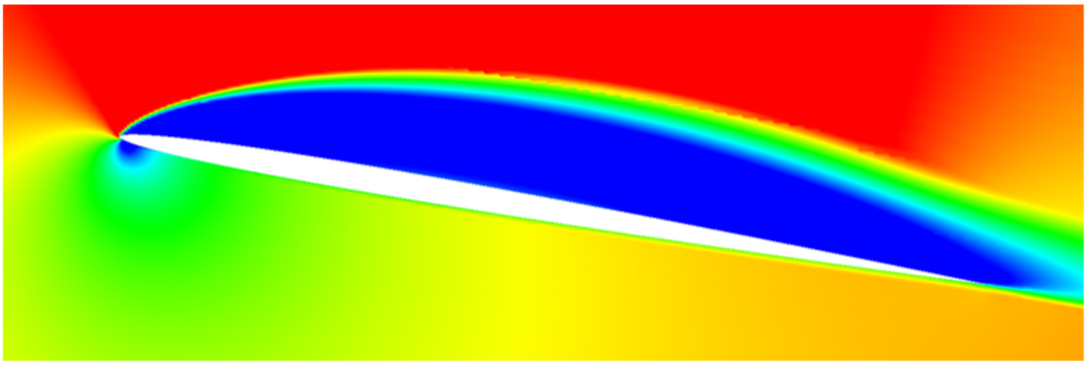
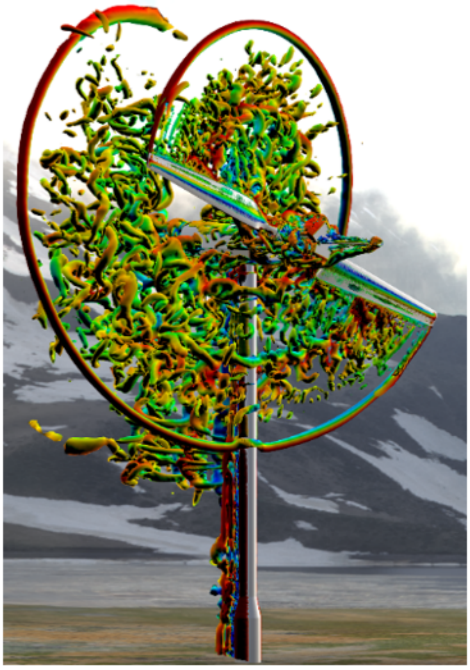
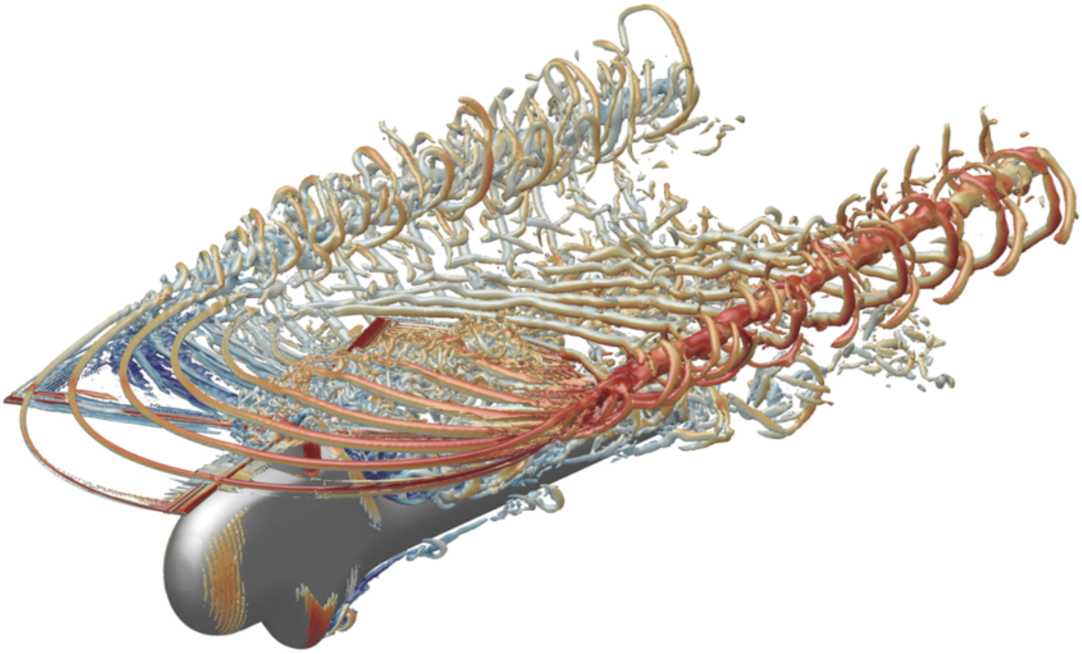
A fluid-multibody coupling for aeroelastic applications to rotorcraft and wind turbines

