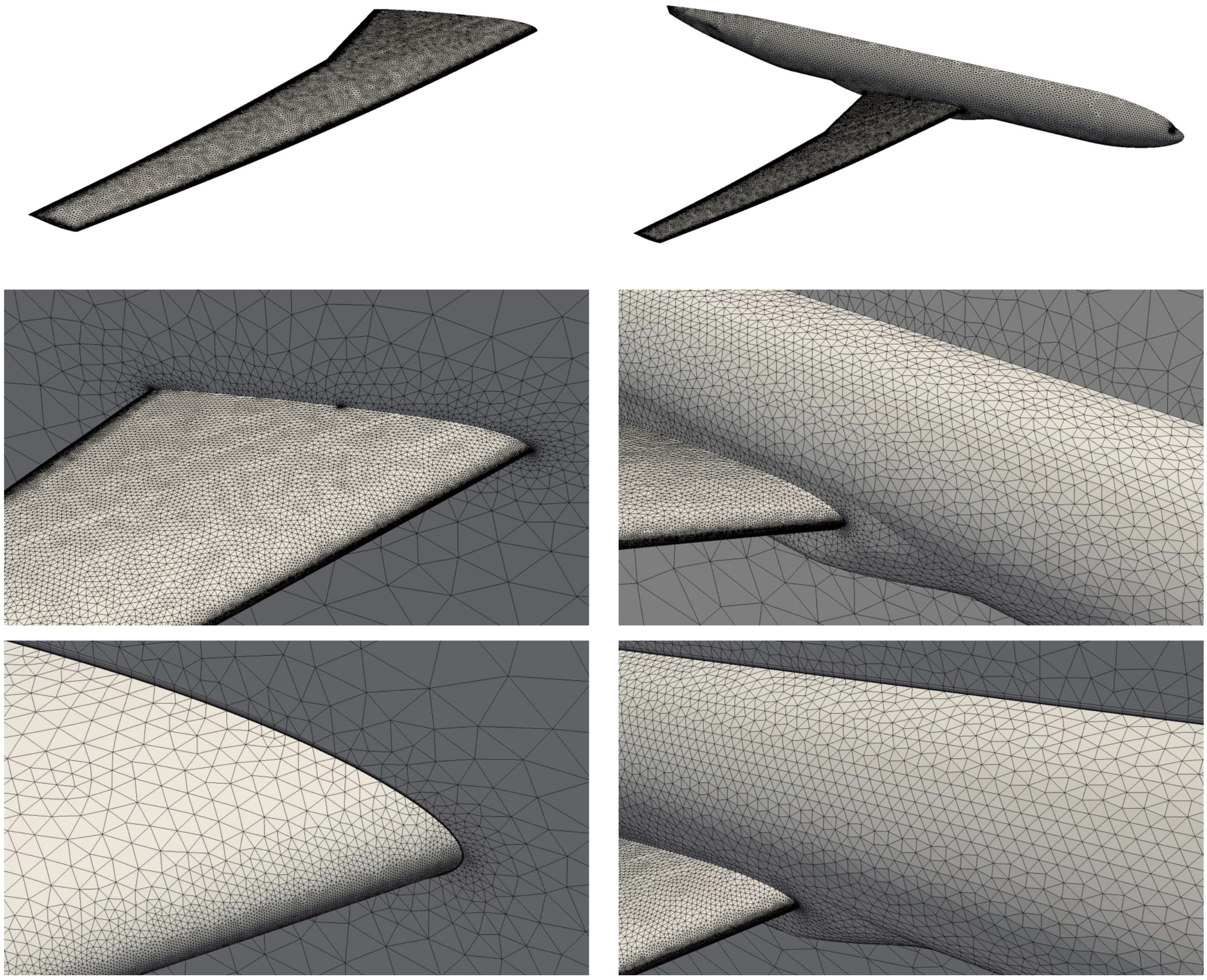


Figure 4.5: Surface aerodynamic meshes of wing-only and wing-fuselage geometries; Detailed views of Euler meshes (middle row) and RANS meshes (bottom row)

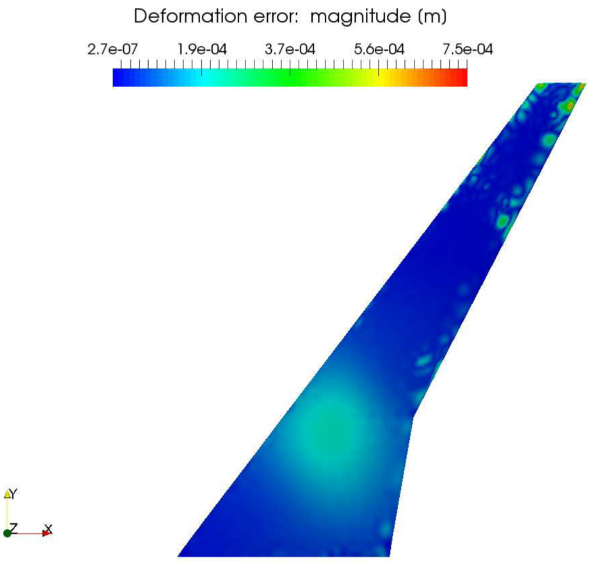


Figure 4.6: Error in deformed boundary mesh

4.2.6 Flow conditions

The transonic flow condition listed in Table 4.2 were considered in majority of the test cases. The corresponding Reynolds number in case of RANS flow simulation is approximately Re = 5184000 m^{-1}.

Parameter	Value
Cruise Mach number	0.85
Cruise altitude	11000 m
Static pressure	22632 Pa
Temperature	216.65 K
Dynamic pressure	11450 Pa
Air density	0.364 kg.m^{-3}

Table 4.2: Flow conditions

4.2.7 Test cases summary

The summary of the considered test cases is given in the Table 4.3. In the cases A and B, the lift coefficient required for the steady level flight at given operating conditions was prescribed, thus the angle of attack was depended on the particular flow simulation (Euler or RANS) and it is not given in the table. In the same cases, the weight of the wing main structure was included in the static aeroelastic simulations. The masses of other components, such as engine, nacelles, control surfaces, flaps and fuel, were neglected, although it is obvious they might significantly influence the wing deformation and the aerodynamic load distribution at the static aeroelastic equilibrium. It was assumed that it is sufficient to include the wing primary structure masses in order to evaluate a capability of the static aeroelastic simulation tool to handle the weight loads. The inclusion of other masses is just a matter of a finite element model preparation, as long as other masses are modeled using finite elements or point masses, the tool is able to handle them.

Case	Geometry	Structural model	Airframe weight	M	C_L	α
A	wing	wing-box	yes	0.85	0.5	-
B1a	wing	wing-box	no	0.85	0.5	-
B1b	wing	beam stick	no	0.85	0.5	-
B2a	wing	wing-box	no	0.6	-	5
B2b	wing	beam stick	no	0.6	-	5
C	wing-fuselage	wing-box	yes	0.85	0.5	-

Table 4.3: Test cases summary

The intention of cases B was an evaluation of the tool capability to apply a beam finite element model in the aeroelastic simulation and compare the result with cases using the wing-box model. Since the applied beam model is equivalent to the wing-box model by the stiffness not the masses distribution, the weight was not included in the simulation.

In all cases, the both Euler and RANS flow simulations were considered and their influence on the static aeroelastic solution was evaluated.

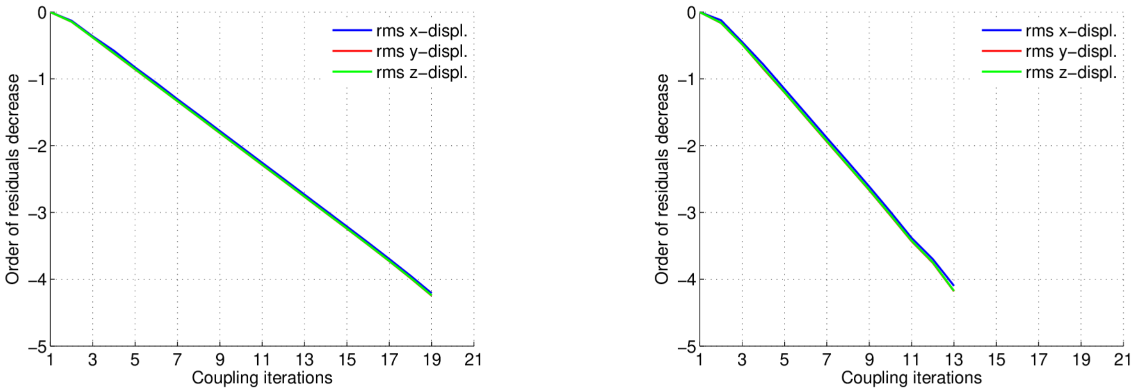


Figure 4.7: Convergence of aeroelastic solution using Euler (left) and RANS (right) flow simulation

4.3 Results

4.3.1 Case A: Wing-only geometry, wing-box structural model, $M = 0.85$

The convergence of the aeroelastic solution with respect to the coupling iterations is plotted in the Figure 4.7. In both Euler and RANS cases, the calculation was stopped when the decrease of the residuals felt bellow the tolerance $\epsilon = 1.10^{-4}$, what was considered as reasonable value. The Euler aeroelastic simulation required 19 coupling iterations, while the RANS simulation converged after 13 coupling iterations. The aerodynamic forces convergence is shown in the Figure 4.8.

The computational cost in terms of CFD iterations is presented in the Table 4.4. The Euler aeroelastic solution required approximately four times more iterations compared to Euler flow solution. The cost of aeroelastic RANS solution is comparable with the RANS flow solution. In both aeroelastic simulation cases the scheme 02 was applied. The convergence evaluation given in the Section 2.6 have shown that it might not be the most effective scheme, thus the computational cost in the Euler aeroelastic simulation could be comparable with pure flow simulation.

	Euler		RANS	
	Rigid	Elastic	Rigid	Elastic
No. of coupling iterations	-	19	-	13
Total no. of CFD iterations	1790	8086	1495	1725
α [deg]	0.72	2.15	1.85	2.91
C_L	0.5003	0.5007	0.4990	0.4980
C_D	0.0122	0.0125	0.0177	0.0189
C_m^a	-0.1754	-0.1170	-0.1128	-0.0683

[a] pitch moment is related to the quarter point of the wing mean aerodynamic chord

Table 4.4: Computational cost and resultant aerodynamic forces coefficients - wing-only geometry, $M = 0.85$, required $C_L = 0.5$

The values of the wing forces coefficient are presented in the Table 4.4. The angle of attack required to achieve lift coefficient needed for steady horizontal flight in cruise conditions is higher in the elastic cases than in the rigid cases. This is related to the wing deformation due to aerodynamic loading. The wing deformation caused the wing drag increase and the negative pitch moment decrease. The pitch moment of the elastic wing is nearly 2/3 compared to the rigid wing in both Euler and RANS simulation cases. Comparing the Euler and RANS aeroelastic simulations, the required angle of attack is higher in the RANS case. This is caused by the influence of the diffusive phenomena in the viscous flow.

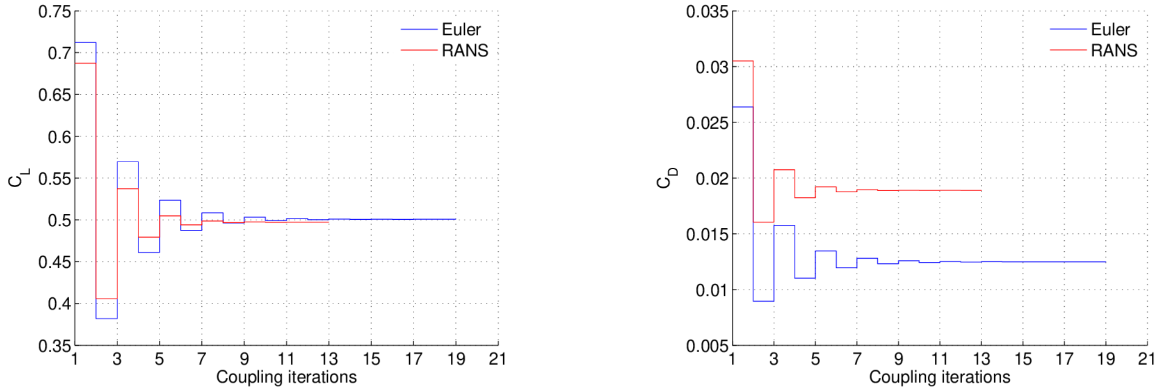


Figure 4.8: Convergence of elastic solution - aerodynamic forces coefficients

The Figure 4.9 presents the pressure contours over the upper wing surface. The Figures 4.10 gives more clear presentation of the pressure coefficient distributions at chosen wing sections. Comparing the RANS aeroelastic simulation to the Euler one, the suction in the inboard part of the wing is higher resulting in the stronger shock wave, which is shifted in the fore direction. In the outboard part, the suction is higher in the Euler simulation case. The mentioned differences in the pressure distribution arise mainly from different angle of attack of the wing combined with negative twist of the wing due to aerodynamic loading. The wing twist is shown in the left part of the Figure 4.11. The right part of the figure shows aerodynamic load distribution over the wing. In both Euler and RANS cases, the loading is higher in the inboard part and lower in the outboard part compared to the load distribution over the rigid wing. The figure shows also the influence of the wing weight on the wing twist and the aerodynamic load distribution. The loading was decreased in the inboard part and slightly increased in outboard part in both simulation cases. The aerodynamic loading, with gravity forces influence, causes the wing deformation presented in the Figure 4.12. The maximum wing tip displacement is 1.54 meters and 1.35 meters in Euler and RANS cases, respectively.

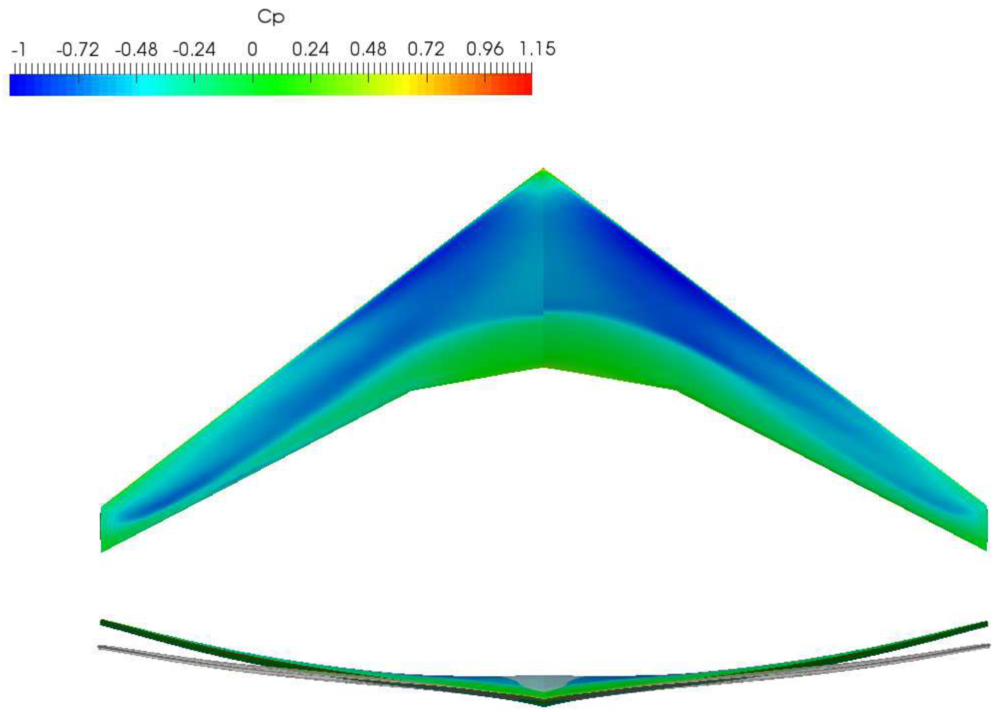


Figure 4.9: Comparison of Euler (left half) and RANS (right half) aeroelastic solution - surface pressure coefficient distribution at static aeroelastic equilibrium state

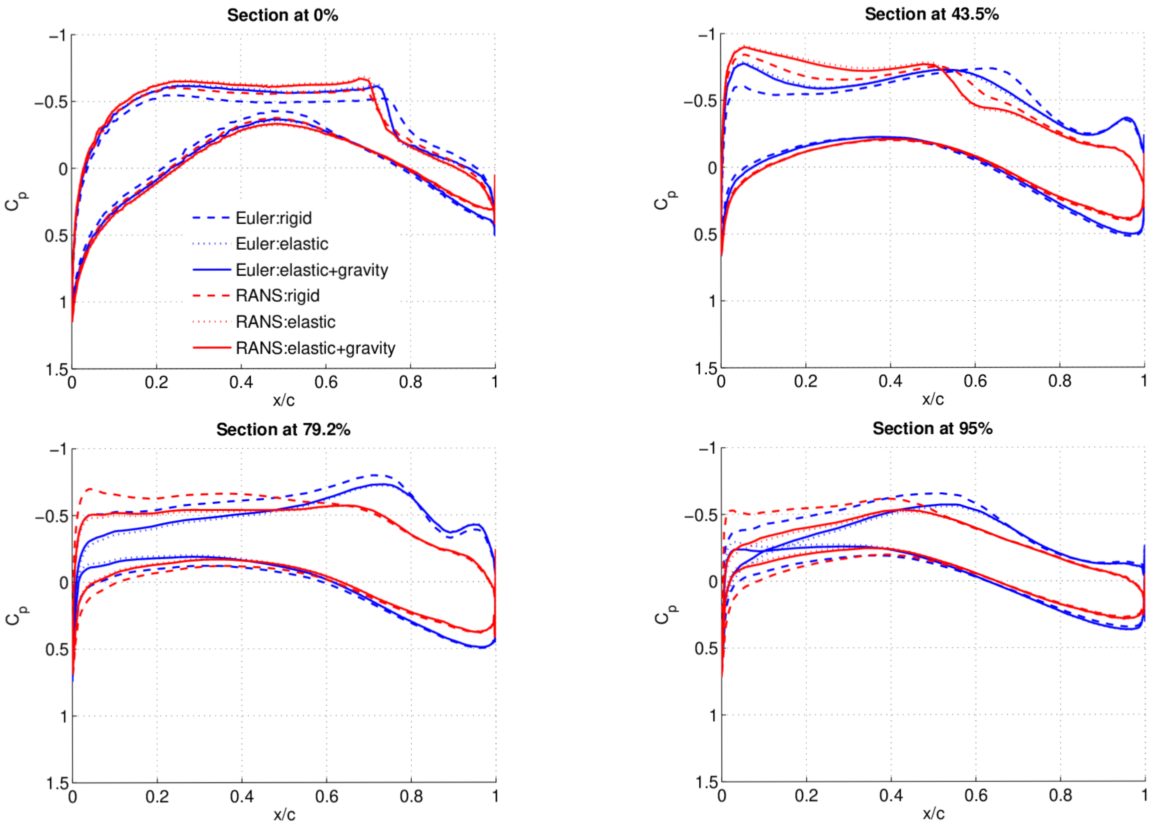


Figure 4.10: Comparison of surface pressure coefficient distribution at chosen sections of CRM wing

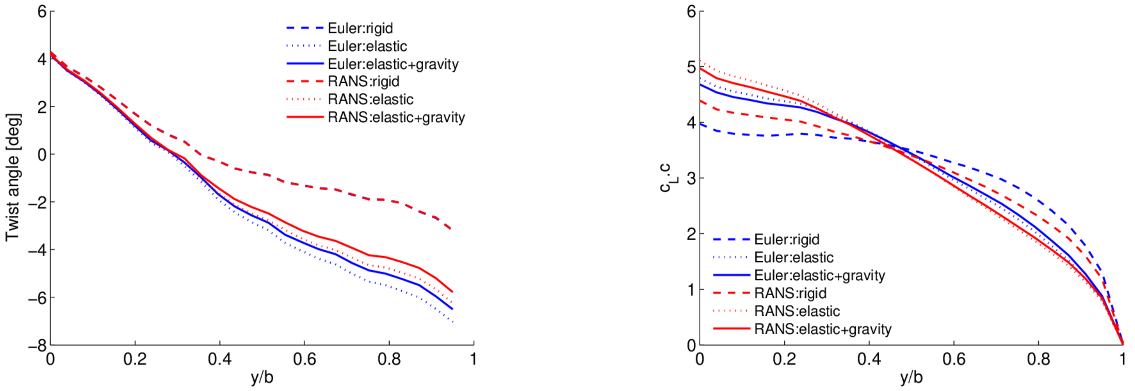


Figure 4.11: Comparison of rigid and aeroelastic solutions - the wing twist angle and the wing loading distribution

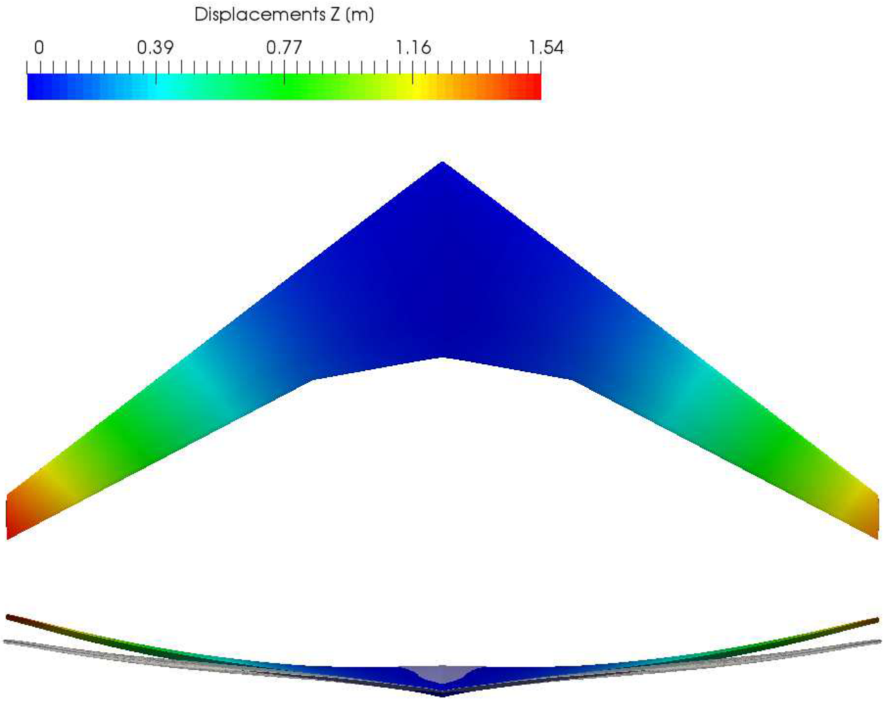


Figure 4.12: Comparison of Euler (left half) and RANS (right half) aeroelastic solution - the wing deformation at static aeroelastic equilibrium state

4.3.2 Cases B: Wing-only geometry, beam stick and wing-box structural models

The intention of the test cases was to validate an ability of the computational aeroelastic tool to handle a simplified structural model. The beam stick model was designed according to inverse design procedure described in the Chapter 3. Evaluation was done for both Euler and RANS simulations at two free stream conditions: $M = 0.85$, $C_L = 0.5$ and $M = 0.6$, $\alpha = 5°$. The results of the static aeroelastic calculation were compared with the case using the wing-box model.