Umberto Boatto

Master of Science Thesis







Delft University of Technology

Challenge the future

A fluid-multibody coupling

for aeroelastic applications to rotorcraft and wind turbines

by

Umberto Boatto

to obtain the degree of Master of Science at the Delft University of Technology

and to be defended publicly on Tuesday February 18th, 2020

Wind Energy Group, Faculty of Aerospace Engineering, Delft University of Technology

Dassault Systèmes

Student number:	4728165	
Thesis Committee:	Prof. dr. D. (Damiano) Casalino	TU Delft, Chairperson
	Dr. ir. A. (Axelle) Viré	TU Delft, Supervisor
	Dr. ir. A. (Sander) van Zuijlen	TU Delft, Examiner
	G. (Gianluca) Romani MSc	Dassault Systèmes, Supervisor

Cover pictures taken from:

Helicopter bottom-right: Romani G., Casalino D., Application of Lattice-Boltzmann Method to Rotorcraft Aerodynamics and Aeroacoustics, 43rd ERF, 2017.

Wind turbine bottom-left: Perot F., Kim M.-S., Meskine M., NREL Wind Turbine Aerodynamics Validation and Noise Predictions using a Lattice Boltzmann Method, 2012, AIAA 2012-2290

Delft University of Technology

Abstract

Accurate aerodynamic and aeroacoustic simulations of helicopters and wind turbines require the inclusion of the blade elasticity into the computational setup. In the case of a rotorcraft, this is required to correctly capture the interactions between the complex rotor aerodynamics and the flexible blade structure occurring during maneuvers and forward flight. In the case of wind turbines, the main motivation is that the lightweight large blades often operate under turbulent inflow conditions due to the effect of other wind turbines, surrounding environment and blade tower, which can trigger aeroelastic phenomena potentially leading to a catastrophic system failure.

In the case of coupled fluid-structure simulations, it is frequent to employ low-order aerodynamic models for the computation of the aerodynamic forces to provide to the Multi-Body Dynamic (MBD) or Finite-Element (FE) solvers for the structural domain. If this approach can be efficient for optimization purposes, parametric studies and the simulation of maneuvers or critical operating conditions, it is often insufficient in predicting the involved flow phenomena responsible for structural vibrations and noise, namely the development of wake and tip vertices, dynamic stall and transonic flows. For this purpose, high-fidelity Computational Fluid Dynamics (CFD) solvers can be exploited for more accurate simulations in the context of detailed design, although this comes at a significantly increased simulation cost, which obstacles the application of these tools to industrial problems. Part of this cost can be recovered by coupling such CFD solvers with MBD tools - capable of modelling flexible bodies - by means of reduced-order one-dimensional FE models in place of expensive three-dimensional FE simulations.

Within this framework, this thesis focuses on the development and verification of a coupling methodology between the Lattice-Boltzmann CFD solver PowerFLOW® and the MBD tool Simpack® - both products belonging to the Simulia® simulation portfolio of Dassault Systèmes - for pitching and plunging airfoils featuring lumped structural parameters under flow conditions compatible with the reference analytical models (incompressible, high Reynolds number and attached flows). This is achieved by firstly verifying the multibody and fluid simulation-setups separately for prescribed aerodynamic forces in the first case, while input harmonic pitching and plunging motions are used in the second. The verification of the MBD setup, returns a perfect match with the reference kinematic states for the same analytical aerodynamic models employed. On the other hand, the CFD setup provides a very good comparison with the theoretical lift and aerodynamic moment prediction only for a truly pitching airfoil, while the equivalent approaches for virtual plunging (flow blowing/suction) and pitching (body forces applied to fluid volumes near the airfoil) return less favourable matches, depending on the motion conditions employed. Next, a coupling methodology is firstly developed for a reduced-order aerodynamic routine and Simpack®, which is positively compared with analytical aeroelastic solutions, confirming its correct working and providing high accuracy after tuning the coupling timestep of the underlying explicit fluid-structure interaction algorithm. Once the methodology is extended to PowerFLOW®, the final CFD-MBD coupling is tested for several amplitudes and reduced frequencies of motion always against theoretical aeroelastic predictions, returning a good match for small-amplitudes and slow motion conditions. In addition, the different approaches to model the airfoil pitching motion (rotating mesh and body forces) in the fluid solver are assessed and compared favourably against each other. As a conclusive effort, the coupling is applied to a bi-dimensional airfoil flutter case, returning a prediction of the flutter velocity within 1% difference with respect to analytical methods.

Preface

This thesis represents the last step of my path as a student in TU Delft, and I believe that the months spent on this project helped me a lot to grow from the professional and personal points of view. All this could have not been possible without the help of all the people that supported me both in Delft and in Stuttgart, and here I want to express all my gratitude to them.

First of all, I want to thank my daily supervisor in Dassault Systèmes Gianluca Romani not just for his technical support and patience, but also for his tips on doing research and, even more importantly, for the countless life-lessons I have learnt from his example. At the same time, I want to acknowledge Prof. Damiano Casalino, as my manager in Dassault Systèmes for his precious advice, work-ethics and insights in the engineering profession, as well as Dr. Axelle Viré and Dr. Sander van Zuijlen for the valuable suggestions provided during these months and for being part of the thesis committee. An important part of my gratitude goes also to the PowerFLOW® team in Stuttgart, in particular Dr. Wouter van der Velden, and the Simpack® team in Gilching, in particular Tom Burton, James Clarke and Sadiq Huq, for their support with the tools and to better understan the context of my project.

I also want to say thanks to Florian Maier for the illuminating discussions on software architecture and scientific research, as well as to my colleagues and friends both in Stuttgart and Delft for the time spent together while studying, working and enjoying our free-time. Finally and foremost, a sincere "thank you" is for my parents, sister and grandparents for always being present and encouraging me in the hard as well as good times.

Umberto Boatto
Delft, February 2020

Contents

List of Figures

List of Tables

List of Acronyms

AC Aerodynamic Center

ALE Arbitrary Eulerian Lagrangian

AoA Angle of Attack

BEMT Blade Element Momentum Theory

BRF Body Reference Frame

CFD Computational Fluid Dynamics

CLI Command Line Interface

CoG Center of Gravity

CSD Computational Structural Dynamics

csv comma separated values

DAE Differential Algebraic Equation

DoF Degree of Freedom

EoM Equation of Motion

FE Finite Element

FSI Fluid Structure Interaction

GUI Graphic User Interface

HAWT Horizontal Axis Wind Turbine

LE Leading Edge

LES Large Eddy Simulation

LBM Lattice Boltzmann Method

lhs left hand side

MBD Multi-Body Dynamic

ODE Ordinary Differential Equation

PDE Partial Differential Equation

RANS Raynolds Averaged Navier Stokes (Equations)

rhs right hand side

TE Trailing Edge

VR Variable Resolution

1
Introduction

This chapter serves as introduction to the project thesis and it starts, in sec.1.1, with an overview of the main simulation methods for rotorcraft and wind turbines aeroelasticity by referring to the related available literature. In addition, also the motivations for pursuing the main thesis goal, which is a fluid-multibody coupling to target such applications, are provided in the first section of this chapter. Next, the research objective and questions are outlined in sec.1.2, while the main research method/approach employed is discussed in sec.1.3. Eventually, sec.1.4 defines the structure of this report.

1.1. Project framework and motivations

Rotary-wing aircraft have always been characterized by aeroelastic phenomena since the early developments during the second world war, and most of these phenomena are due to the complex interactions between rotor aerodynamics - where transonic, highly unsteady, 3D, stalled and reversed flows can simultaneously appear - and the lightweight flexible structure of the blades [9], [7]. Regarding horizontal-axis wind turbines (HAWT), the necessity to increase their power-harvesting purpose has led in the last decades to larger and larger rotor diameters, resulting in an ever increasing importance of the elastic behaviour of the slender flexible blades and their interaction with the random, 3D and unsteady wind inflow conditions [10], [11], [12]. Therefore, in the context of designs based on computational simulations, accurate aeroelastic simulations of rotorcraft and large HAWT require to account for the flexibility of the structure when computing the aerodynamic forces exerted over the rotor blades. This can be obtained by solving a fully coupled aeroelastic problem where aerodynamic, elastic and inertial rotational forces (centrifugal and Coriolis) are simultaneously acting on the rotor blades.

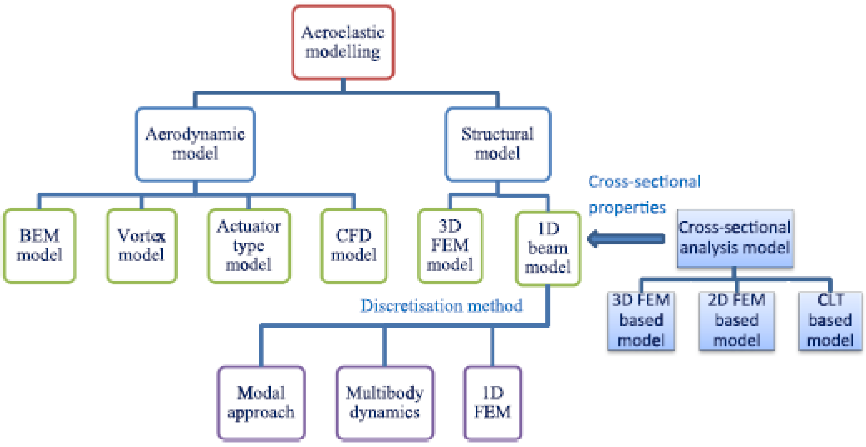


Figure 1.1: Overview of the possible simulation methods for aeroelastic applications in the context of HAWT [1], but also applicable to rotorcraft in general.