	Euler			RANS		
	Wing-box	Beam stick	Difference	Wing-box	Beam stick	Difference
No. of coupling iter.	19	20	-	15	16	-
No. of CFD iter.	8183	8875	-	4931	4830	-
C_L	0.4996	0.4929	1.3%	0.5008	0.4898	2.2%
C_D	0.0126	0.0126	0.0%	0.0179	0.0175	2.2%
C_m^a	-0.1080	-0.1125	4.2%	-0.0614	-0.0615	0.2%

[a] pitch moment is related to the quarter point of the wing mean aerodynamic chord

Table 4.5: Comparison of aeroelastic simulation results using wing-box and beam stick structural models - Euler and RANS flow at $M = 0.85$ and required $C_L = 0.5$

The computational cost and the resultant values of aerodynamic forces are given in the Tables 4.5 and 4.6. The results, in terms of the aerodynamic forces, of the static aeroelastic computation using the simplified structural model are in good agreement with results given by the simulation applying the wing-box model. The largest difference is in pitching moment given by the aeroelastic simulation using Euler flow solution at $M = 0.85$. In that case, the negative moment is larger about 4% in case of applied beam stick structural model. Other forces differences are up to 2.5%.

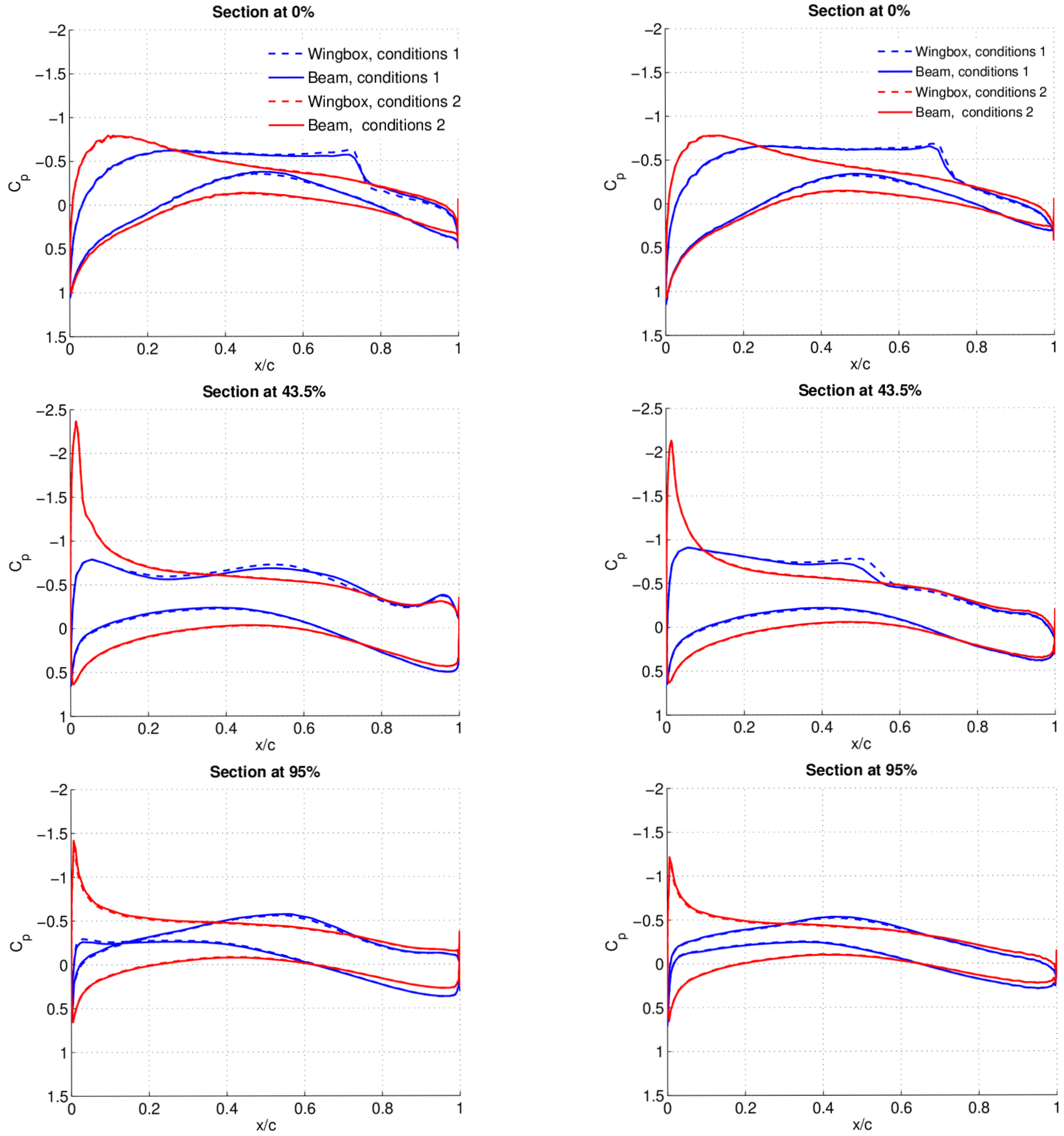
	Euler			RANS		
	Wing-box	Beam stick	Difference	Wing-box	Beam stick	Difference
No. of coupling iter.	7	8	-	7	7	-
No. of CFD iter.	3191	2802	-	1631	1220	-
C_L	0.6721	0.6687	0.5%	0.5972	0.5936	0.6%
C_D	0.0188	0.0188	0.0%	0.0214	0.0214	0.0%
C_m^a	-0.1115	-0.1122	0.6%	-0.0722	-0.0725	0.4%

[a] pitch moment is related to the quarter point of the wing mean aerodynamic chord

Table 4.6: Comparison of aeroelastic simulation results using wing-box and beam stick structural models - Euler and RANS flow at $M = 0.6$ and $\alpha = 5°$

The Figure 4.13 shows plots of the chord wise pressure distribution at selected wing sections. The pressure distribution given by the aeroelastic simulation using the beam structural model agrees well with the reference results in all cases. The pressure coefficient contours on the upper wing surface for the Euler flow simulation case at $M = 0.85$ are presented in the Figure 4.14. The figure shows good agreement with the reference simulation.

The plot of maximum span wise wing thickness distribution in Figure 4.15 illustrates that application of essentially two dimensional structural model does not produce any unrealistic geometrical changes of the deformed wing. However, this kind of structural model in combination with RBF transformation method requires use of additional rigid elements in the location of the wing root, as it is described in the section 4.2.2. If additional elements are not applied, significant change in wing thickness will occur as the result of the non-unique solution of the aero-structural coupling matrix.


Section at 0%

Section at 0%

Section at 43.5%

Section at 43.5%

Section at 95%

Section at 95%

Figure 4.13: Comparison of aeroelastic simulation results using wing-box and beam stick structural models - pressure distribution at different sections of CRM wing, Euler (left) and RANS (right) simulation at $M = 0.85$, $C_L = 0.5$ (condition 1) and $M = 0.6$, $\alpha = 5°$ (condition 2)

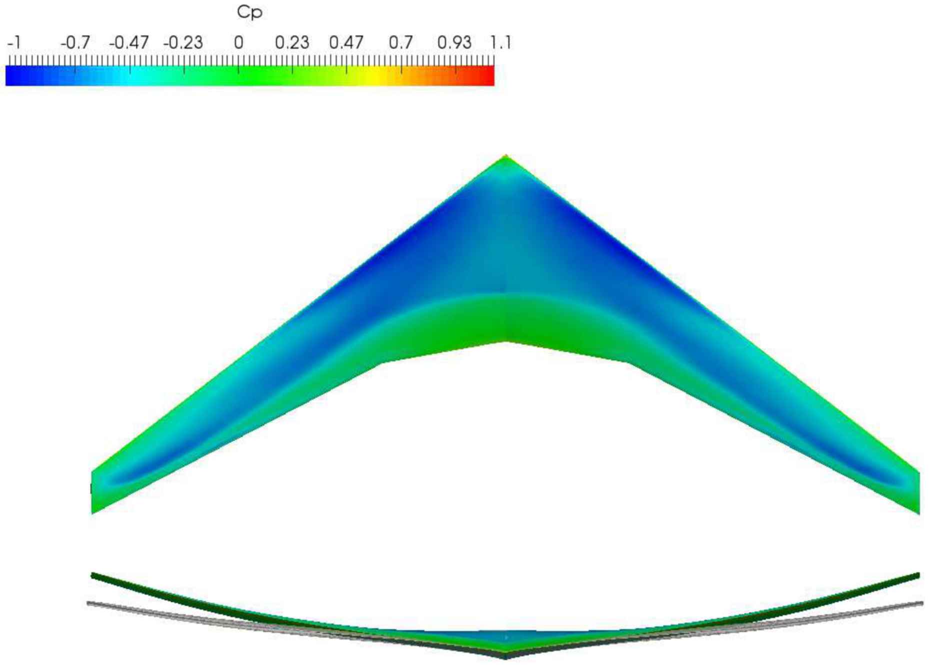


Figure 4.14: Comparison of aeroelastic simulation results using wing-box (left half) and beam stick (right half) structural model - surface pressure distribution; Euler flow simulation at $M = 0.85$, $C_L = 0.5$ (condition 1)

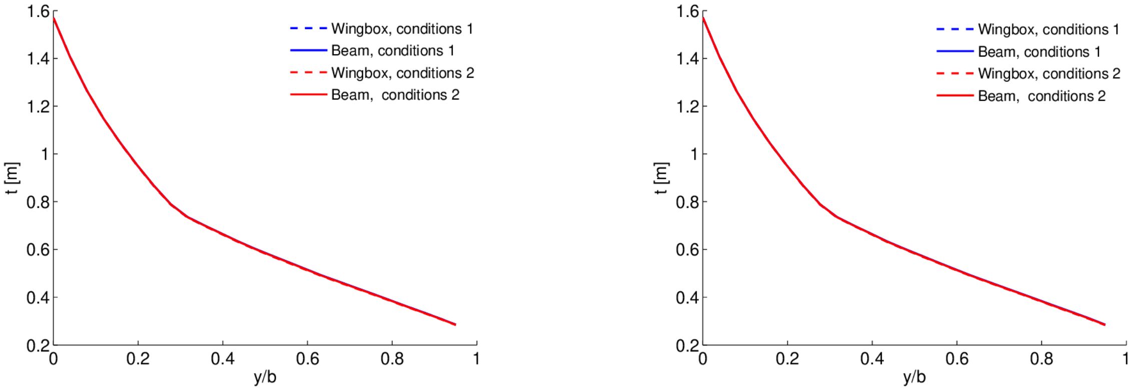


Figure 4.15: Comparison of aeroelastic simulation results using wing-box and beam stick structural models - maximum wing thickness, Euler (left) and RANS (right) simulation at $M = 0.85$, $C_L = 0.5$ (condition 1) and $M = 0.6$, $\alpha = 5°$ (condition 2)

4.3.3 Case C: Wing-fuselage geometry, wing-box structural model, $M = 0.85$

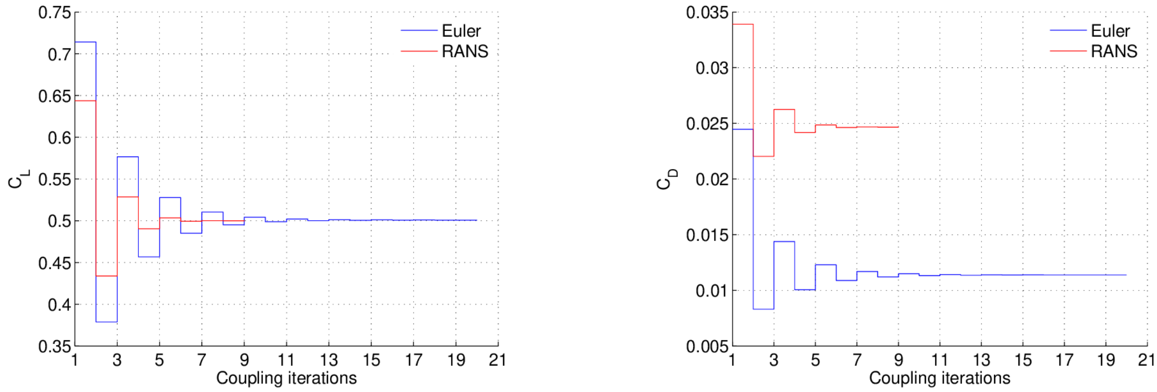


Figure 4.16: Convergence of elastic solution - aerodynamic forces, the wing-fuselage geometry with the wing-box structural model, RANS simulation

Convergence of the aerodynamic forces during the aeroelastic solution is shown in Figure 4.16. The simulation solving Euler flow converged after 20 coupling iterations. In the case of the RANS flow 9 coupling iterations were enough to get the residual reduction below prescribed tolerance. The Table 4.7 presents the computational cost and the resultant forces coefficients. The cost of the aeroelastic simulation is about 4 and 2 times higher compared to the aerodynamic solution in the Euler and RANS cases, respectively.

The drag coefficient in the RANS aeroelasic simulation, in the cruise operating conditions, is about 5.1% higher compared to the aerodynamic solution of the rigid wing. The negative pitching moment, thus the balancing force of the horizontal tail unit, is about 22% higher comparing the same cases.

	Euler		RANS	
	Rigid	Elastic	Rigid	Elastic
No. of coupling iterations	-	20	-	9
Total no. of CFD iterations	1790	8683	1495	3358
$\alpha[^\circ]$	0.72	2.03	1.58	3.12
C_L	0.5003	0.5009	0.4998	0.5001
C_D	0.0117	0.0114	0.0232	0 .0247
C_m^a	-0.1683	-0.1628	-0.0511	-0.0729

Table 4.7: Computational cost and resultant aerodynamic forces coefficients - wing-fuselage geometry, required $C_L = 0.5$

Figure 4.17 presents contours of the pressure coefficient on upper surface of the wing-fuselage configuration. Clearer insight on results is given in the Figure 4.18, where the pressure distribution is plotted over the chosen wing sections. The position of the sections is defined in the percents of the half-span counting from the airplane symmetry plane. The result examination suggests that in both Euler and RANS aeroelastic simulations the wing torsional deformation together with higher required angle of incidence resulted in redistribution of pressure over wing surfaces. The effect of the wing torsional deformation is more obvious in the sections near the wing tip, where the suction was decreased as a result of the sectional angles of attack decrease (see left part of the Figure 4.19) compared to rigid cases.

Plot of the loading distribution over the wing span in the right part of the Figure 4.19 shows that in both cases, Euler and RANS simulations, the wing deformation led to increased loading of the wing inboard part, while the outboard part was alleviated.

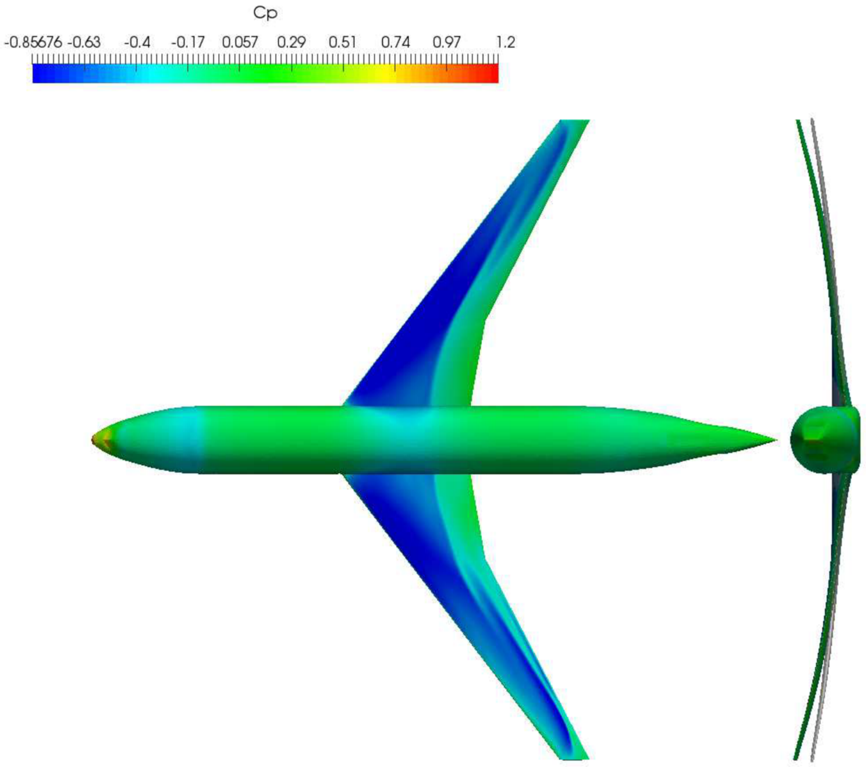


Figure 4.17: Comparison of Euler (upper half) and RANS (bottom half) aeroelastic solution - contours of the surface pressure coefficient distribution at static aeroelastic equilibrium state