Coupled fluid-structure simulations for rotorcraft and HAWT can be performed by means of a plethora of methods, as visible in fig.1.1. As the figure clearly shows, dedicated aerodynamic and structural models are employed with multiple levels of fidelity available. Concerning the aerodynamic part of the problem, this can be simulated by using, in order of fidelity, Blade Element Momentum Theory (BEMT), vortex and actuator type models, as well as CFD. If BEMT is appreciated for its reduced cost and simplicity in the implementation, it requires many semi-empirical corrections based on wind turbine or rotorcraft applications [1] (for instance tip and root losses, turbulent wake states, dynamic stall) which require the tuning of their parameters to provide acceptable results. On the other hand, CFD is much more computationally expensive, although it can return more accurate predictions of blade loading and rotor wake convection due to the higher fidelity in the physical modelling [12].

Focusing on the structural side of the problem, two main approaches are customary, either based on Multi-Body Dynamic (MBD) or Finite Element (FE) simulations. The first ones focus more on the overall system, employing a higher level of abstraction and a limited number of Degrees of Freedom (DoF). In contrast, the latter family takes more care of the body material and geometric details, with a much large number of unknowns defined on top of surface meshes. FE methods can be much more accurate than MBD ones, especially in the case of 3D-shell formulations, but rather expensive and often too sophisticated, making them not the most natural modelling choice for propeller/blade elasticity [1]. A very good compromise is offered by modern MBD tools enhanced with flexible-body capabilities by means of reduced-order linear or non-linear 1D-beam FE formulations, and among the others, Simulia Simpack® is a quite popular MBD solver used in the wind energy and rotorcraft community [13].

After describing methods for the separate fluid and structure domains, the two main approaches of coupled fluid-structure simulation methods can be presented. When it comes to conventional design practices, rotorcraft development still relies on low-fidelity engineering comprehensive codes. These are specialized multi-physics tools featuring reduced order models for the aerodynamics and structure, and are employed to simulate complex transient maneuvers, to perform rotor trimming, as well as for optimization purposes and parametric studies thanks to their low computational cost, although they often provide poor accuracy when predicting complex phenomena (high frequency impulsive noise and air-loads) [14], [15]. Regarding wind turbine design, similar low-fidelity and cost-competitive tools are available and employed in full-scale aeroelastic design, as described in [1].

If engineering comprehensive codes can be sufficiently reliable for the structural dynamics part of the problem, their aerodynamic modules are frequently inadequate to capture the complex flow dynamics characterizing both rotorcraft and HAWT. Therefore, advanced simulation-based design practices require the coupling of high fidelity CFD tools with Computational Structural Dynamic (CSD) methods to improve predictions of aerodynamic phenomena. However, this comes at the price of a much larger simulation cost, which limits their applicability to complex maneuvers, rotor trimming, optimization and parametric studies. An extensive review of these CFD-CSD couplings for rotor-crafts can be found in [16], while in the case of HAWT, valuable examples are [17] and [18].

Even when high-fidelity CFD simulations are at stake, further distinctions can be made. Most of industrial CFD practises are based on steady or unsteady Reynolds-Averaged formulations of the Navier-Stokes equations (RANS), which model the effect of turbulent structures on the space-time solution of the mean flow. This approach is moderately expensive, but limited by the calibration issues of the turbulence models and it can not adequately predict the complex dynamics of stalled flows at high angle of attach (AoA) occurring in HAWT [1]. More accurate methods are Large Eddy Simulations (LES), where the space-time behaviour of the large scales of turbulence are solved, and only the effect of sub-grid turbulence - which tends to have a universal behaviour - is modelled [12]. The major drawback of LES is the tremendous cost for wall-bounded flows (unless wall models are employed) and their applicability with industrial turnaround times is limited to simple geometries. A compromise between the two above mentioned approaches are Detached Eddy Simulations[1], but further research efforts are required to completely setting up this methodology for industrial applications. Given the deficits of traditional CFD methods, alternative fluid simulation methods, for example the Lattice Boltzmann Method (LBM) can be considered. The LBM, and in particular the commercial solver Simulia PowerFLOW®, provide inherently unsteady and compressible flow solutions comparable to LES, in terms of accuracy, with the advantage of low numerical dissipation, automatically generated Cartesian volume meshes and highly efficient parallelization, all aspects that make this method a valuable alternative to LES for industrial problems featuring complex geometries, as well as aeroacoustic and thermal simulations [19].

Based on the previously outlined framework and literature overview, the main motivation for developing a fluid-multibody coupling involving a LBM-based CFD solver and a MBD simulation tool with flexible body enhancements consists in contributing to the development of high-fidelity aeroelastic

[1] A hybrid approach between RANS near the wall to model sub-grid structures and LES to resolve the engineering-relevant flow dynamics far from the wall [1].

simulations that can be competitive and exploitable also by the industry. Such high-fidelity simulations - based on fully-coupled Fluid-Structure-Interaction (FSI) algorithms - aspire to become a reliable and useful tool for designers of rotorcraft and wind-turbines to investigate complex aeroelastic and acoustic phenomena, where the prediction of the aerodynamics is further improved by including the flexibility of the structure. However, due to the time constraints of a master graduation project, this thesis limits only to the development and verification of a coupling methodology for 2D airfoil problems. Therefore, the actual goal of this thesis is to assess whether the coupling can return aeroelastic quantities as close as possible to analytical reference solutions, in their range of applicability, and to investigate the possibilities and limitations of both PowerFLOW® and Simpack® in modelling such problems.

1.2. Research objectives and questions

Within the previously outlined framework and literature, as well as motivations, the objective of this master thesis research project can then be formulated as it follows:

To develop an aeroelastic computational methodology consisting in a coupling between a CFD tool (Simulia PowerFLOW®) and a MBD solver (Simulia Simpack®) by independently verifying the simulation setup of each tool on its own domain of interest, assessing the correct working and tuning the coupling FSI algorithm, as well as by testing the accuracy of the final coupling for a range of conditions.

Given the above research objective, the following main research question can be formulated:

How to develop a methodology to couple PowerFLOW® and Simpack® by accounting for the possibilities and limitations offered by the two solvers and how accurately can such a coupling predict the aeroelastic solutions provided by analytical models?

The previous main research question can be further elaborated into the following four subquestions:

- *Which are the available FSI algorithms to couple fluid and structural dynamics tools, what are the issues/aspects to take care of and what is the most appropriate one, given the physical phenomena of interest and the target applications?*
- *What are the methods that can be exploited to model pitching and plunging airfoil motions in the PowerFLOW® CFD setup and to what extent are these approaches suitable for FSI?*
- *What are the possible approaches to model aerodynamic forces in the MBD Simpack® setup, in the context of a coupling with an external tool?*
- *Independently of the FSI algorithm, what are the possible coupling strategies and media to couple PowerFLOW® and Simpack® based on the capabilities and limitations of both solvers, software aspects and feasibility constraints?*

Part of these questions have been answered during the literature study activity performed at the beginning of this research project, and the outcome of this activity constitutes the background knowledge blocks shown on the left side of fig.1.2. On the other hand, another part of these questions have been answered by the work carried out during this thesis by following a precise methodology, as discussed in the next section.

1.3. Research approach/method

The combination of the previous research objective and questions prescribes a precise working methodology. This consists in a step-by-step approach where each activity needs to be carefully verified before moving to the following step and to reach the final coupling methodology between Simpack® and PowerFLOW®. Such a sequential approach is described in the research approach diagram in fig.1.2. In the diagram, the required know-how blocks on the left side are essential to achieve all the result blocks on the right side. The latter follow one after the other, namely to achieve the final CFD-MBD coupling, it is necessary to first verify the simulation setups of the MBD and CFD solvers independently. This means that both the kinematic states computed by Simpack® under prescribed airload components and the PowerFLOW® aerodynamic forces for prescribed motions need to be verified before the coupling can be designed. Besides, the coupling design requires an intermediate step, where the complexity of

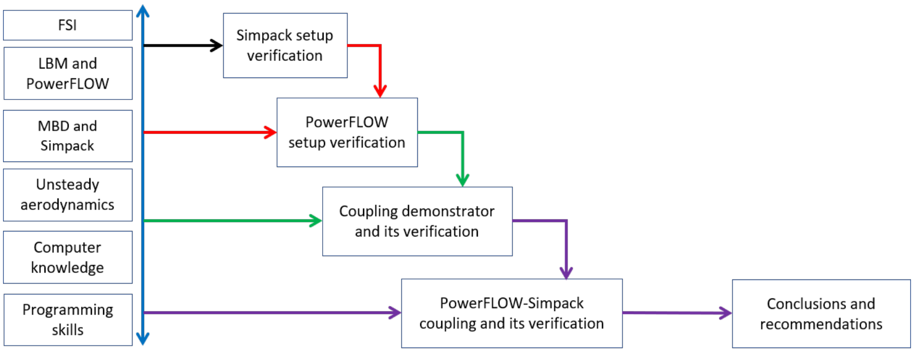

FSI
LBM and PowerFLOW
MBD and Simpack
Unsteady aerodynamics
Computer knowledge
Programming skills
Simpack setup verification
PowerFLOW setup verification
Coupling demonstrator and its verification
PowerFLOW-Simpack coupling and its verification
Conclusions and recommendations

Figure 1.2: Research approach/method of this thesis project.

the CFD-LBM solver is replaced by a surrogate code based on analytical models (coupling demonstrator block in fig.1.2) for the sake of the assessment, as well as tuning of the FSI algorithm and coupling methodology. Only at the end, the actual PowerFLOW®-Simpack® coupling can be investigated under various motion and inflow conditions to understand its limitations, as well as to draw conclusions and give recommendations for further work.