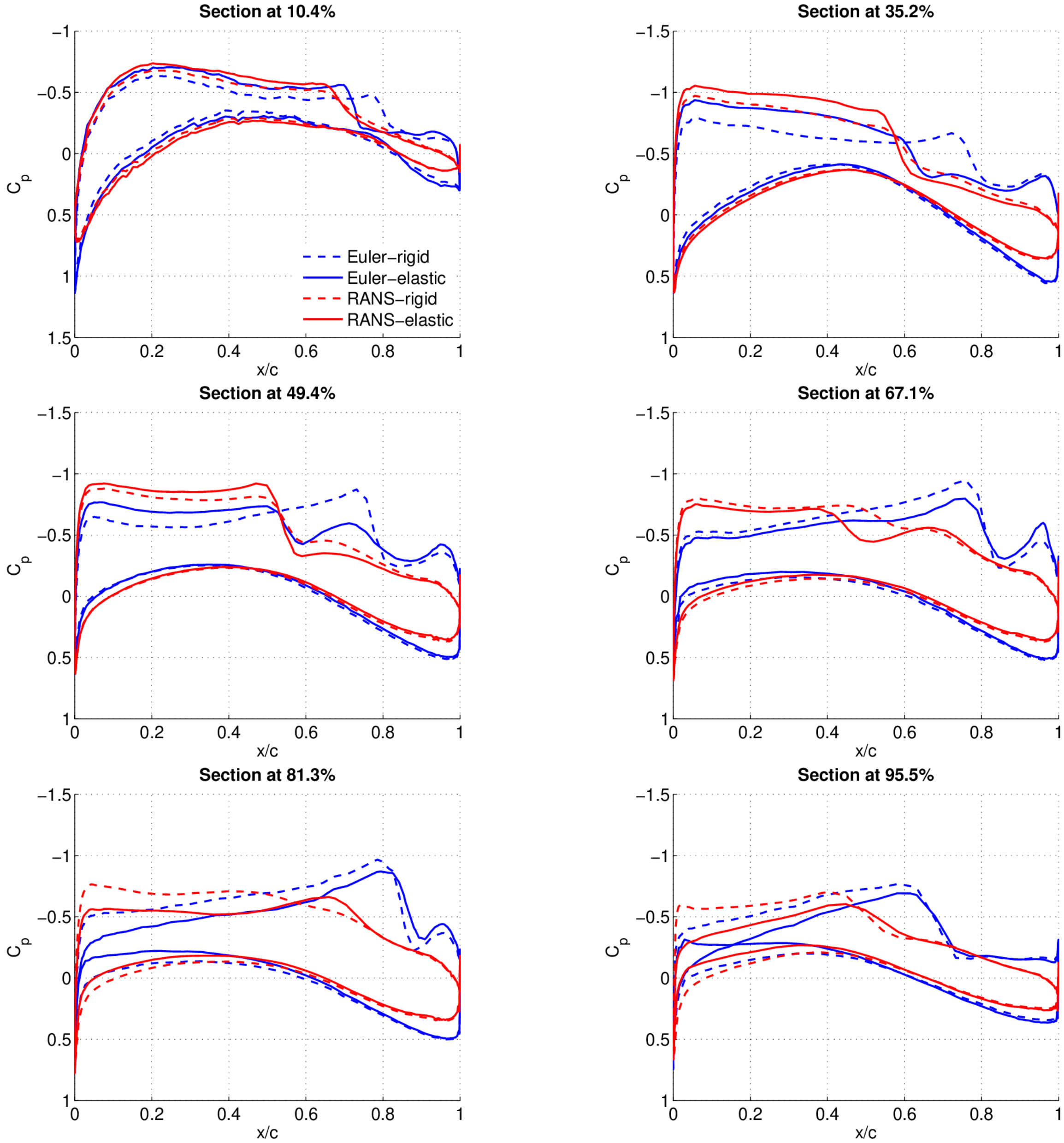


Figure 4.18: Comparison of surface pressure coefficient distribution at chosen sections of the wing - CRM wing-fuselage geometry

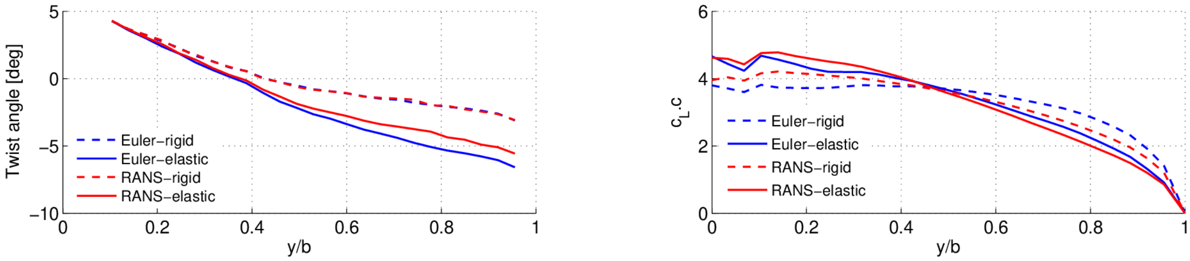


Figure 4.19: Comparison of rigid and aeroelastic solutions - the wing twist angle and the wing loading distribution

4.4 Summary

The first focus of the presented test cases was a validation of the computational tool functionality. Different aerodynamic and structural models were used in order to test the tool at various conditions. The considered aerodynamic models were wing-only and wing-fuselage configurations, the fuselage was assumed as rigid. In both cases, the wing-box structural models represented the wing structure. The effect of gravity forces was included considering wing structure mass only. The masses of other components were neglected, although it is obvious that they might influence the static aeroelastic results.

The performed analyses have shown the effect of the wing flexibility on the aerodynamic characteristics and load distribution over the wing. The primary cause of the load redistribution is the wing twist due to aerodynamic forces. The effect of wing weight was an alleviation of the load, as it was expected.

The beam stick structural model was employed in order to evaluate the ability of the computational aeroelasticity tool to handle simplified model of a wing structure. The results show that the simplified model can be employed but adjustment of the model is required. The need for adjustment arises from the properties of fluid-structure interface definition applied in the tool, which is based on the radial basis functions method. There must be some structural nodes which are not in the plane of the model. This requirement was fulfilled by additional nodes, placed in the wing root section, connected to beam node by rigid elements. The cases, employing the beam stick model, also validated proposed method for design of a simplified structure model. The aeroelastic solution using the reference wing-box model was compared with the one employing the beam model. The aeroelastic solutions were compared considering several flow conditions and flow models. The results are comparable in respective cases.

The effect of flow model employed in aerodynamic prediction on aeroelastic solution was evaluated. The aerodynamics was predicted by solving either Euler or RANS equations. In both cases, the aerodynamic forces are calculated from the pressure acting on the wing surface and subsequently interpolated to the structural model. Thus, the main source of dissimilarity between considered cases originates in different pressure distribution, which is caused by the diffusive effects in the viscous flow. In the presented cases, the Euler aeroelastic solution resulted in the higher wing tip deflection than in the RANS solution. Therefore, assuming that the flow predicted by the RANS simulation is closer to the reality, the Euler aeroelastic prediction over-predicts the wing loading in these particular cases.

Chapter 5

Aerodynamic Shape Optimization of Elastic Wing

5.1 Introduction

An aerodynamic shape optimization using high fidelity flow solvers has been employed for improvement of aircraft aerodynamic design over the last decades. The growing interest in this field was enabled by developments of the Computational Fluid Dynamics (CFD) solvers. CFD solvers became the accepted analysis tools in the aerospace industry reducing the number of tunnel measurements and flight tests during an aircraft development. Due to large number of design variables usually needed for aerodynamic shape design of aircraft, the gradient-based algorithms combined with adjoint solvers are the only meaningful methods for practical application. Locality of those algorithms is a drawback if applied in design space where multiple local optima are likely to occur, the wing shape design is probably such a case. This restricts the process to find only a local optimum near an initial starting point. The solution might be the hybrid optimization algorithms combining non-deterministic (gradient-free) and gradient-based approaches.

The practical application of the gradient-based methods has been probably started by introduction of adjoint sensitivity analysis for Navier-Stokes equation by Pironneau in 1973 [68] and later for incompressible Euler equations [69]. The application in transonic flow regime was enabled by adjoint derivation for compressible Euler equation by Jameson in 1988 [70]. Later he extended the adjoint for Navier-Stokes equations [71]. But the stability and reliability was problematic for long period of time. Nowadays, there are only few adjoint Navier-Stockes solvers applying linearized turbulence models, others rely on approximation by frozen eddy viscosity.

Obviously, the aircraft design is a multi-disciplinary problem. Increased flexibility of aircraft primary structure, as result of modern material application, requirement for lightweight structure and aerodynamically efficient shapes, even emphasizes the multi-disciplinary nature. The wing deformation due to aerodynamic load results, among others effects, in aerodynamic characteristics change. Thus, the performance gain, as result of rigid model optimization, might be decreased or neglected, if applied to real aircraft. The solution might be inclusion of an airframe elasticity to aircraft shape optimization.

In the section, the aerodynamic shape optimization of a common airliner elastic wing is presented and compared with optimization of the same wing assuming rigid structure. The aim is an evaluation of possible benefit and the computational cost of the aerodynamic shape optimization of the elastic wing. A gradient-based optimization approach is applied in connection with adjoint method used for calculation of aerodynamic forces gradients. The computational aeroelasticity tool is employed for estimation of the elastic wing aerodynamic characteristics.

5.2 Principle of Aerodynamic Shape Optimization

The scheme of applied aerodynamic shape optimization loop is given in Figure 5.1. An optimization algorithm directs a decision making in shape design process in order to improve desired aerodynamic characteristics (drag, glide ration, ...) by minimizing relevant objective function. A parameterization method is employed to describe a given geometry by set of parameters creating a design space. Since the parameterization deforms the surface mesh of the geometry, a mesh deformation tool must be incorporated to propagate the shape deformations into a CFD volume mesh. In the next step, the flow field is solved using either CFD solver, in case of a rigid model optimization, or coupled CFD with Computational Structural Mechanics (CSM) solver, in case of an elastic model. The flow solution provides the values of flow field variables and integral aerodynamic characteristics of the current design.

Gradients of desired variables (drag, lift, moment coefficients), with respect to all surface mesh nodes displacements, are calculated on current shape using adjoint of flow equations solver. In the elastic optimization case, the gradients are calculated on current aeroelastic deformed shape. The gradients with respect to design parameters are obtained by multiplication of the surface gradient vector by parameterization Jacobian matrix. The function and gradient values are fed to the optimizer and the loop is repeated until convergence criteria are met.

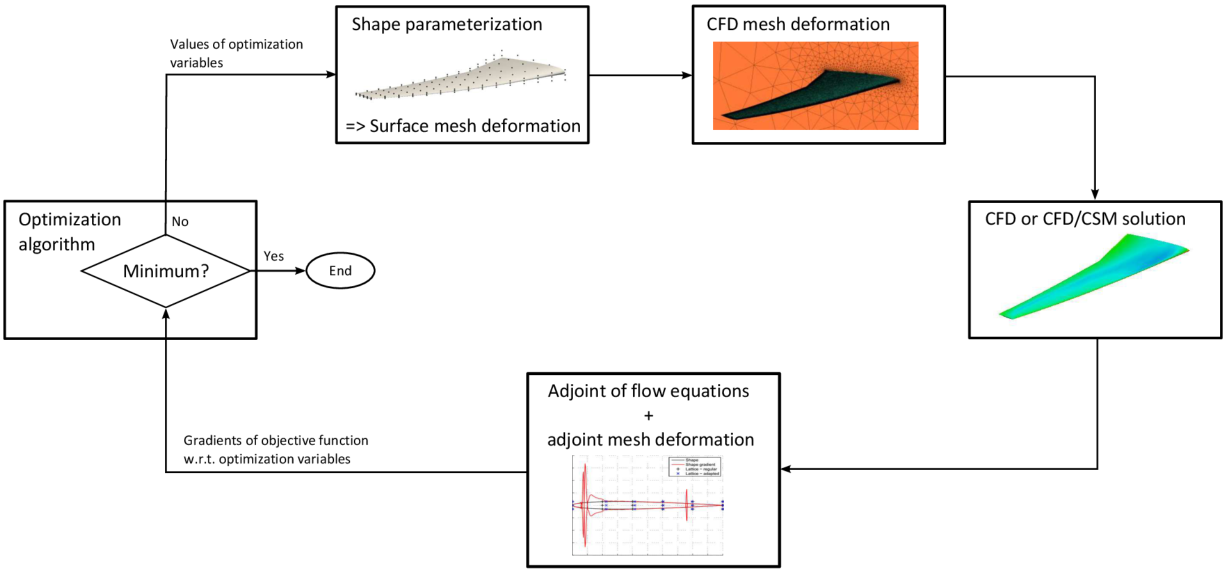

Values of optimization variables
Shape parameterization
=> Surface mesh deformation
CFD mesh deformation
Optimization algorithm
No
Minimum?
Yes
End
CFD or CFD/CSM solution
Adjoint of flow equations
+
adjoint mesh deformation
Gradients of objective function
w.r.t. optimization variables

Figure 5.1: The aerodynamic shape optimization

5.3 Tools

Flow and adjoint solvers

The employed flow and adjoint solver was Edge described in section 2.3.2.

The flow was solved as compressible inviscid. Assuming steady and fully turbulent flow for all designs and the constrained minimum wing wetted area, only lift-induced and wave drag can be minimized. Both flow features can be resolved by Euler flow simulation. Therefore, the optimization using the Euler flow simulation can give reliable and computationally less expensive estimation of the drag reduction which would be achieved by RANS flow simulations.

Aeroelastic solver

The computational aeroelasticity tool (see Section 2.6), coupling the CFD solver Edge with the beam finite element solver introduced in Section 2.4.1, was employed for the static aeroelastic calculation.

Geometry parameterization

A Free Form Deformation (FFD) parameterization based on NURBS was employed for a geometry parameterization. This implementation uses RBF coordinates transformation in order to better control deformations and geometric constraints. Description of the method and a study of numerical properties can be found in [72]. The main idea of this particular FFD method is to fill an orthogonal control points lattice by the geometry using the RBF coordinate transformation. Thus, the parameterization behaves as the lattice would be fitted to a geometry. The Figure 5.2 shows a geometry surrounded by a control points lattice, the geometry after transformation is shown in the Figure 5.3.

The study given in [72] suggests that the highest possible NURBS degree is beneficial from the perspective of the computational cost and the obtained optimization result. Therefore, the highest allowed NURBS degree will be imposed in all test cases.

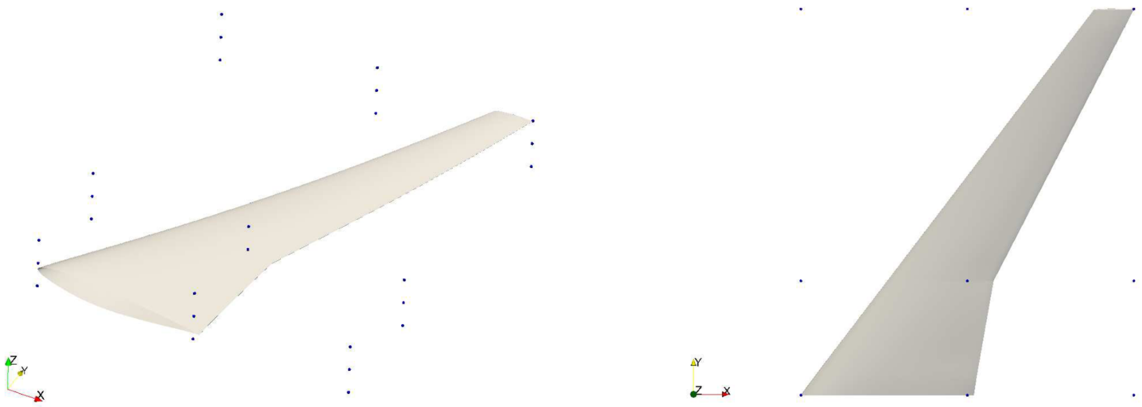


Figure 5.2: FFD parameterization lattice of control points

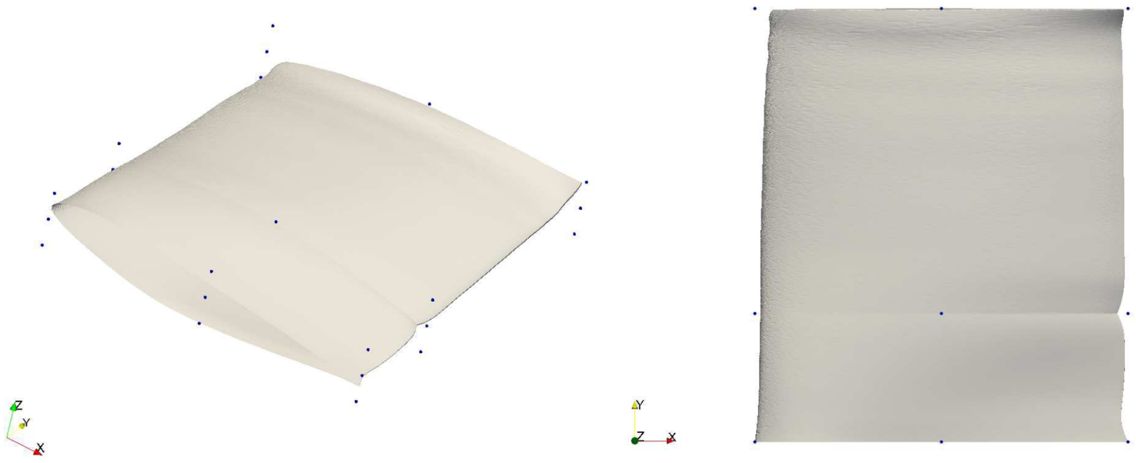


Figure 5.3: Wing geometry fitted to orthogonal FFD lattice of control points

Optimization algorithm

The optimizations were performed by gradient-based optimization algorithm - Sequential Quadratic Programming (SQP) in NLPQLP [58].

Mesh generation and deformation

The initial CFD grid was created in Ansys ICEM CFD. The spring analogy mesh deformation tool was employed to propagate surface shape changes, resulting from the optimization process, into the CFD volume mesh. For large deformations, such as the wing deformation due to the aerodynamic loading, the RBF mesh deformation combined with the spring analogy method was incorporated. The advantage of the RBF based mesh deformation is a capability to handle large deformations but the surface deformation is not exactly recovered. The error depends on the settings of the mesh deformation solver. Therefore, the inexact surface shape was corrected by the spring analogy mesh deformation, in order to avoid introduction of an error into the optimization process by incorrect mesh deformation.

5.4 Test Cases

5.4.1 Common description

The aerodynamic shape optimization test cases, based on cases proposed in [73], concern drag minimization of the transonic wing of the airliner model (so called Common Research Model [60]) at Mach number $M = 0.85$ and altitude $h = 11000m$. The lift coefficient required for steady horizontal flight in such conditions is $C_L = 0.5$. The formulation of the optimization was:

$$\begin{aligned} \text{minimize } F(X) &= C_D \\ \text{subject to } C_L &= 0.5 \\ C_m &\geq -0.1754 \\ V &\geq V^{CRM} \\ t_f &\geq t_f^{CRM} \\ t_r &\geq t_r^{CRM} \\ \text{fixed trailing edge} & \\ \text{wing planform shape fixed} & \end{aligned} \tag{5.1}$$

The goal is to decrease the drag while the lift remains constant. The geometrical constraint on the internal volume is meant to ensure minimal space for the fuel. Other constraints on the wing thicknesses at front and rear spar positions, t_f and t_r respectively, are meant to guarantee the same minimal structural height, thus the minimal structural stiffness is ensured. Therefore, the structural stiffness can remain frozen in the elastic wing optimization case.

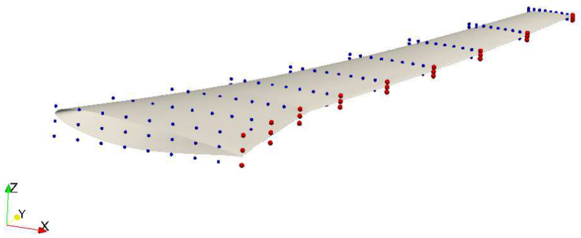


Figure 5.4: The FFD lattice around CRM wing, blue points are free to move in Z direction, red points are fixed

Table 5.1: Aerodynamic characteristics of the initial CRM wing

	Rigid	Aeroelastic
Lift coefficient, C_L	0.5000	0.5000
Drag coefficient, C_D	0.0120	0.0120
Pitch moment coefficient, C_m	-0.1750	-0.1717
Angle of attack, α	0.721	0.816

The constraints on fixed wing planform shape and fixed trailing edge are not explicitly prescribed but they are fulfilled by the choice of optimization variables. The Figure 5.4 shows the FFD lattice fitted on real geometry (in fact, the lattice is orthogonal and the geometry is transformed to fill it). The blue points in the figure are optimization parameters allowed to move in vertical direction, therefore the airfoil shapes together with twist angle are able to change, while the planform shape is kept constant. The fixation of trailing edge is achieved by fixed FFD lattice points along the trailing edge, represented by red color in the figure.

5.4.2 Initial design

Rigid wing optimization

The initial geometry for the rigid wing optimization cases was the CRM wing, as it was given in [60]. The CRM geometry was designed to provide common representative model of an airliner operating in transonic conditions for validation of the state-of-the-art CFD solvers. Thus, the geometry of the CRM wing corresponds to the flight shape (1-g shape, bended and twisted due to aerodynamic loading) at nominal cruise conditions at Mach $M = 0.85$ and $C_L = 0.5$ at altitude 12 000 m. Therefore, the subject of the optimization is a rigid wing flight shape at its nominal cruise conditions.