1.4. Report structure

The structure of this report follows the logical scheme shown in fig.1.2. Therefore, after this introductory chapter, a description of the background knowledge required to achieve the research project goals is presented in ch.2. In ch.3, the steps carried out to verify the MBD setup for prescribed aerodynamic forces from analytical models are provided. Next, in ch.4, the CFD setup is described and verified for prescribed harmonic pitch and plunge motions against the analytical prediction. The following ch.5 deals with an extensive explanation of the design process and features of the CFD-MBD coupling. Finally, in ch.6, the aeroelastic results of the coupling between PowerFLOW® and Simpack® are presented and discussed.

2
Background

The goal of this chapter is to serve as background for the whole thesis project by covering the most important aspects and topics characterizing the research. To begin with, FSI is discussed in sec.2.1, then the LBM fluid model and the PowerFLOW® flow solver are presented in sec.2.2. Next, the MBD theory for rigid bodies and the main features of the Simpack® tool are discussed in sec.2.3. Finally, a general introduction to unsteady aerodynamic phenomena characterizing wind turbines and helicopters and a more detailed description of a few analytical unsteady aerodynamic models can be both found in sec.2.4.

2.1. Fluid-Structure Interaction methods

FSI studies phenomena where the overall behaviour of a body is influenced by the mutual interactions between fluid forces and the elasticity of the material itself [20]. Based on this definition, FSI can be seen as a generalization of aeroelasticity, as it does not limit to typical aeroelastic applications (turbomachinery, fixed wing and civil engineering) but FSI methods can be applied to parachutes, hemodynamics and sound generation in blow musical instruments [21]. This wide range of FSI applications, corresponds to an equally wide range of phenomena investigated, from the small fully-linearized oscillations of vibro-acoustics to the large non-linear motions of aerospace systems affected by complex flow phenomena, like transonic shock-boundary layer interaction [2]. Given the significant recent improvements in computing power and numerical methods, FSI is primarily a computational subject. Therefore, FSI experts and researchers are currently focusing on the development of either comprehensive numerical methods to solve simultaneously fluid and structure equations, or accurate, robust and efficient couplings between CFD and structural solvers. Both methods are relevant for industrial applications when it comes to simulation-based design approaches where system performance is assessed at a global multi-physics level to develop an engineering product. According to [22] and [23], FSI is, in general, characterized by four cornerstones, such as the fluid domain/solver, structural domain/solver, time stepping approach (monolithic or staggered) and interface treatment (matching the different structure and fluid mashes and fluid mesh deformation). The first two aspects are discussed in detail in sec.2.2 and sec.2.3 of this chapter, while the last two are the focus of this section, which starts by defining the FSI problem in terms of its fundamental governing equations for the flow, structure and fluid-solid interface.

2.1.1. The FSI problem

A FSI problem can be seen as a "three-field problem" for the fluid, structure and interface domains combined together [3]. Regarding the fluid flow domain, its behaviour is often described by employing the ALE formulation[1] of the Navier Stokes equations. This is shown below in their semi-discrete finite-volume approximation:

$$\frac{d}{dt}\Big[\boldsymbol{A}\cdot\boldsymbol{W}\Big]+\boldsymbol{\Phi}_c(\boldsymbol{W},\boldsymbol{x},\dot{\boldsymbol{x}})=\boldsymbol{R}(\boldsymbol{W},\boldsymbol{x}) \tag{2.1}$$

In eq.2.1, $\boldsymbol{W}$ is the vector of the conservative flow quantities (unknowns) and $\boldsymbol{A}$ is the matrix resulting from the finite-volume discretization of the domain. Besides, $\boldsymbol{\Phi}_c$ is the vector of the convective fluxes in the ALE representation, thus it takes into account also the convection due to the volume mesh motion. Finally, $\boldsymbol{R}$ is the vector of diffusive fluxes, while $\dot{\boldsymbol{x}}$ and $\boldsymbol{x}$ are respectively the velocity and the position

[1] The Arbitrary Eulerian Lagrangian formulation is a method of writing equations for a moving material when the observer moves in space with a velocity which differs from the one of the material. The ALE formulation is used both in solid and fluid mechanics, and in the latter case it is also called dynamic mesh formulation.