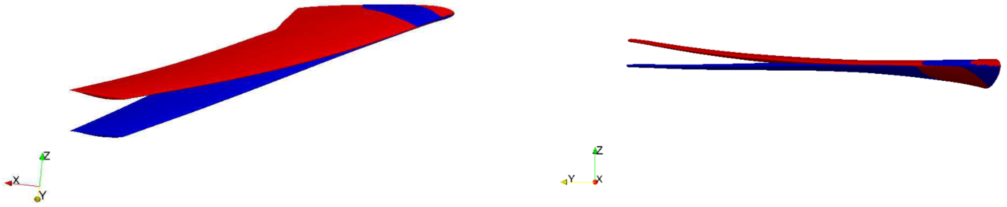


Figure 5.5: Initial wing geometry - flight shape (red) for rigid wing optimization and jig shape (blue) for elastic wing optimization

Elastic wing optimization

For the elastic wing optimization, it is desirable to use an undeformed wing geometry, as the static aeroelastic analysis determines correct flight shape for given operating condition. Therefore, so called jig shape was designed from the flight shape giving the undeformed wing surface and structural models. The jig shape CFD mesh was created from the mesh of the rigid wing flight shape. Applied structural model is a beam stick finite element model created by 41 beam elements. Each node of the beam element is connected to two additional nodes by rigid elements, which are beneficial for coupling with aerodynamic surface.

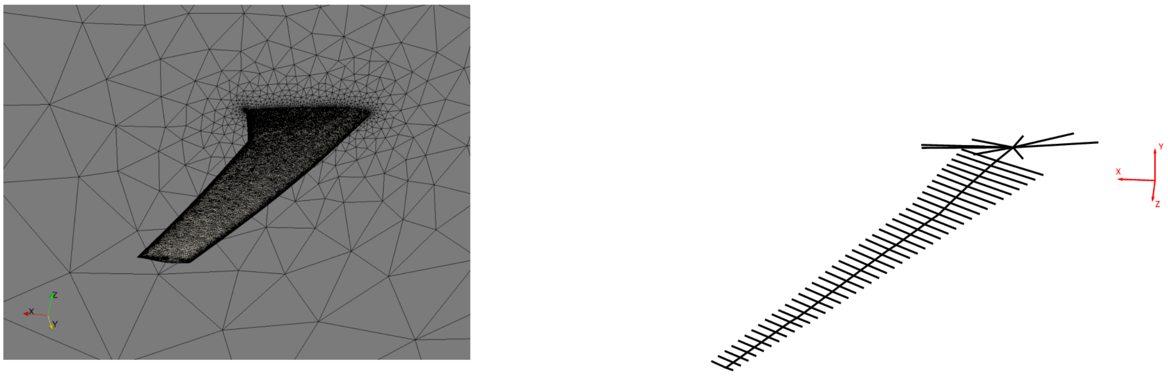


Figure 5.6: Computational models - the CFD mesh with 933502 nodes (left) and the beam finite element model with 41 beam elements (right)

5.5 Results

5.5.1 Rigid wing optimization

The single point optimization of the rigid wing was performed in order to evaluate influence of the parameterization set-up on the optimization results and convergence. Three sets of FFD parameters were considered with 18, 60 and 216 optimization variables. In all cases the maximum NURBS degree was used.

N	$C_{D_{opt}}$	$C_{L_{opt}}$	$C_{m_{opt}}$	V_{opt}	Cost^b	% of C_D decrease
Baseline	0.0120	0.5000	-0.1750	84.4453	–	–
18	0.0115	0.5001	-0.1700	85.6504	48	4.16
60	0.0114	0.5001	-0.1755	84.8349	56	5.38
216	0.0112	0.5000	-0.1754	84.4451	84	6.72

[a] N is number of optimization parameters

[b] Total cost of optimization in terms of number of flow and adjoint of flow solutions

Table 5.2: Optimized CRM wing design - influence of number of variables on the local optimum

The results, introduced in Table 5.2 and the cost function convergence history in Figure 5.7, show that increasing number of parameters allows optimization to converge to better solution, because higher number of parameters gives more control over the geometry shape. The optimization objective - the drag coefficient - was reduced by 6.72% compared to the baseline, from 0.0120 to 0.0112 in the finest parameterization case. The coarsest parameterization allowed the objective to decrease by 4.16%. The cost in terms of flow and adjoint solution numbers grows with increasing number of parameters, as the optimization process requires more function evaluations in order to satisfy given convergence tolerance.

In all cases, the target wing lift coefficient as well as the maximum allowed negative pitching moment were reached at the optimum. The tuning of airfoil shapes and twist angle resulted in redistribution of the lift along wing span closer to ideal elliptical distribution of the $C_L c$, as it is illustrated in Figure 5.8. Moreover, the lift resultant was shifted towards wing root, what may result in lower bending loading of the wing structure. The same figure shows the sectional twist along the wing span suggesting the negative twist was increased at optimum in all cases.

The Figure 5.9 compares the baseline geometry and pressure distribution with the optima of all considered cases. It is clear that coarser parameterization does not allow such a fine geometry modification, thus the shock is not completely removed, e.g. at locations near the root and in case of the coarsest parameterization near the tip as well. The finest parameterization optimization resulted

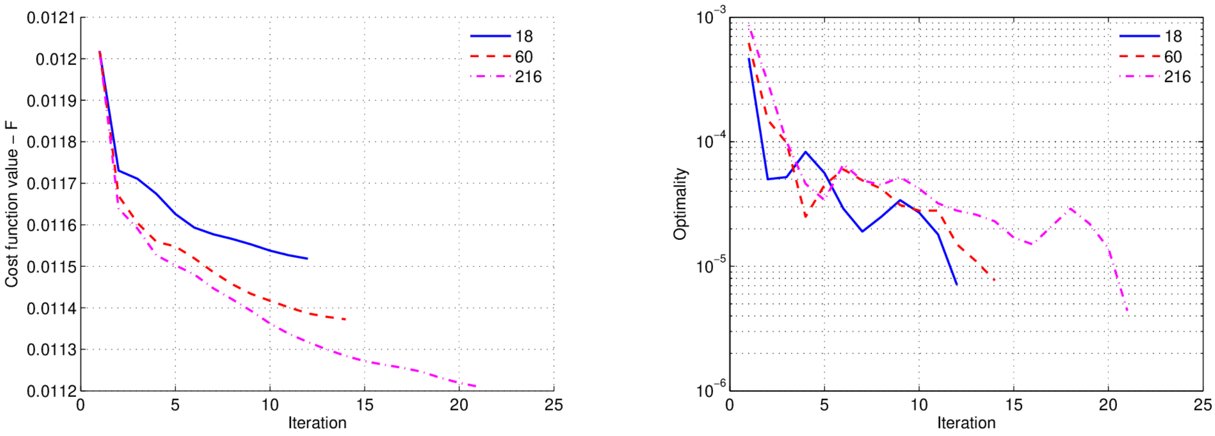


Figure 5.7: History of the single point optimization of the rigid wing for various number of optimization parameters

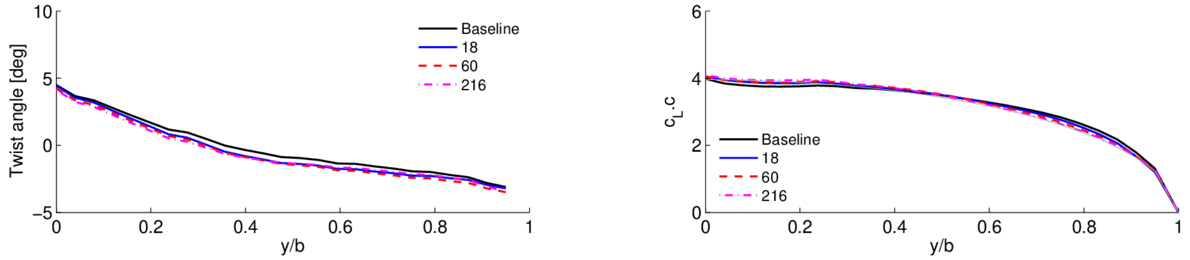


Figure 5.8: Single point optimization of the rigid wing - comparison of the wing twist and $C_L c$ distribution for various number of optimization parameters (grey line shows the elliptical distribution of $C_L c$)

in nearly shock free solution at the nominal cruise condition. The pressure change is more gradual towards the trailing edge of the optimized wing contrary to the baseline wing with steep increase of the pressure due to shock.

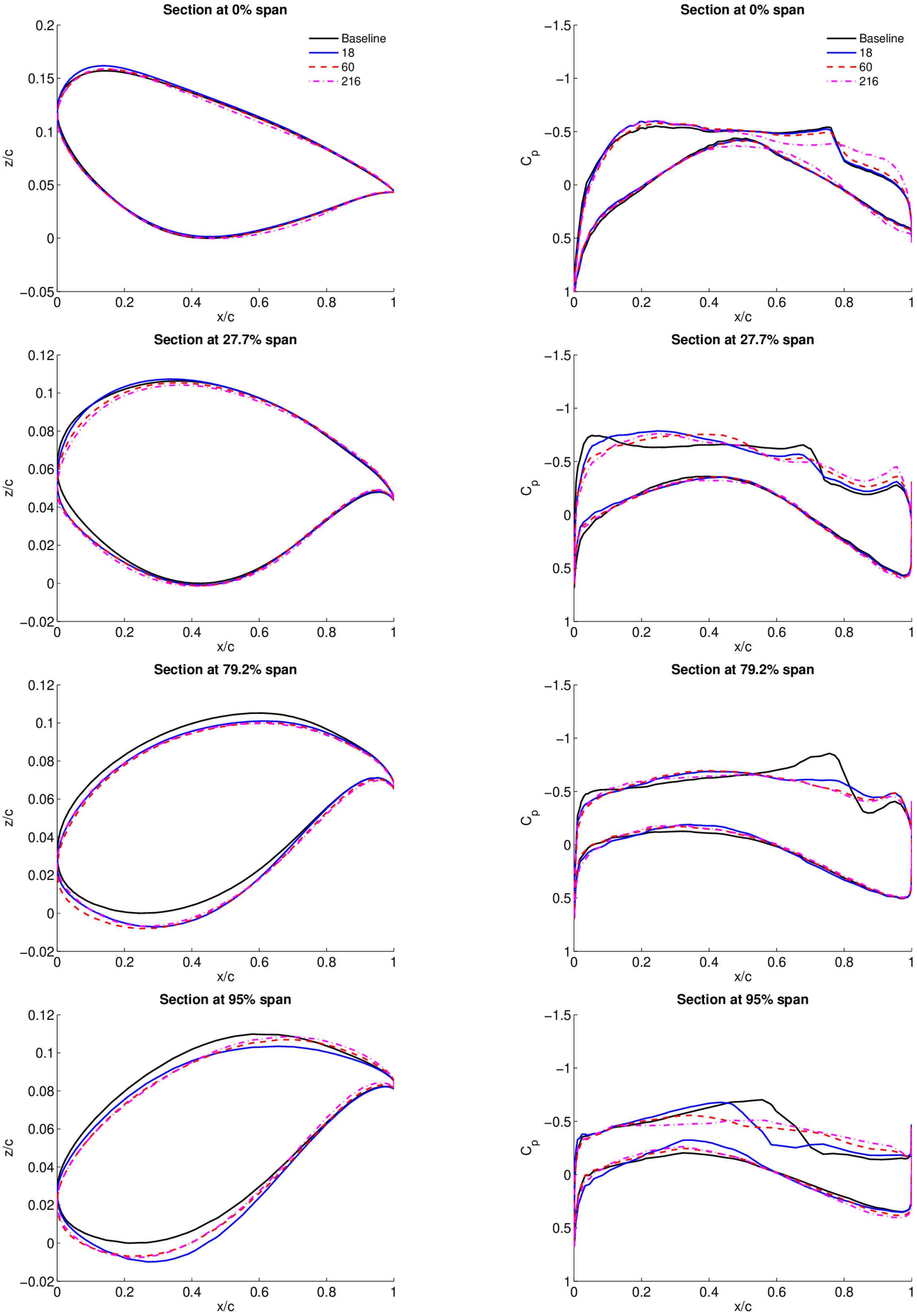


Figure 5.9: Single point optimization of the rigid wing - comparison of section geometry and pressure distribution for various number of optimization parameters

5.5.2 Elastic wing optimization

The comparison of the results of the rigid and elastic wing optimization cases is presented in Table 5.3. The constraint imposed on the value of the lift and the pitch moment coefficients was fulfilled, in both cases. The wing internal volume constraint was satisfied in the rigid wing case. In the other case, the violation was about 0.19%, what might be considered as constraint satisfaction. The optimization objective - the drag coefficient - was reduced by 6.72% and 6.17% in the rigid and elastic wing optimization, respectively. The computational cost in the case of the elastic wing, in terms of number of the flow and the adjoint solutions, is nearly twice as high as in the rigid wing case.

Table 5.3: Comparison of the rigid and elastic wing optimization results

	$C_{D_{opt}}$	$C_{L_{opt}}$	$C_{m_{opt}}$	V_{opt}	Cost[a]	C_D decrease
Baseline	0.0120	0.5000	-0.1750	84.46	–	–
Rigid wing	0.0112	0.5000	-0.1754	84.46	84	6.72%
Elastic wing	0.0113	0.4998	-0.1755	84.28	149	6.1739%

[a] Total cost of optimization in terms of number of flow and adjoint of flow solutions

The plots in Figure 5.10 suggest that constraints on the wing thickness were fulfilled in both cases. The thickness at rear spar position remained nearly unchanged in both cases, while the thickness at the front spar was slightly increased near the wing root.

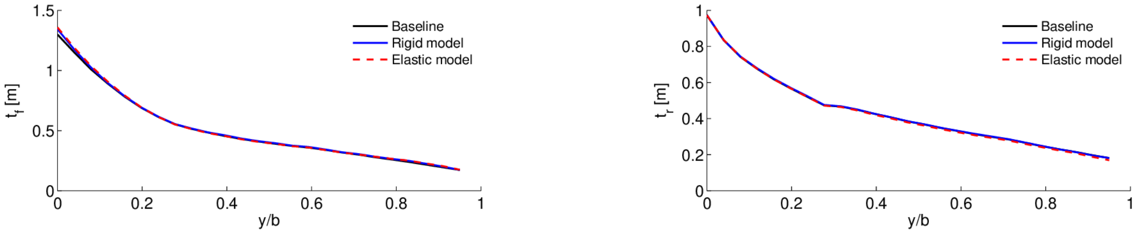


Figure 5.10: Comparison of the wing thickness, at position of the front (left) and rear (right) wing spar

The tuning of the wing shape and the twist angle of the wing sections resulted in redistribution of the lift along wing span. The resultant distribution is closer to ideal elliptical C_Lc distribution, as it is illustrated in Figure 5.11. Moreover, the lift force resultant was shifted towards wing root, what might result in lower bending loading of the wing structure. The same figure shows that the wing twist was changed towards higher negative values in both cases.

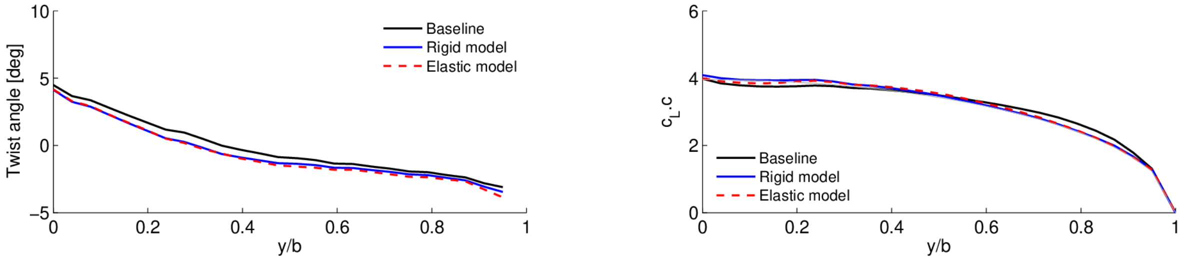


Figure 5.11: Comparison of span-wise wing twist and lift distributions (grey line shows the elliptical distribution of C_Lc)

The Figure 5.13 presents the pressure coefficient distribution over the upper surface and at chosen wing sections. The plots suggest that in both cases the optimization resulted in nearly shock free solution at the nominal cruise condition. The pressure change is more gradual towards the trailing edge of the optimized wing contrary to the baseline wing with steep increase of the pressure due to shock.

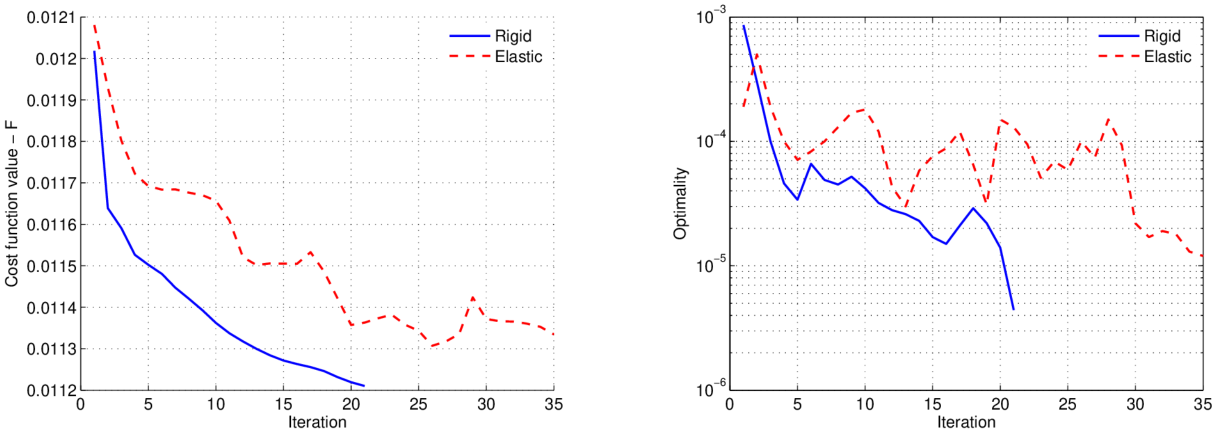


Figure 5.12: History of the optimization

The optimization history plots shown in Figure 5.12 suggest that the objective function decrease was smooth and the optimization criteria were met after 21 iterations in rigid wing case. The peaks in the plot of the objective function history in case of the elastic wing together with abrupt changes of the optimality imply that there was some source of an error in the optimization chain. The probable source is the neglect of the structural deformation influence on the gradient of the aerodynamic forces and moments. Thus, the implementation of the coupled fluid-structure adjoint equation solver is required.

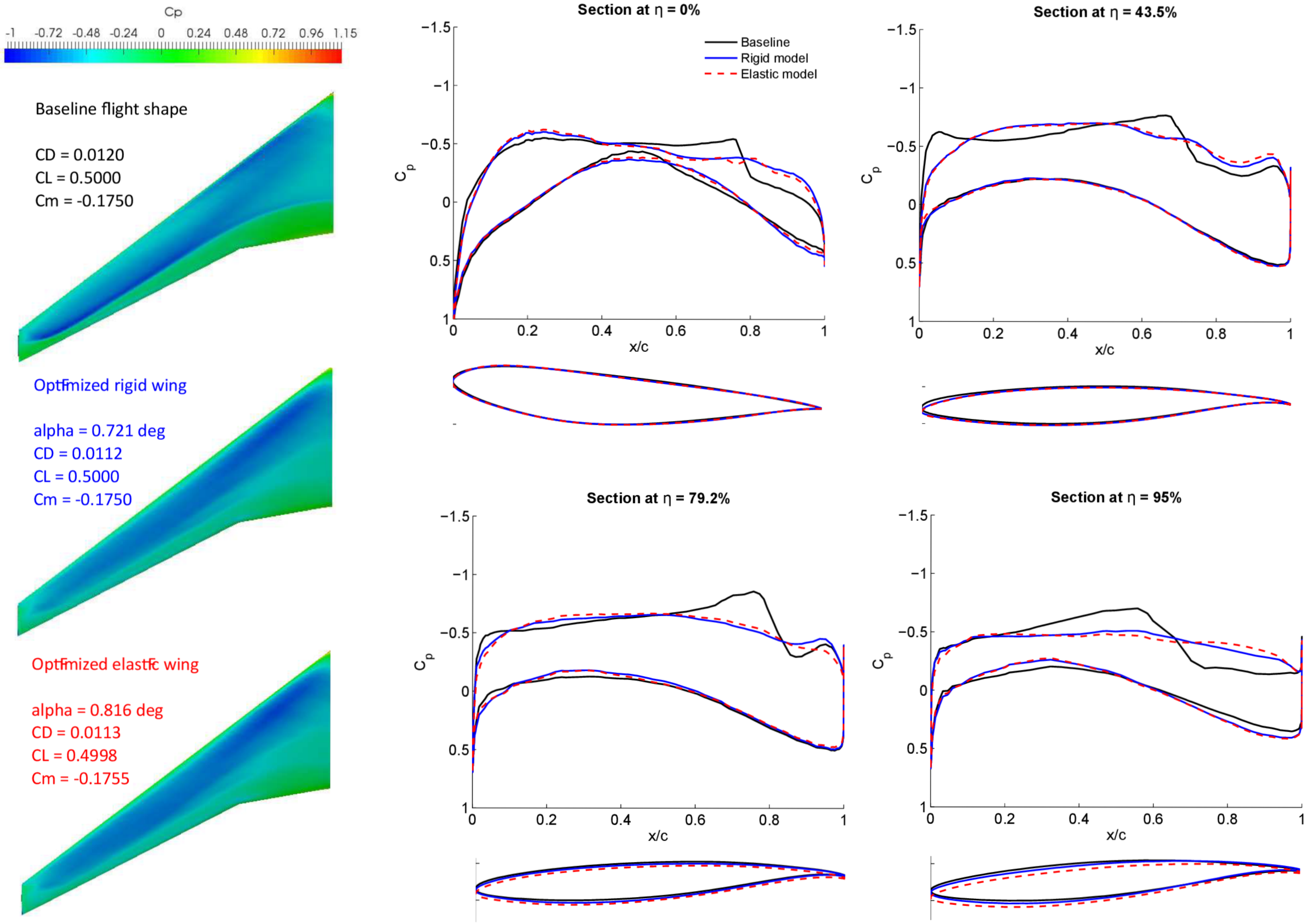


Figure 5.13: Comparison of surface pressure coefficient distribution - baseline and optimized wings

5.6 Summary

The chapter presents the optimization of the elastic wing compared to the rigid wing optimization. Different initial designs were applied in respective cases for the reasonable comparison. The rigid wing was optimized starting from the flight shape at the design operating conditions. Whereas in case of the elastic wing the initial design was the jig shape, which under loading by aerodynamic forces at the design flight conditions deforms to the same flight shape as in rigid case.