vectors of the moving frame with respect to the fixed one. More details on these terms are reported in [2] and [3]. When it comes to the structure domain, a discretized finite-element representation of the flexible body is employed, as shown in eq.2.2, consisting in a finite set of generalized coordinates $\boldsymbol{q}_i$ by means of the diagonal mass $\boldsymbol{M}$ and stiffness $\boldsymbol{K}$ matrices, the damping $\boldsymbol{D}$ matrix, the external fluid-force element $\boldsymbol{F}_{fld}$ coming from the eq.2.1, and a force vector of external loads $\boldsymbol{F}_{ext}$:

$$\boldsymbol{M}\,\ddot{\boldsymbol{q}} + \boldsymbol{D}\,\dot{\boldsymbol{q}} + \boldsymbol{K}\,\boldsymbol{q} = \boldsymbol{F}_{fld} + \boldsymbol{F}_{ext} \tag{2.2}$$

Besides, equations for the fluid-structure interface are required, as the interface acts as a pseudo-structure enforcing the deformation of the fluid volume mesh so that it can follow the structural deformation. The interface equation can be the same as eq.2.2, but usually simplified formulations with just fictitious stiffness matrices are employed to avoid additional unsteady motions in the mesh [8]. More details on the interface equations can be found in [3]. Finally, initial and boundary conditions need to be provided and the most natural choice is to employ the Dirichlet-Neumann conditions prescribing the continuity of stresses (equilibrium condition) and the continuity of displacements (compatibility condition) [2] and [3].

2.1.2. Time stepping approaches

In the previous section, the FSI problem was mostly discussed in spatial terms, here its solution in time is discussed by means of two main approaches, either the monolithic or the staggered/partitioned one. The previous distinction deals with the solution of the algebraic system resulting from the space-time discretization of the governing equations. In the monolithic approach, the structure and fluid domains are discretized together and solved simultaneously as a unique block, while in the staggered one, different tools/solvers are used to solve the structure and fluid equations separately.

Monolithic FSI

Due to the simultaneous treatment of the fluid and structure equations, this method leads very easily to large and cumbersome systems of equations. This occurs even for simple problems, as in general monolithic FSI methods are hard to implement and to solve, due to the conditioning of the overall system matrix [24]. Furthermore, monolithic FSI is also intrinsically inefficient and it does not allow to use dedicated methods and setups for the flow and the structure domains. A relevant example is that monolithic FSI forces to employ the smallest timestep between the two, and in typical engineering applications the difference can be of orders of magnitude. Therefore, the employment of staggered FSI and separate solvers allows not only to use different and optimized timesteps, but also different time marching schemes, namely implicit ones with large timesteps for the structure, and explicit schemes with small timesteps for the fluid domain [4]. Regarding practical aspects, there are also other reasons to prefer staggered FSI to the monolithic one. One of them is the impossibility of using dedicated solvers/methods, leading to loss of modularity and preventing from recycling existing, validated and specialized tools for the separate domains. For these reasons, monolithic approaches are not so popular and few examples can be found in literature and commercial codes, primarily on simple geometries, and where fluid and structure scales are limited and compatible [3]. Although there are many motivations for preferring staggered FSI methods to the monolithic approach, the latter has the advantage of being very little influenced by those stability issues affecting staggered schemes. More in detail, the interface equation might not be relevant for monolithic FSI, due to the simultaneous treatment of fluid and structure equations with similar meshes, whereas, in the case of staggered FSI, the interface is the location where the two domains are coupled by means of boundary conditions and the interface stiffness equations. Therefore, such a interface treatment aspect affecting staggered FSI might lead to instabilities due to the merge of data obtained with different space-time resolutions and methods, as well as to the strategy of coupling in time the two solvers. A discussion on the errors caused by such instabilities affecting staggered FSI can be found in [8].

Tightly-coupled staggered schemes

In staggered FSI schemes, the compatibility of stresses and displacements at the interface has to be satisfied instantaneously in time and continuously in space. Therefore, following the approach in [25], fluid forces $\boldsymbol{P}_{n+1}$ are functions of the structural displacements $\boldsymbol{u}_{n+1}$ at the same time step t_{n+1}, and such structural displacements are functions of fluid forces evaluated at the same time step likewise:

$$\boldsymbol{P}_{n+1} = f(\boldsymbol{u}_{n+1}) \qquad \text{and} \qquad \boldsymbol{u}_{n+1} = g(\boldsymbol{P}_{n+1}) \qquad \text{at} \quad t_{n+1} \tag{2.3}$$