The resultant drag reduction is nearly same in both cases but the computational cost in the elastic wing case is almost doubled. Moreover, the objective function decrease was not as smooth as it might be expected. The several increases of the objective value during the optimization were probably result of inexact gradient calculation. Although the gradients were calculated on the aeroelastic deformed shape, the error in gradient calculation applying pure flow adjoint equation, thus neglecting the influence of the wing structure deformation on the aerodynamic forces gradients, is significant.

Thus, the derivation and implementation of the coupled fluid-structure adjoint equations is a necessity for the further work on the elastic wing optimization. The real benefit of this approach is expected in the multi-point optimization considering more operation conditions.

Chapter 6

Numerical Study of Benchmark Supercritical Wing at Flutter Condition

6.1 Introduction

Transonic flutter and LCO are two dynamically non-linear phenomena whose prediction is largely dependent on wind tunnel and flight testing. On the computational side, the most used method for aerodynamic predictions in aeroleastic computations is the doublet lattice method [7, 8]. It is essentially a linear method and as such fails predicting non-linear aeroelastic phenomena. Despite the progress in the Computational Fluid Dynamics over the past decades the predictions of the transonic flutter, and LCO and in general, of the non-linear aeroelasticity, remains still a challenge [74, 75, 76]. The computational predictions of the non-linear phenomena are facing several challenges: the fluid-structure coupling, code validation and time synchronization [77]. In his presentation, Bendiksen points out the time synchronization as the most important issue, because a loosely coupled aeroelastic code can give incorrect aeroelastic solution. Similar observations were made in [78, 79, 80, 81] and resulted in the extension of the loosely coupled schemes to a second order accuracy in order to mitigate the time synchronization problem. Apart from the time synchronization problem, other challenges may include the prediction of the flow separation and the flow transition on fluid side and the structural damping on structural side.

This chapter presents the author's contribution to the research oriented on the assessment of time synchronization for the CFD-CSM coupled problem which was performed by comparing time converged solution of the test case using loosely and strongly coupled fluid-structure interaction. The test case used here is a transonic flow around the Benchmark Super-Critical Wing (BSCW) at flutter condition. It is a rectangular wing and at the chosen flow condition the flow is without separation. Results were compared to the experimental data provided by NASA [82]. A flutter boundary for different Mach numbers was calculated using the proposed method. The results were published online as the journal paper "Computational Fluid Dynamics Study of Benchmark Supercritical Wing at Flutter Condition" [83].

6.2 CFD-CSM Solver

6.2.1 CFD code Edge

The CFD flow solver used in the study is the Edge solver described in the subsection 2.3.2.

The convergence within each time step is controlled by setting a number of minimum and maximum subiterations or by the level of residual reduction. In this study a fixed number of subiterations was specified to get a minimal reduction of the residuals below certain value, usually 2.5 orders of magnitude.

6.2.2 Structural solver

The employed structural solver is part of the Edge solver. It solves differential linear equations valid for a dynamic system with small displacements

$$M\ddot{x} + C\dot{x} + Kx = f \tag{6.1}$$

where x is the vector of structural coordinates, and $f(t)$ is the corresponding vector of forces. The M , C and K are the mass, damping and stiffness matrix, respectively. The equation of motion is reduced to the form,

$$a_k\ddot{q}_k + 2\zeta_k a_k\omega_k\dot{q}_k + a_k\omega_k^2 q_k = Q_k, \qquad\qquad k \in [1, N_m] \tag{6.2}$$

where ζ is the damping ratio for mode k and

$$Q_k = \psi_k^T f \tag{6.3}$$

is the corresponding generalized force. The structural damping matrix, C, is a linear combination of the mass and stiffness matrices M and K, i.e. considered as a proportional or Rayleigh damping. In this study the damping was set to $C = 0$, because a flutter case was solved.

6.2.3 Coupling scheme

The coupling scheme is a partitioned coupling scheme. The data between solvers were exchanged on subiteration level. The Figure 6.1 shows a comparison of this scheme with the usual Conventional Staggered Schene (CSS) [78, 79].

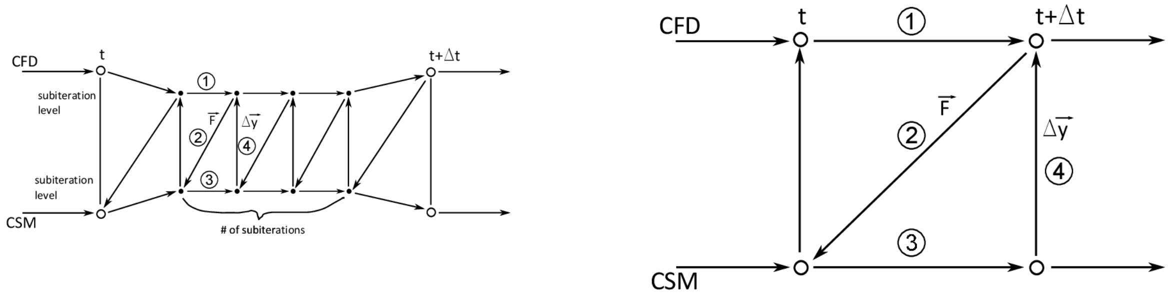


Figure 6.1: Coupling schemes - coupling on subiteration level (left) and Conventional Serial Staggered Scheme (right)

6.2.4 Mesh deformation

Since the selected test case has just plunge and pitch modes, the rigid motion (translation and rotation) of the CFD mesh was applied instead of the mesh deformation. The rigid mesh motion is prescribed using transformation of the modal coordinates to the physical displacement and rotation. This approach replaced the surface deformation, defined as a linear combination of the mode shapes and modal coordinates, and its the propagation to the volume mesh using mesh deformation techniques. The other attribute of the approach is the constant shape of whe wing at any displacement as it is illustrated in the Figure 6.2. The figure shows a pitching airfoil in the original and rotated positions obtained using either linear mode shapes or by rotating a mesh around pivotal point.

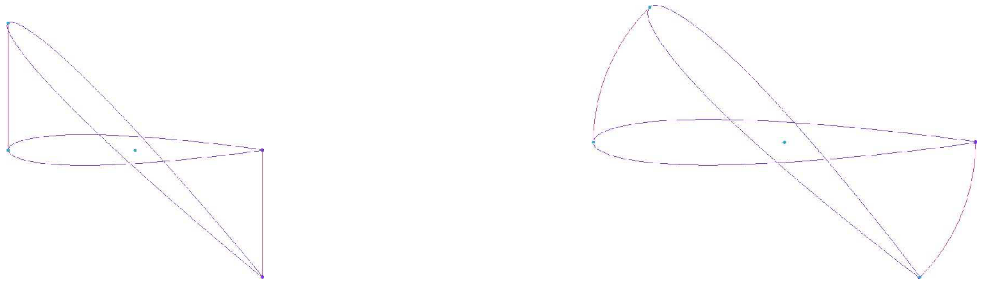


Figure 6.2: Rotation of the airfoil prescribed through linear mode shapes (left) and through rotational matrix (right)

6.3 Test Case

6.3.1 Experimental setup

The test case is the Benchmark Super Critical Wing (BSCW), which was experimentally tested at NASA TDT facility [82]. This test case was chosen as one of the cases used in the Aeroelastic Prediction Workshop I and II [14, 15, 84]. The test data include measurements for the rigid, forced oscillations and aeroelastic problem. Although the wing itself is rigid, the elastic behavior in aeroelastic tests is allowed by using the Pitch and Plunge Apparatus, which allows simultaneous plunge and pitch of the wing [82]. Figure 6.3 shows the BSCW model in NASA TDT wind tunnel section.

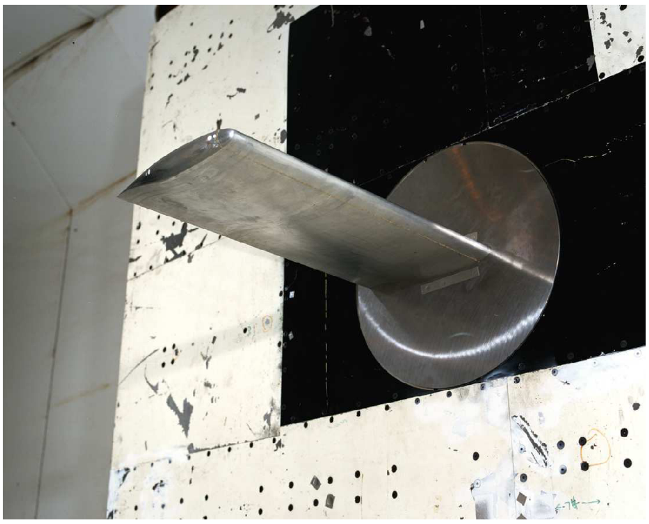


Figure 6.3: BSCW wing at NASA TDT wind tunnel section, Courtesy of NASA

The test case considered in this study is a case of flow around the BSCW wing at Mach number $M = 0.74$, angle of incidence $\alpha = 0°$, Reynolds number $Re = 4.45$ million and dynamic pressure $p = 8082Pa$ $(168psf)$ which is an experimentally measured flutter onset for this wing. The test medium is R-12 coolant gas. The case involves a transonic flow with shock wave on the upper side of the wing. The flow at the angle of attack $\alpha = 0$ degrees is fully attached.

The structural model has two modes: the plunging mode with frequency $f = 3.3Hz$ and pitching mode with frequency $f = 5.2Hz$ ([84]). The pivotal point location is at 50% of the airfoil chord - see Figure 6.4.

6.3.2 Computational setup

The CFD mesh shown in the Figure 6.5 consists of 13 millions points and is composed of tetra, prism and penta elements.

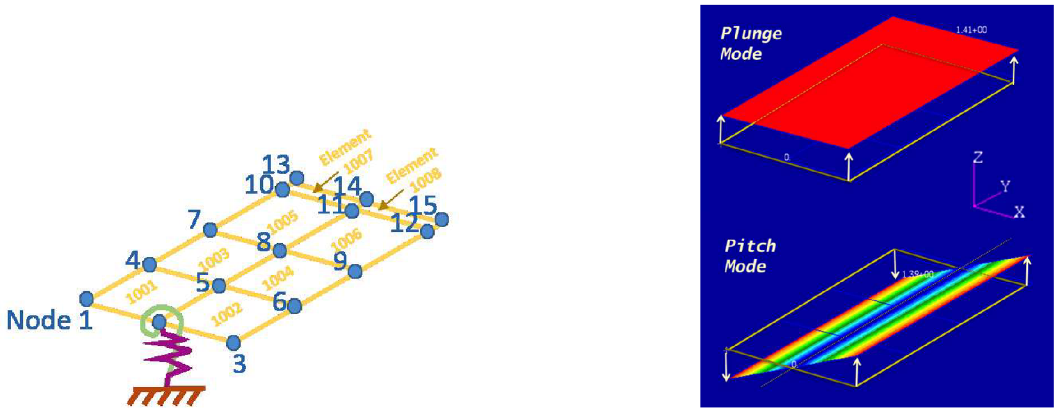


Figure 6.4: Finite element model [84] diagram (left) and modes diagram (right)

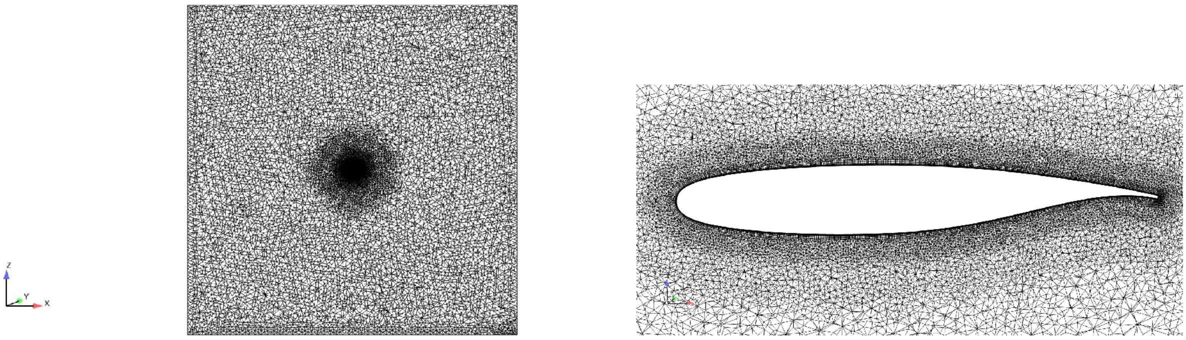


Figure 6.5: Computational mesh (left) and detail of the mesh around wing (right)

The analysis was ran in the unsteady Raynold-Averaged Navier-Stokes (URANS) mode using the Spalart Allmaras model [85]. Each solution started with a steady RANS analysis which was subsequently used as an initial guess for URANS analysis of a steady wing. The URANS analysis was ran for about 1000 time steps to get the well-converged URANS solution. This solution was then used as an initial guess for URANS coupled aeroelastic analysis. The coupled aeroelastic calculations modeled 5s, in one case 10s, of the physical time.

6.4 Results

6.4.1 Time step convergence study

Three time steps were used, $\Delta t = 0.001secs$, $\Delta t = 0.002secs$ and $\Delta t = 0.004secs$. The time step study was done using the strong coupling scheme, which means that the coupling is performed at each subiteration of the numerical scheme. The different number of subiterations was set for particular time steps - for $\Delta t = 0.001secs$ it was set to 20, $\Delta t = 0.002secs$ set to 30 and $\Delta t = 0.004secs$ set to 40 - so that the total reduction of residuals within each time step was approximately the same. The optimal number of subiteration was found for the case $\Delta t = 0.002secs$ by running the coupled simulation for different settings of subiteration number. The convergence of the scheme vs. number of subiterations was measured by the critical damping ratio. The results starts to be independent of the number of subiterations once it is larger than 20. To allow for some margins, the number of subiterations equal to 30 was chosen.

The similar study was done for the CSS scheme which is an example of the weak coupled scheme. The time steps were identical to those used in the previous tests and the number of subiteration for the weak coupled scheme was the same as for the strong coupled scheme.

Figure 6.6 shows the critical damping ratio for the weak coupled scheme compared to the strongly coupled scheme. For the clarity of the graphs, values of the damping and frequency for the pitching mode are shown.

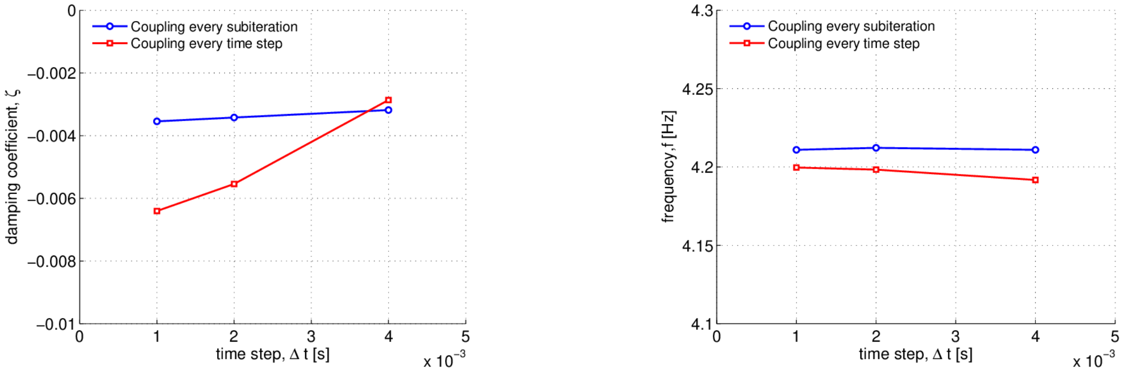


Figure 6.6: Time step study for weak coupled scheme compared to strong coupled scheme - damping coefficient (left) and frequency (right) of pitching mode

In the case of the strong coupled scheme the damping ratio is slightly changing with the time step, the frequency is almost unchanged. The loose coupled scheme converges to a different aeroelastic solution as the time-step is reduced; this corresponds to finding of Bendiksen [77]. The value of the damping ration for the smallest time steps is more than twice as large as the value of the damping for the largest time step and there is no sign that the smaller time step would tend to produce asymptotically converged solution. The dependency of the frequency on time step is not as strong as for the damping coefficient. There is some effect of the coupling scheme on the frequency, the weak coupled scheme predicts lower frequency than the strong coupling scheme. Both schemes predict lower than experimentally measured frequency $f = 4.3Hz$ but the difference is small. Thus, the result is in good agreement with the experiment.

6.4.2 Influence of number of exchanges during each time step on flutter solution

From the previous analysis it is clear that the strong coupled scheme leads to reasonably time converged result. However, the scheme has several drawbacks, the major one is the scheme time consumption compared to weak coupled scheme. The main reason is time burden incurred by the mesh deformation which has to be used after each exchange between CFD and CSM solvers. One of the potential savings can be gained by reducing the frequency of exchanges within each time step.

Figure 6.7 shows the results where the exchange is done in every subiteration and then 2, 3, 5 and 7 times during each time step. For completeness the loosely coupled scheme result was added.

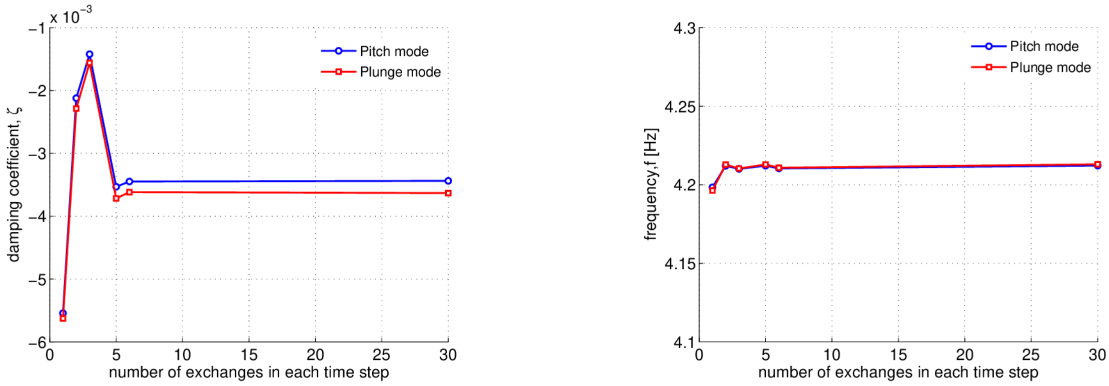


Figure 6.7: Weak vs strong coupling - damping coefficient (left) and frequency (right)

The result start to be reasonably converged when the number of exchanges during each time step is five.

6.4.3 Estimate of flutter dynamic pressure

The BSCW wing analysis at the dynamic pressure, which was determined experimentally as a flutter onset dynamic pressure, led to predictions indicating that the wing model is already in flutter. Therefore, the two other lower dynamic pressures were calculated and the final values for the damping ratios and frequency were interpolated and used to estimate the values of the "CFD determined" flutter onset. The Mach number was kept at $M = 0.74$.

Figure 6.8 shows the values of the damping ratio and frequency vs. dynamic pressure.

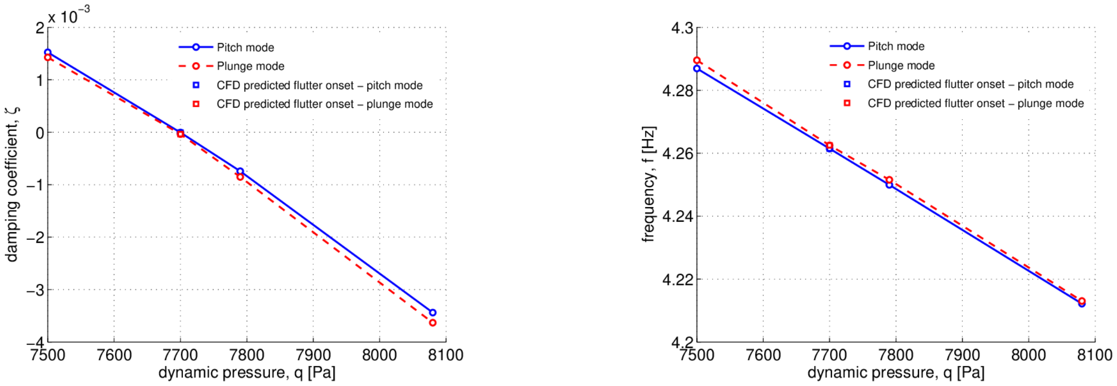


Figure 6.8: Damping coefficient (left) and frequency (right) vs dynamic pressure

From the interpolation curves the value of the flutter onset dynamic pressure was estimated to be at $q = 7700Pa$, which is point where the damping ratio for the pitching motion is zero. The case was then simulated at this value of the dynamic pressure modeling total of 10s sequence. The final values of the damping ratio and frequency are shown in the Table 6.1.

	Pitch		Plunge	
	Damping	Frequency	Damping	Frequency
		Hz		Hz
CFD, $q = 8082Pa$	−0.0034382	4.212	−0.0034382	4.212
CFD, $q = 7700Pa$	−0.0000052	4.261	−0.0000352	4.262
WT, $q = 8082Pa$	0.0	4.3	0.0	4.3

Table 6.1: Results of CFD flutter analysis for pitch and plunge modes compared to the wind tunnel data

Figure 6.9 shows the 10 seconds sequence of the displacement and pitch angle of the airfoil.

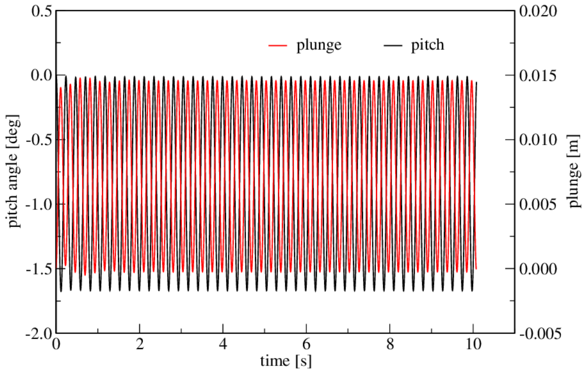
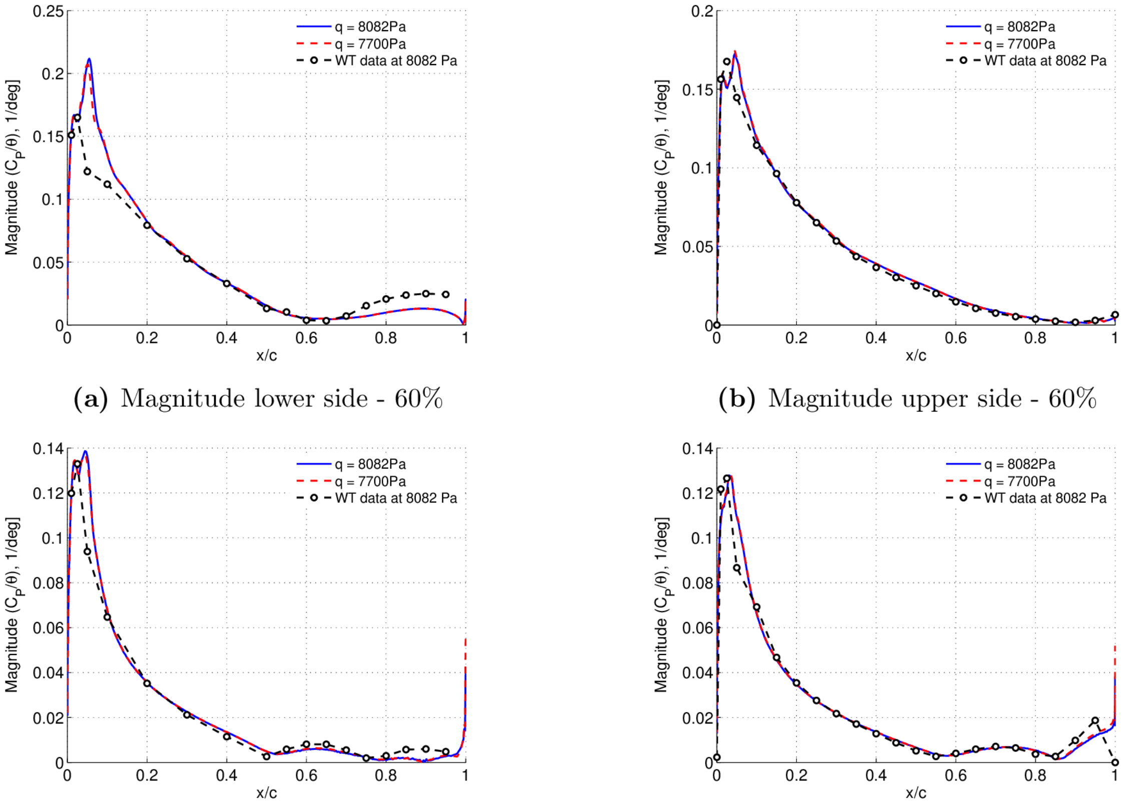


Figure 6.9: Pitch and plunge coordinate for the flutter dynamic pressure

Figures 6.10 and 6.11 show magnitude and phase of the pressure frequency response function (FRF) at 60% and 95% of span compared to experimental data.



(a) Magnitude lower side - 60%

(b) Magnitude upper side - 60%

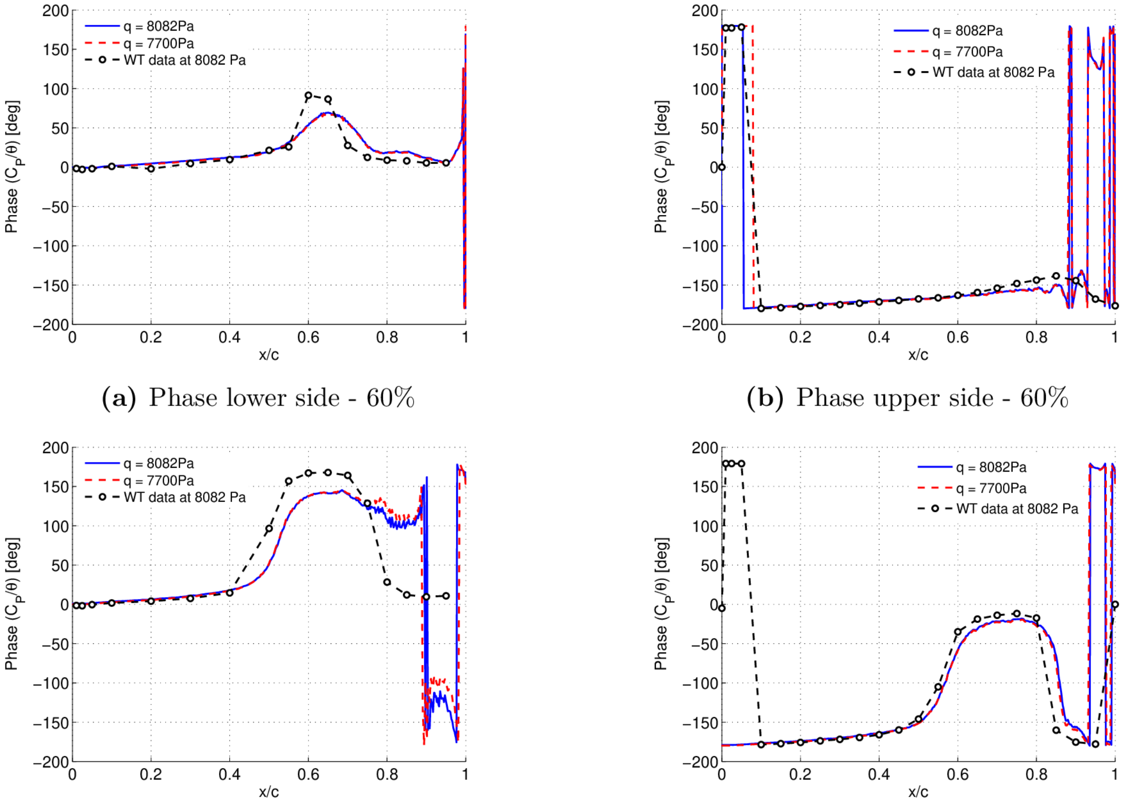
(c) Magnitude lower side - 95%

(d) Magnitude upper side - 95%

Figure 6.10: Magnitude in 60% and 95%

The comparisons of the magnitude of FRF show fairly good agreement with the slight difference around the leading edge on the lower side of the wing at 60% of span and region of trailing edge at 95% of span. As of now, there is not any explanation what is the cause to the first difference. The second one is most probably caused by the trailing edge separation, which was not accurately modeled by CFD method.

The comparison of the FRF phase shows good agreement of the numerical analysis with the experimental results. The differences observable at the upper sides of the 60% and 95% are just the 360 degree shifts, thus the results are comparable with the experiment. The most significant difference, on the lower side at 95%, might be caused by the problems of the CFD to predict the trailing edge separation.



(a) Phase lower side - 60%

(b) Phase upper side - 60%

(c) Phase lower side - 95%

(d) Phase upper side - 95%

Figure 6.11: Phase in 60% and 95%

6.4.4 Estimate of the flutter boundary

The similar procedure as used in subsection 6.4.3 was used to calculate the flutter boundary for the BSCW wing at angle of attack $\alpha = 0 degs$ and range of Mach numbers from $M = 0.6$ to $M = 0.9$. Figure 6.12 shows the comparison of the measured dynamic flutter pressure and flutter freuquency vs. Mach number with the values obtained numerically.

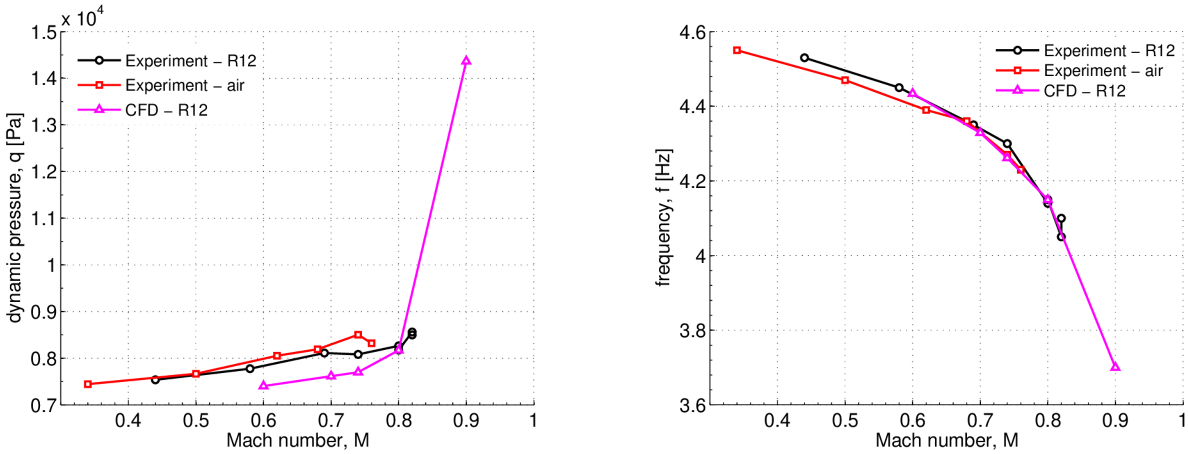


Figure 6.12: Flutter boundary - comparison between CFD and wind tunnel data

The wind tunnel data are available at range of Mach number up to $M = 0.82$ for two gases - air and R-12 gas. The CFD solution is available for R-12 gas only. The figures show strong non-linear dependency of the flutter dynamic pressure on Mach number for Mach numbers larger than $M = 0.82$

which can not be predicted by linear computational methods.

6.5 Summary

This chapter presents numerical study of the Benchmark Super-Critical Wing at a condition where the wind tunnel data indicated a flutter onset. The wing is considered rigid, the aeroelasticity is brought to the system by PAPA apparatus which allows model to rotate around pivotal point and to plunge in the vertical direction. The flow around the wing is transonic without any large areas of the flow separation, which makes the case an ideal test case for validation of the coupled numerical analysis.

The analysis shows that the most important factor influencing the validity of the result is the coupling scheme. The second factor influencing the accuracy of the flutter predictions is the ability of the CFD code to predict various features of the transonic flow including flow separation and the transition of the laminar boundary layer to turbulent boundary layer.

The strong coupled scheme gives result which is changing marginally with different time step. As long as the other factors, which potentially require using very short time steps (such as the flow separation), are not present, the scheme can use of relatively large time step during time integration, as long the sufficient convergence to the pseudo steady state solution within each time step is guaranteed. The loosely coupled scheme, such as the Serial Staggered Scheme, converged to the different result as the time step of the simulation was refined and there is no indication that with refined time step the results would show asymptotic convergence to the ”correct” result.

The subsequent tests with ”loosening” the strong coupling scheme by using only several subiteration levels to time synchronization rather then every subiteration showed that such an approach can produce similar results to the strongly coupled scheme as long as there are five or more time synchronization steps during each time step.

Chapter 7

Conclusions

7.1 Outcome of the Thesis

Computational aeroelasticity tool

A tool for simulation and design optimization of the static aeroelastic models has been implemented and tested. It allows to include effect of the static structural deformation of the airframe in the aerodynamic analysis. The implementation is based on the communication between CFD and CSM solvers using I/O operations via hard-copied files. The influence of various settings of the flow solver on the convergence of an aeroelastic solution was tested. The results suggest that optimal settings are case dependent.

Additionally, a numerical structural solver has been designed and implemented and subsequently applied in the computational aeroelasticity tool. It is linear elastic preprocessor and solver for structural models using the beam finite elements. Moreover, it is able to solve static deformation of the finite element model consisting of arbitrary elements. In this case, the stiffness and mass matrices must be provided by an external preprocessor.

Inverse design method for equivalent beam model

The method of inverse design of the beam model properties has been proposed and tested. The method finds the beam properties to get equivalent static deformation of the wing to the reference one under the same loading. The results suggest that the proposed inverse method might be applicable for design of simplified structural model of a complex wing geometry.

Numerical study of wing at flutter condition

The time synchronization scheme for the coupled CFD-CSM problem was evaluated on the wing flutter test case. The research was conducted within the Aeroelastic Prediction Workshop II in cooperation with colleagues from Swedish Defense Research Agency, FOI. The results were compared with the test case performed at the NASA Transonic Dynamic Tunnel and they are in close agreement with experimental results.

7.2 Conclusion

The objective of the thesis was to design and implement a tool for aeroelastic simulations. The tool should be applicable for the aerodynamic design and analysis of an elastic airplane.

The aeroelastic simulation tool, programmed in the Matlab environment, was designed in the way that it allows to employ arbitrary flow and structural solvers. The fluid-structure transformation interface was defined using the radial basis functions.

The practical applicability of the computational aeroelasticity tool in the aerodynamic analysis was tested on the cases employing different types of the aerodynamic and structural models. The test cases have shown that tool is able to handle complex geometries, such as a wing-fuselage model of transport aircraft with the swept wing.

The application of the designed simulation tool in the aerodynamic shape optimization of the elastic model was evaluated. The optimization problem was defined aiming for a drag reduction of the transonic wing with aerodynamic and geometric constraints. The aerodynamic design optimization was performed for the single operating condition. The results suggest that further development must be performed as the employed gradient calculation neglected influence of the wing deformation due to aerodynamic loading, on the gradients of aerodynamic forces.

7.3 Perspectives

Computational aeroelasticity tool

The obvious continuation of the presented work would be the application of the computational aeroelasticity tool for solving time-dependent dynamic aeroelastic problems. In order to do this, the tool must be implemented in the way that the communication between solvers is based on the direct approach via random-access memory. Therefore, the tool as well as the linear elasticity solver must be programmed in the language such as Fortran or C and implemented into the applied CFD solver Edge.

Inverse design method for equivalent beam model

There is still space for further development of the method, such as extension for design of beam stick model dynamically equivalent to higher fidelity structural model. It could also solve the suggested problem of indeterminacy of some resultant stiffness characteristics.

Aero-structural optimization

The result of aerodynamic shape optimization of elastic wing have shown the need for coupled fluid-structure equations adjoint solver. The solver should extend existing flow equation adjoint solver implemented in the CFD package Edge. This extension will allow to perform aero-structural optimization which would increase design efficiency of aircraft from both aerodynamic and structural perspectives.

Bibliography

[1] L. Garrick, “Aeroelasticity-frontiers and beyond,” *Journal of Aircraft*, vol. 13, no. 9, pp. 641–657, 1976.

[2] H. Von Wagner, “Dynamischer auftrieb von tragflugeln,” *Zeitchrift fuer Angewandte Mathematik und Mecdhanik*, vol. 5, p. 17, 1925.

[3] H. G. Küssner, “Zusammenfassender bericht über den instationären auftrieb von flügeln,” *Luftfahrtforschung*, vol. 13, no. 12, pp. 410–424, 1936.

[4] H. G. Küssner, “Schwingungen von flugzeugflugeln,” *Jahrbuch der deutscher Versuchsanstalt für Luftfahrt (especially Section E3 Einfluss der Baustoff-Dämpfung, pp 319–320)*, 1929.

[5] T. Theodorsen, “General theory of aerodynamic instability and the mechanism of flutter,” *NACA Report 496*, 1935.

[6] E. C. Yates, jr., “Modified-strip-analysis method for predicting wing flutter at subsonic to hypersonic speeds.,” *Journal of Aircraft*, vol. 3, no. 1, pp. 25–29, 1966.

[7] W. P. Rodden, “The development of the doublet-lattice method,” *Proceedings of the CEAS/International Forum on Aeroelasticity and Structural*, vol. 2, pp. 1–7, June 1997.

[8] R. Yurkovich, “Status of unsteady aerodynamic prediction for flutter of high-performance aircraft,” *Journal of Aircraft*, vol. 40, pp. 832–842, September-October 2003.

[9] O. O. Bendiksen and K. A. Kousen, “Transonic flutter analysis using the euler equations,” 1987.

[10] E. Lee-Rausch and J. T. Baitina, “Wing flutter computations using an aerodynamic model based on the navier-stokes equations,” *Journal of Aircraft*, vol. 33, no. 6, pp. 1139–1147, 1996.

[11] J. J. Alonso, L. Martinelli, and A. Jameson, “Multigrid unsteady navier-stokes calculations with aeroelastic applications,” in *AIAA paper*, Citeseer, 1995.

[12] E. Thompson, R. Kolonay, F. Eastep, and J. Camberos, “Aeroelastic analysis with transpiration enabled euler flow solver,” in *48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, p. 2331, 2007.

[13] Z. Feng, A. Soulaı, Y. Saad, *et al.*, “Nonlinear krylov acceleration for cfd-based aeroelasticity,” *Journal of Fluids and Structures*, vol. 25, no. 1, pp. 26–41, 2009.

[14] J. Heeg, P. Chwalowski, J. P. Florance, C. D. Wieseman, D. M. Schuster, and B. Perry, “Overview of the aeroelastic prediction workshop,” AIAA 2013-783, 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition. January, 2013.

[15] J. Heeg, P. Chwalowski, D. M. Schuster, D. Raveh, A. Jirasek, and M. Dalenbring, “Plans and example results for the 2nd aiaa aeroelastic prediction workshop,” AIAA 2015-0437, 56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference. January.

[16] K. G. Bhatia, "Airplane aeroelasticity: practice and potential," *Journal of aircraft*, vol. 40, no. 6, pp. 1010–1018, 2003.

[17] F. J. Blom, "Considerations on the spring analogy," *International journal for numerical methods in fluids*, vol. 32, no. 6, pp. 647–668, 2000.

[18] D. Zeng and C. R. Ethier, "A semi-torsional spring analogy model for updating unstructured meshes in 3d moving domains," *Finite Elements in Analysis and Design*, vol. 41, no. 11, pp. 1118–1139, 2005.

[19] S. Jakobsson and O. Amoignon, "Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization," *Computers & Fluids*, vol. 36, no. 6, pp. 1119–1136, 2007.

[20] E. Luke, E. Collins, and E. Blades, "A fast mesh deformation method using explicit interpolation," *Journal of Computational Physics*, vol. 231, no. 2, pp. 586–601, 2012.

[21] Z. Wang, V. Parthasarathy, and N. Hariharan, "A fully automated chimera methodology for multiple moving body problems," *International Journal for Numerical Methods in Fluids*, vol. 33, no. 7, pp. 919–938, 2000.

[22] R. Mittal and G. Iaccarino, "Immersed boundary methods," *Annu. Rev. Fluid Mech.*, vol. 37, pp. 239–261, 2005.

[23] J. Yang, S. Preidikman, and E. Balaras, "A strongly coupled, embedded-boundary method for fluid–structure interactions of elastically mounted rigid bodies," *Journal of Fluids and Structures*, vol. 24, no. 2, pp. 167–182, 2008.

[24] B. Yildirim, S. Lin, S. Mathur, and J. Y. Murthy, "A parallel implementation of fluid–solid interaction solver using an immersed boundary method," *Computers & Fluids*, vol. 86, pp. 251–274, 2013.

[25] A. H. van Zuijlen and H. Bijl, "Implicit and explicit higher order time integration schemes for structural dynamics and fluid-structure interaction computations," *Computers & structures*, vol. 83, no. 2, pp. 93–105, 2005.

[26] 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*, vol. 195, no. 17, pp. 1973–2001, 2006.

[27] W. G. Dettmer and D. Perić, "A new staggered scheme for fluid–structure interaction," *International Journal for Numerical Methods in Engineering*, vol. 93, no. 1, pp. 1–22, 2013.

[28] T. Rendall and C. Allen, "Unified fluid–structure interpolation and mesh motion using radial basis functions," *International Journal for Numerical Methods in Engineering*, vol. 74, no. 10, pp. 1519–1559, 2008.

[29] H. Wendland, "Hybrid methods for fluid-structure-interaction problems in aeroelasticity," in *Meshfree Methods for Partial Differential Equations IV*, pp. 335–358, Springer, 2008.

[30] J. R. Martins, J. J. Alonso, and J. J. Reuther, "A coupled-adjoint sensitivity analysis method for high-fidelity aero-structural design," *Optimization and Engineering*, vol. 6, no. 1, pp. 33–62, 2005.

[31] M. Marcelet, J. Peter, and G. Carrier, "Sensitivity analysis of a strongly coupled aero-structural system using the discrete direct and adjoint methods," *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique*, vol. 17, no. 8, pp. 1077–1106, 2008.

[32] O. Amoignon and M. Berggren, "Discrete adjoint-based shape optimization for an edge-based finite-volume solver," 2003.

[33] M. H. Hounjet and J. J. Meijer, *Evaluation of elastomechanical and aerodynamic data transfer methods for non-planar configurations in computational aeroelastic analysis.* National Aerospace Laboratory NLR, 1995.

[34] M. J. Smith, D. H. Hodges, and C. E. S. Cesnik, "Evaluation of computational algorithms suitable for fluid-structure interactions," *Journal of Aircraft*, vol. 37, no. 2, pp. 282–294, 2000.

[35] A. d. Boer, A. H. van Zuijlen, and H. Bijl, "Review of coupling methods for non-matching meshes," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 8, pp. 1515–1525, 2007.

[36] J. R. Cebral and R. Löhner, "Conservative load projection and tracking for fluid-structure problems," *AIAA journal*, vol. 35, no. 4, pp. 687–692, 1997.

[37] A. Beckert, "Coupling fluid (cfd) and structural (fe) models using finite interpolation elements," *Aerospace Science and Technology*, vol. 4, no. 1, pp. 13–22, 2000.

[38] R. Zwaan and B. Prananta, "Fluid/structure interaction in numerical aeroelastic simulation," *International journal of non-linear mechanics*, vol. 37, no. 4, pp. 987–1002, 2002.

[39] T. Rendall and C. Allen, "Improved radial basis function fluid–structure coupling via efficient localized implementation," *International journal for numerical methods in engineering*, vol. 78, no. 10, pp. 1188–1208, 2009.

[40] A. Beckert and H. Wenland, "Multivariate interpolation for fluid-structure-interaction problems using radial basis functions," *Aerosp. Sci. Technol*, pp. 1–11, 2001.

[41] H. K. Versteeg and W. Malalasekera, *An introduction to computational fluid dynamics: the finite volume method.* Harlow: Pearson Prentice Hall, 2nd edition ed., 2007.

[42] A. Tewari, *Aeroservoelasticity: modelling and control.* Springer, 2015.

[43] P. Eliasson, "EDGE, a Navier-Stokes solver for unstructured grids," in *Proc. to Finite Volumes for Complex Applications III*, pp. 527–534, ISBN 1 9039 9634 1, 2002.

[44] A. Jameson, "Time dependent calculations using multigrid, with application to unsteady flows past airfoils and wings," AIAA 91-1596, AIAA 10th Computational Fluid Dynamics Conference, June 24-26, 1991, Honolulu HI, USA.

[45] J. Przemieniecki, *Theory of Matrix Structural Analysis.* Dover Civil and Mechanical Engineering, Dover, 1985.

[46] J. T. Batina, "Unsteady euler airfoil solutions using unstructured dynamic meshes," *Journal, Vol. 28, No. 8*, 1990.

[47] D. Zeng and C. R. Ethier, "A semi-torsional spring analogy model for updating unstructured meshes in 3d moving domains," *Finite Elements in Analysis and Design*, vol. 41, 2005.

[48] C. L. Bottasso, D. Detomi, and R. Serra, "The ball-vertex method: a new simple spring analogy method for unstructured dynamic meshes," *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 39â€"41, 2005.

[49] D. A. Field, "Laplacian smoothing and delaunay triangulations," *Communications in applied numerical methods*, vol. 4, no. 6, pp. 709–712, 1988.

[50] L. A. Freitag, “On combining laplacian and optimization-based mesh smoothing techniques,” *ASME APPLIED MECHANICS DIVISION-PUBLICATIONS-AMD*, vol. 220, pp. 37–44, 1997.

[51] P. I. Liakopoulos and K. C. Giannakoglou, “Unstructured remeshing using an efficient smoothing scheme approach,” in *ECCOMAS CFD 2006: Proceedings of the European Conference on Computational Fluid Dynamics, Egmond aan Zee, The Netherlands, September 5-8, 2006*, Delft University of Technology; European Community on Computational Methods in Applied Sciences (ECCOMAS), 2006.

[52] A. De Boer, M. Van der Schoot, and H. Bijl, “Mesh deformation based on radial basis function interpolation,” *Computers & structures*, vol. 85, no. 11, pp. 784–795, 2007.

[53] I. Ghazlane, G. Carrier, A. Dumont, M. Marcelet, and J.-A. Désidéri, “Aerostructural optimization with the adjoint method,” in *EUROGEN 2011*, 2011.

[54] S. A. Dunn, “Technique for unique optimization of dynamic finite element models,” *Journal of Aircraft*, vol. 36, no. 6, pp. 919–925, 1999.

[55] P. M. Trivailo, T. Gilbert, E. Glessich, and D. Sgarioto, “Inverse problem of aircraft structural parameter identification: application of genetic algorithms compared with artificial neural networks,” *Inverse Problems in Science and Engineering*, vol. 14, no. 4, pp. 337–350, 2006.

[56] M. S. A. Elsayed, R. Sedaghati, and M. Abdo, “Accurate stick model development for static analysis of complex aircraft wing-box structures,” *AIAA journal*, vol. 47, no. 9, pp. 2063–2075, 2009.

[57] R. Guyan, “Reduction of stiffness and mass matrices,” *AIAA Journal*, vol. 3, no. 2, 195.

[58] K. Schittkowski, “Nlpqlp: A fortran implementation of a sequential quadratic programming algorithm with distributed and non-monotone line search-user’s guide,” 2006.

[59] P. T. Boggs and J. W. Tolle, “Sequential quadratic programming,” *Acta numerica*, vol. 4, pp. 1–51, 1995.

[60] J. C. Vassberg, M. A. DeHaan, S. M. Rivers, and R. A. Wahls, “Development of a common research model for applied cfd validation studies,” *AIAA paper*, vol. 6919, p. 2008, 2008.

[61] Z. Lyu, G. K. Kenway, and J. R. Martins, “Aerodynamic shape optimization investigations of the common research model wing benchmark,” *AIAA Journal*, vol. 53, no. 4, pp. 968–985, 2014.

[62] O. Amoignon, J. Navrátil, and J. Hradil, “Study of parameterizations in the project cedesa,” *AIAA Paper*, vol. 570, 2014.

[63] G. K. Kenway, J. R. Martins, and G. J. Kennedy, “Aerostructural optimization of the common research model configuration,” *Group (ADODG)*, vol. 6, no. 7, pp. 8–9, 2014.

[64] G. K. Kenway and J. R. Martins, “Multipoint high-fidelity aerostructural optimization of a transport aircraft configuration,” *Journal of Aircraft*, vol. 51, no. 1, pp. 144–160, 2014.

[65] P. D. Dunning, B. K. Stanford, and H. A. Kim, “Aerostructural level set topology optimization for a common research model wing,” in *10th AIAA Multidisciplinary Design Optimization Conference, 13-17 January 2014, National Harbor, Maryland, USA*, pp. 1–21, 2014.

[66] C. V. Jutte, B. K. Stanford, C. D. Wieseman, and J. B. Moore, “Aeroelastic tailoring of the nasa common research model via novel material and structural configurations,” in *AIAA SciTech Conference*, pp. 13–17, 2014.

[67] “Nasa common research model,” 2012.

[68] O. Pironneau, "On optimum profiles in stokes flow," *Journal of Fluid Mechanics*, vol. 59, no. 01, pp. 117–128, 1973.

[69] O. Pironneau, "On optimum design in fluid mechanics," *Journal of Fluid Mechanics*, vol. 64, no. 01, pp. 97–110, 1974.

[70] A. Jameson, "Aerodynamic design via control theory," *Journal of scientific computing*, vol. 3, no. 3, pp. 233–260, 1988.

[71] A. Jameson, L. Martinelli, and N. Pierce, "Optimum aerodynamic design using the navier–stokes equations," *Theoretical and computational fluid dynamics*, vol. 10, no. 1-4, pp. 213–237, 1998.

[72] O. Amoignon, J. Hradil, and J. Navratil, "A numerical study of adaptive ffd in aerodynamic shape optimization," in *52nd Aerospace Sciences Meeting*, vol. 899, 2014.

[73] L. Osusky, B. H., and Z. D. W., "Aerodynamic design optimization discussion group: Single- and multi-point optimization problems based on the crm wing," aiaa aerodynamic design optimization discussion group, 2013.

[74] E. Dowell, J. Edwards, and T. Strganac, "Nonlinear aeroelasticity," *Journal of Aircraft*, vol. 40, pp. 857–874, September-October 2003.

[75] E. H. Dowell, "Some recent advantages in nonlinear aeroelasticity, fluid-structure interaction in the 21st century," AIAA 2010-3137, 51st AIAA/ASME/ASCE/AHS/ASC Structure, Structural Dynamics and Material Conference, April 12-15, Olrando, Florida, 2010.

[76] O. O. Bendiksen, "Review of unsteady transonic aerodynamics: Theory and applications," *Progress in Aerospace Sciences*, vol. 47, p. 135–167, 2011.

[77] O. O. Bendiksen, "Transonic aeroelasticity: Theoretical and computational challengess," Presentation at I Workshop on Recent Advaces in Aeroelasticity: Computaiton, Experiment and Theory, July 1-2 2010, São José dos Campos, SP, Brasil, available at www.mec.ita.br/workshop2010/Pdf/Oddvar.pdf, last access December 2015.

[78] C. Farhat and M. Lesoinne, "Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelasti problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 182, pp. 499–515, 2000.

[79] 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*, vol. 190, p. 3147–3170, Mar. 2001.

[80] E. Longatte, V. Verreman, and M. Souli, "Time marching for simulations of fluid-structure interaction problems," *Journal of Fluids and Structures*, vol. 25, pp. 95–111, 2009.

[81] B. Miller, A. R. Crowell, and J. J. McNamara, "Loosely coupled time-marching of fluid-thermal-structural interactions," AIAA Paper 2015-1666, 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference. January 2015.

[82] J. Heeg and D. J. Piatak, "Experimental data from the benchmark supercritical wing wind tunnel test on an oscillating turntable," AIAA Paper 2013-1802, 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 8-11 Apr 2013; Boston, MA, United States.

[83] A. Jirásek, M. Dalenbring, and J. Navrátil, "Computational fluid dynamics study of benchmark supercritical wing at flutter condition," *AIAA Journal*, 2016. accessed August 30, 2016.

[84] "Aeroelastic prediction workshop 2," http://nescacademy.nasa.gov/workshops/aepw2/public/.

[85] P. R. Spalart and S. R. Allmaras, "A one-equation turbulence model for aerodynamic flows," *Recherche Aerospatiale*, vol. 1, pp. 5–21, 1994.

[86] A. Jirásek, M. Dalenbring, and J. Navrátil, "Numerical study of benchmark super-critical wing at flutter condition," in *54th Aerospace Sciences Meeting*, vol. 1551, 2016.

List of Symbols

$\mathbf{C}$	structural damping matrix
C_D	drag coefficient
C_L	lift coefficient
C_m	pitch moment coefficient
$\mathbf{F}$	vector of nodal forces
f	frequency
$\mathbf{F}_f$	vector of forces at fluid nodes
$\mathbf{F}_s$	vector of forces at structural nodes
$\mathbf{H}$	fluid-structure coupling matrix
i	internal energy
I_x	torsional constant
I_y	cross-section moment of inertia about y axis
I_z	cross-section moment of inertia about z axis
$\mathbf{K}$	stiffness matrix
$\mathbf{M}$	mass matrix
M	Mach number
N_f	basis function for the fluid
$\mathbf{n}_{n_i}$	control surface normal vector
N_s	basis function for the structure
p	pressure
Re	Reynolds number
S	area of surface
S_{Mi}	momentum source (i denotes x,y and z directions)
t	time
T	temperature
$\mathbf{u}$	vector of nodal displacements
$\mathbf{u}_f$	fluid nodes displacement vector
$\mathbf{u}_s$	structural nodes displacement vector
W	virtual work
α	angle of attack
μ	dynamic viscosity
ϕ_k	weighting function
Φ	dissipation function
ρ	density of fluid
θ	element rotation
ζ	damping ratio

List of Acronyms

BSCW	Benchmark Super Critical Wing
CFD	Computational Fluid Dynamics
CRM	Common Research Model
CSM	Computational Structural Mechanics
DOF	degree of freedom
FRF	frequency response function
NASA	National Aeronautics and Space Administration
NS	Navier-Stokes
RANS	Reynolds Averaged Navier-Stokes
RBF	radial basis function
RMS	root mean square
SA	Spalart-Almaras
SQP	Sequantial Quadratic Programming
TDT	Transonic Dynamic Tunnel
URANS	Unsteady Reynolds Averaged Navier-Stokes

Appendices

Appendix A

Matlab Code of Structural Solver

```
function [Unod,extras] = fem_solver(model,properties,mode)
% function [Unod,extras] = beam_fem3d(model,properties,mode)
%
% output: Unod is matrix of nodal dispalcements in global coordinate system
%                [u_i v_i w_i thetax_i thetay_i thetaz_i]
%         extras — structure with additional values, it is:
%                .K — unconstrained global stiffness matrix
%                .M — unconstrained global mass matrix
%                .Kp — constrained global stiffness matrix
%                .Mp — constrained global mass matrix
%                .v — displacement vector (constrained dofs not included)
%
% input: model, properties — for required fields refer to help for
%                            mode 1: beam_fem3d
%                            mode 2: fem3d_nastran
%
%          mode — 1: beam preprocessor and solver
%          mode — 2: stiffness matrix provided
%

Unod = [];
extras = [];

switch mode
    case 1
        [Unod,extras] = beam_fem3d(model,properties);
    case 2
        [Unod,extras] = fem3d_nastran(model,properties);
end


function [Unod,extras] = beam_fem3d(model,properties)

% function [Unod,extras] = beam_fem3d(model,properties)
%
% output: Unod is matrix of nodal dispalcements in global coordinate system
%                [u_i v_i w_i thetax_i thetay_i thetaz_i]
%         extras — structure with additional values, it is:
%                .K — unconstrained global stiffness matrix
%                .M — unconstrained global mass matrix
%                .Kp — constrained global stiffness matrix
%                .Mp — constrained global mass matrix
%                .v — displacement vector (constrained dofs not included)
%
% input: model — structure with following data
%          .nodes — is matrix of node coordinates and their ID [ID x y z],
%                   each row is one element
%          .forces - matrix of node loading [Fxi Fyi Fzi Mxi Myi Mzi]
%                    each row is one element
%          .constraints - is two column matrix, first column is node number,
%                         2nd is number of DOF,
%                         if some DOF is constrained = 1, otherwise 0
%                         DOF are [u v w thetax thetay thetaz]
%          .mass - if 'with_inertia' -> inertia are included to mass matr.
%                - if 'no_inertia' -> inertia are included to mass matr.
%                - (inertia = Iy, Iz, J )
%          .elements - beam elements definition [ID PID N1 N2]
%          .virtual_elements - beam elements definition [ID N1 N2]
%                              !!!!! N1 must be on real beam !!!!!!
%          .gravity_acceler - value of gravity accleration. If blank
%                             gravity is not taken into account
%        properties - structure with material and section properties
%          .E - Young's modulus
%          .nu - Poisson's ratio
%          .rho - material density
%          .A - is the cross-sectional area
%          .Asy,.Asz - the y-direction, the z-direction effective shear area
%                   if zero is given, then shear effect is not included
%          .Iy,.Iz - is the cross-section moment of inertia about the y, z axis
%          .J -  the polar moment of inertia
%          NOTE: for A, Asyi, Aszi, Iyi, Izi, Ji:
%                if a scalar is given => charact. is uniform along beam
%                if a vector is given => charact. is elementwise distributed
%                it is driven by Ai in this code!!
%          .orient - defines orientation of ye axis (cross section
%                            orientation) in global c.s
%          .nmod = number of eigen modes to extract
%

% ======== PREPROCESSOR
constr = model.constraints;

ndof = 6;                            % number of node degree of freedom
nele = size(model.elements,1);       % number of elements
nnod = nele + 1;                     % number of nodes / for 2 node element

% constrain definition
B=zeros(nnod,ndof);

% constr => two column matrix, first column is node number, 2nd is number
% of DOF

for i=1:size(constr,1)
    iB = find(model.nodes(:,1) == constr(i,1));
    jB = constr(i,2);
    B(iB,jB) = 1;
end

B = reshape(B',ndof*nnod,1);

% ===== SOLVER

[K,M,Nid] = global_matrices(model,properties);

% external forces
Fex = node_forces(model,properties,Nid);

% Inertial forces - gravity
Fi = node_inert_forces(model,size(Fex),nnod,ndof,M);
```

```
% sum of forces F
F = Fex + Fi;

% application of constraints to K and F => Kp and Fp
Kp = K;
Fp = F;

ind = find(B == 1);
Kp(ind,:)=[];
Kp(:,ind)=[];
Fp(ind)=[];

if exist('M','var')
    Mp = M;
    Mp(ind,:)=[];
    Mp(:,ind)=[];
end

w = Kp\Fp;

% including of constrained DOF into displacement vector
v=zeros(size(B));
ind= B==0;
v(ind,1)=w;

Unod = reshape(v,ndof,nnod)';

% "axial" displacements due to bending: in plane XY (dis. UV),
% beam is along X -> bending: in Unod I have V, I need U
% approximation is based on rotations

Unod = bending_trick(model,properties.orient,Unod);

% -------------------------------------------------------------------------
% transformation of displacements to "virtual nodes"

Unod = virt_node_displ(model,Unod);

% -------------------------------------------------------------------------
extras.K = K;
extras.M = M;
extras.Kp = Kp;
extras.Mp = Mp;
extras.dispvec = v;


end



function [Unod,extras] = fem3d_nastran(model,properties)

% function [Unod,eigendata,modal,extras] = beam_fem3d(model,properties)
%
% output: Unod is matrix of nodal dispalcements in global coordinate system
%               [u_i v_i w_i thetax_i thetay_i thetaz_i]
%
% input: model - structure with following data
%          .nodes - is matrix of node coordinates and their ID [ID x y z],
%                   each row is one element
%          .forces - matrix of node loading [Fxi Fyi Fzi Mxi Myi Mzi]
%                    each row is one element
%          .constrainsts - is matrix of constrainted DOF, each row is one
%                         node, if some DOF is constrained = 1, otherwise 0
%                         DOF are [u v w thetax thetay thetaz]
%
%        properties - structure with following fields:
%          .K - stiffness matrix
%          .M - mass matrix
%          .dof_ord - order of degrees of freedom in matrices
%                   - first column = node number, second column DOF
%                   - DOFs: 1,2,3 -> x-displ, y-displ, z-displ
%                           4,5,6 -> rotations about x, y, z
%

constr = model.constraints;
dof_ord = properties.dof_ord;

% ======== PREPROCESSING
nnod = size(model.nodes,1);

% constrain definition
B=zeros(size(dof_ord,1),1);

for i=1:size(constr,1)
    ind = find((constr(i,1) == dof_ord(:,1)) & constr(i,2) == dof_ord(:,2));
    B(ind) = 1;
end

% ===== PROCESSING

% global assembled stiffness and mass matrix K and M from NASTRAN

K = properties.K;
M = properties.M;

% Forces - external
Fex = node_forces_nastran(model,properties);

% Inertial - gravity
G = zeros(size(Fex));
if isfield(model,'gravity_acceler')
    ag = model.gravity_acceler;
    vec_ag = zeros(size(dof_ord,1),1);
    for i=1:3
       ind = find(dof_ord(:,2)==i);
       vec_ag(ind) = ag(i);
    end
    G = M*vec_ag;

end

% all forces F
F = Fex + G;

Kp = K;
Fp = F;
Mp = M;

% application of constraints to Kp and Fp
ind = find(B == 1);
Kp(ind,:)=[];
Kp(:,ind)=[];
Fp(ind)=[];
if exist('M','var')
    Mp = M;
    Mp(ind,:)=[];
    Mp(:,ind)=[];
end
w = Kp\Fp;

% including of constrained DOF into displacement vector
v=zeros(size(B));
ind = find( B==0 );
v(ind)=w;
Unod = zeros(nnod,6);

for i=1:nnod
    Nid = model.nodes(i,1);
   ind = find(dof_ord(:,1) == Nid);
   dofs = dof_ord(ind,2);

   Unod(i,dofs) = v(ind)';
end

extras.K = K;
extras.M = M;
extras.Kp = Kp;
extras.Mp = Mp;
extras.dispvec = v;

end



function [K,lam] = beam3d(A,Asy,Asz,E,nu,Iy,Iz,J,P1,P2,Coy,Coz)
%   BEAM3D
%
%   The routine provides the 12 x 12 stiffness matrix for a three
%   dimensional beam element in global coordinates. Shear effects
%   are included if the user specifies shear areas different from
%   zero. The formulation being used here is described in detail on
%   Przemieniecki, J.S., "Theory of Matrix Structural Analysis",
%   McGraw-Hill, 1968. The syntax is:
%              K = beam3d(A,Asy,Asz,E,nu,Iy,Iz,J,P1,P2,Coy,Coz)
%   where: A is the cross-sectional area;
%          Asy is the y-direction effective shear area;
%          Asz is the z-direction effective shear area;
%          E is the Young's modulus;
%          nu is the Poisson's ratio, and it is assumed:
%              G=E/(2*(1+nu));
%          Iy is the cross-section moment of inertia about the
%              y axis;
%          Iz is the cross-section moment of inertia about the
%              z axis;
%          J is the polar moment of inertia;
%          P1 and P2 are vectors of the {x,y,z} coordinates of the
%              nodes;
%          Coy is the vector of direction cosines of the local
%              y direction with respect to the global system of
%              coordinates, i.e., {loy,moy,noy};
%          Coz is the corresponding vector for the local
%              z direction, i.e., {loz,moz,noz};


L = norm(P2-P1);               % Find length of beam.

lox = (P2(1)-P1(1))/L;         % Find direction cosines and lambda matrix.
mox = (P2(2)-P1(2))/L;
nox = (P2(3)-P1(3))/L;
```

```
lam = [lox mox nox; Coy'; Coz'];
zerom = zeros(3);
lambda = [lam zerom zerom zerom
          zerom lam zerom zerom
          zerom zerom lam zerom
          zerom zerom zerom lam];

if Asy == 0,                            % Find shear−deformation parameters.
   phiy = 0;
else
   phiy = 24*(1+nu)*Iz/(Asy*(L^2));
end
if Asz == 0,
   phiz = 0;
else
   phiz = 24*(1+nu)*Iy/(Asz*(L^2));
end

G = E/(2*(1+nu));               % Create matrix in local coordinates.
kl = zeros(12);
kl(1,1)  = E*A/L;
kl(2,2)  = 12*E*Iz/((L^3)*(1+phiy));
kl(3,3)  = 12*E*Iy/((L^3)*(1+phiz));
kl(4,4)  = G*J/L;
kl(5,3)  = −6*E*Iy/((L^2)*(1+phiz));
kl(5,5)  = (4+phiz)*E*Iy/(L*(1+phiz));
kl(6,2)  = 6*E*Iz/((L^2)*(1+phiy));
kl(6,6)  = (4+phiy)*E*Iz/(L*(1+phiy));
kl(7,1)  = − kl(1,1);
kl(7,7)  = kl(1,1);
kl(8,2)  = −kl(2,2);
kl(8,6)  = −kl(6,2);
kl(8,8)  = kl(2,2);
kl(9,3)  = −kl(3,3);
kl(9,5)  = −kl(5,3);
kl(9,9)  = kl(3,3);
kl(10,4)  = −kl(4,4);
kl(10,10)  = kl(4,4);
kl(11,3)  = kl(5,3);
kl(11,5)  = (2−phiz)*E*Iy/(L*(1+phiz));
kl(11,9)  = −kl(5,3);
kl(11,11) = kl(5,5);
kl(12,2)  = kl(6,2);
kl(12,6)  = (2−phiy)*E*Iz/(L*(1+phiy));
kl(12,8)  = −kl(6,2);
kl(12,12) = kl(6,6);
for i = 1:11
   for j = i+1:12
      kl(i,j) = kl(j,i);
   end
end

K = lambda'*kl*lambda;            % Create matrix in global coordinates.


function M = mbeam3d(rho,A,Iy,Iz,Jx,P1,P2,Coy,Coz)
% function M = mbeam3d(rho,A,Iy,Iz,Jx,P1,P2,Coy,Coz)
%   MBEAM3D
%
%   This routine provides the 12 x 12 mass matrix for a three
%   dimensional beam element in global coordinates. The effects of
%   shear deformations on the beam element are not included in the
%   present formulation. On the other hand, effects of rotary
    %   inertia are included through the terms that have the moments of
    %   inertia Iy and Iz, as well as torsional inertia effects are
    %   included through the terms with the polar moment of inertia
    %   Jx. If the user does not desire to include such effects, he
    %   can simply specify these properties as being zero. The
    %   formulation being used here is described in detail on
    %   Przemieniecki, J.S., "Theory of Matrix Structural Analysis",
    %   McGraw-Hill, 1968 (Section 11.5, pp. 292-297).
    %   The syntax is:
    %              M = mbeam3d(rho,A,Iy,Iz,Jx,P1,P2,Coy,Coz)
    %   where: rho is the material density;
           %          A is the cross-sectional area;
           %          Jx is the polar moment of inertia;
           %          Iy is the cross-section moment of inertia about the
           %              y axis;
           %          Iz is the cross-section moment of inertia about the
           %              z axis;
           %          P1 and P2 are vectors of the {x,y,z} coordinates of the
           %              endpoints;
           %          Coy is the vector of direction cosines of the local
           %              y direction with respect to the global system of
           %              coordinates, i.e., {loy,moy,noy};
           %          Coz is the corresponding vector for the local
           %              z direction, i.e., {loz,moz,noz};
           %

           L = norm(P2-P1);               % Find length of beam.

           lox = (P2(1)-P1(1))/L;         % Find direction cosines and lambda matrix.
           mox = (P2(2)-P1(2))/L;
           nox = (P2(3)-P1(3))/L;
lam = [lox mox nox; Coy'; Coz'];
zerom = zeros(3);
lambda = [lam zerom zerom zerom
          zerom lam zerom zerom
          zerom zerom lam zerom
          zerom zerom zerom lam];

ml = zeros(12);                    % Create matrix in local coordinates.
ml(1,1) = 1/3;
ml(2,2) = 13/35 + 6*Iz/(5*A*L^2);
ml(3,3) = 13/35 + 6*Iy/(5*A*L^2);
ml(4,4) = Jx/(3*A);
ml(5,3) = -11*L/210 - Iy/(10*A*L);
ml(5,5) = (L^2)/105 + 2*Iy/(15*A);
ml(6,2) = 11*L/210 + Iz/(10*A*L);
ml(6,6) = (L^2)/105 + 2*Iz/(15*A);
ml(7,1) = 1/6;
ml(7,7) = 1/3;
ml(8,2) = 9/70 - 6*Iz/(5*A*L^2);
ml(8,6) = 13*L/420 - Iz/(10*A*L);
ml(8,8) = ml(2,2);
ml(9,3) = 9/70 - 6*Iy/(5*A*L^2);
ml(9,5) = -13*L/420 + Iy/(10*A*L);
ml(9,9) = ml(3,3);
ml(10,4) = Jx/(6*A);
ml(10,10) = ml(4,4);
ml(11,3) = -ml(9,5);
ml(11,5) = -(L^2)/140 - Iy/(30*A);
ml(11,9) = -ml(5,3);
ml(11,11) = ml(5,5);
ml(12,2) = -ml(8,6);
ml(12,6) = -(L^2)/140 - Iz/(30*A);
ml(12,8) = -ml(6,2);
ml(12,12) = ml(6,6);
for i = 1:11
   for j = i+1:12
      ml(i,j) = ml(j,i);
   end
end
mlocal = rho*A*L*ml;

M = lambda'*mlocal*lambda;            % Create matrix in global coordinates.
```

Appendix B

Computational Aeroelasticity Tool - Convergence Tests

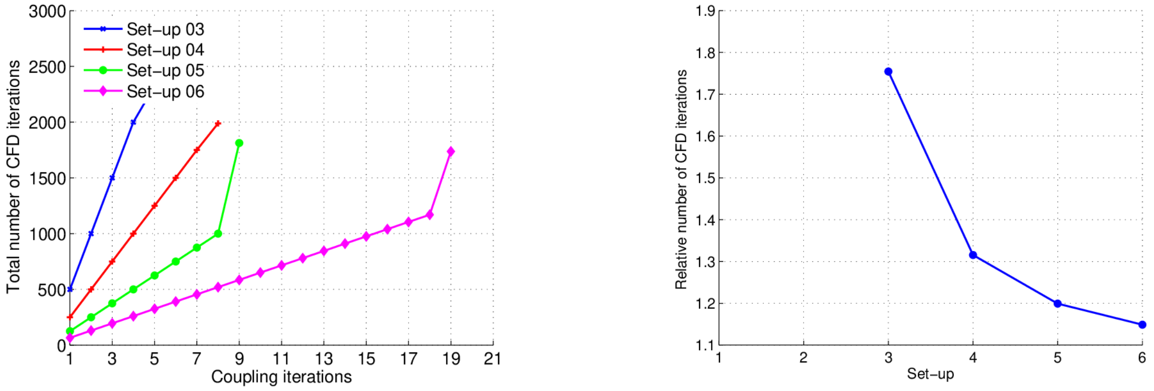


Figure B.1: Computational cost of the aeroelastic solution - influence of the Edge CFD solver settings, Euler simulation at $M = 0.6$, $\alpha = 5°$

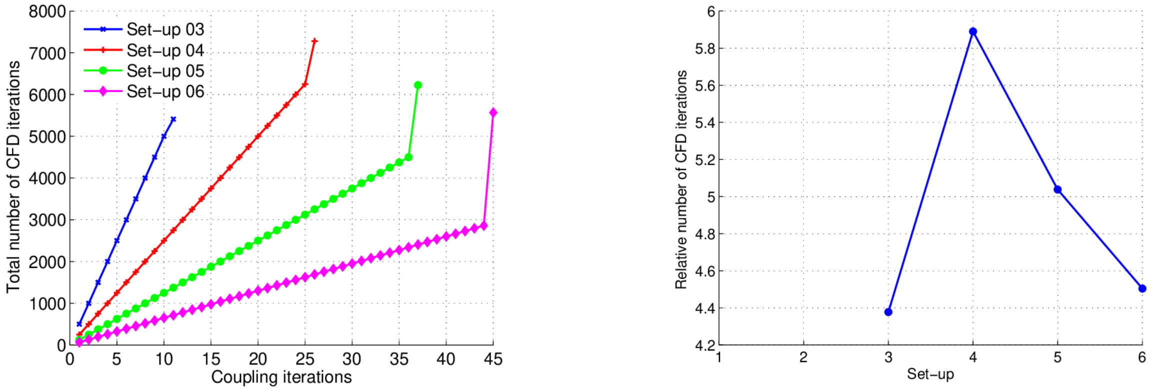


Figure B.2: Computational cost of the aeroelastic solution - influence of the Edge CFD solver settings, Euler simulation at $M = 0.88$, $\alpha = 0°$

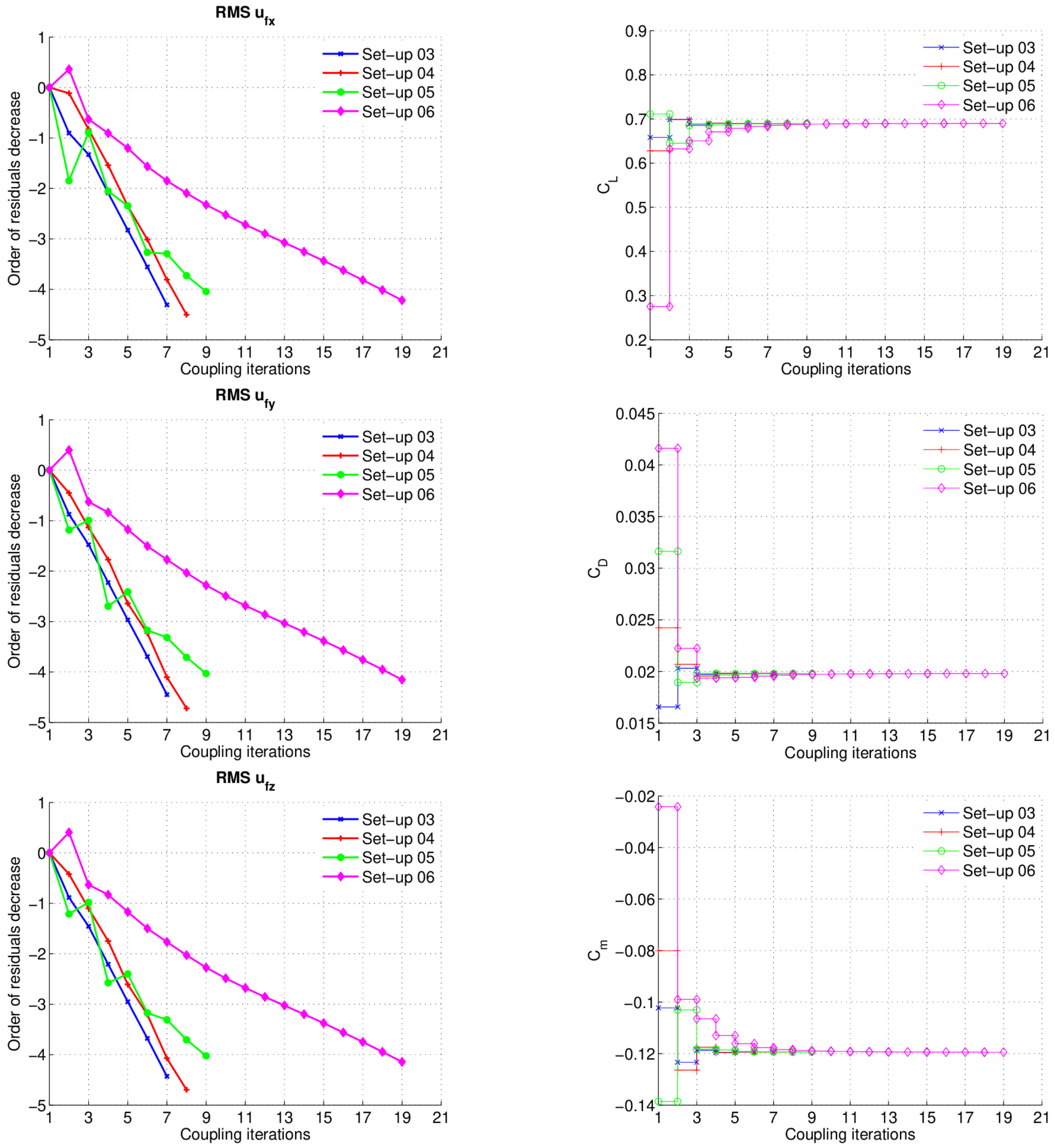


Figure B.3: Convergence of the aeroelastic solution - influence of the Edge CFD solver settings, Euler simulation at $M = 0.6$, $\alpha = 5^\circ$

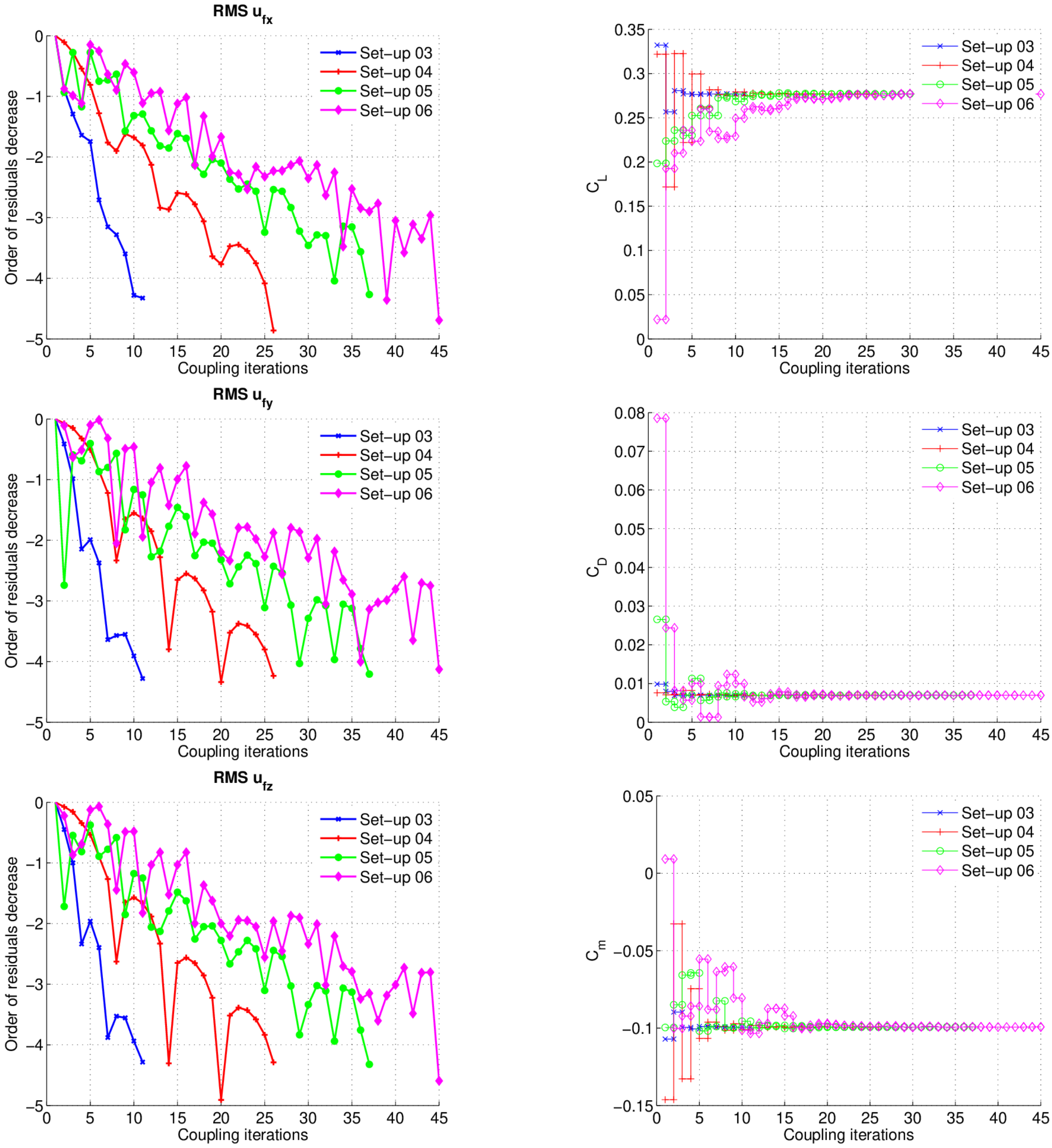


Figure B.4: Convergence of the aeroelastic solution - influence of the EDGE CFD solver settings, Euler simulation at $M = 0.88$, $\alpha = 0°$