Eq.2.3 naturally leads to a time-implicit formulation of the coupling, where equations are usually solved by means of iterative methods during each timestep. This is shown in fig.2.1 where the initial guess for the forces computed with the displacements from the previous time step is improved by repeating the coupling cycle (subiteration) until the difference between two subsequent force values is below a certain threshold. The previously described approach - called block Gauss Seidel iteration scheme [26] - is one of the most simple tight/strong coupling schemes to implement, although it can suffer from poor convergence [21]. In general, the disadvantage of this method is that the simulation cost is very likely to be higher, but a larger coupling timestep can be used, which, combined with an efficient iterative method (such as semi-implicit, fully implicit, predictor-corrector as discussed in [27]), can return much more accurate results for a moderate increase of the simulation cost. However, given the PowerFLOW® explicit formulation in time and the impossibility to repeat the computation of the solution for an arbitrary time step upon request, implicit FSI schemes are not further investigated and employed in this thesis project. Finally, more information regarding fundamental aspects like under/over-relaxation, convergence criteria and appropriate tolerances can be found in [28].

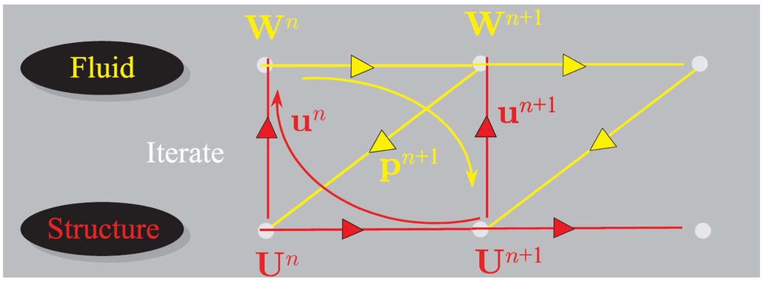


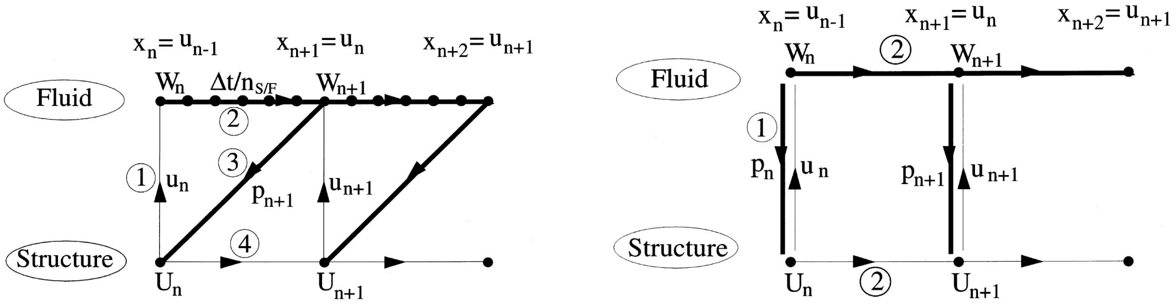
Figure 2.1: Iterative version of the CPS scheme according to [2].

Loosely-coupled staggered schemes

Contrarily to tightly-coupled schemes, loose couplings approximate eq.2.3 by considering the value of displacements or forces at the previous time step n, therefore leading to explicit or weak formulations:

$$\boldsymbol{P}_{n+1} \approx f(\boldsymbol{u}_n) \qquad \text{or} \qquad \boldsymbol{u}_{n+1} \approx g(\boldsymbol{P}_n) \qquad \text{at} \quad t_{n+1} \tag{2.4}$$

As already mentioned, staggered schemes tend to be unstable in general, especially if large coupling timesteps are employed. Some authors argue that this instability is intrinsic in this kind of approximate methods, while others believe it is consequence of inadequate time integrators [21]. In any case, such an instability is related to the approximation in the transfer of forces and displacements, as these are not computed at the correct time instant and a lag of one timestep is introduced [2]. The effect of the approximation in eq.2.3 can be seen as an artificial mass effect, because the acceleration of the fluid due to body motion introduces additional inertia into the system. As described in [21], the magnitude of this phenomenon depends on the ratio between fluid and structure density, which for typical aeroelastic applications is very small. Consequently, this leads to a negligible effect on the solution, making loose couplings still of interest, provided that small coupling timesteps are employed in order to minimize the error caused by the approximation in eq.2.4. In all the cases where the added mass effect is not negligible - such as hydroelasticity - tight couplings are required [21]. There are two distinct families of loose couplings, either serial or parallel schemes. In the first, the fluid and structural solvers run one after the other in a sequential fashion, while in the second case they run simultaneously. The simplest type of serial loose coupling is the Conventional Sequential Staggered (CSS) procedure [29] visible in fig.2.2-(a), while its counterpart for parallel loose couplings is the Conventional Parallel Staggered (CPS) scheme shown in fig.2.2-(b). Both methods are straightforward to implement and conceptually very simple, but can suffer from instabilities and poor accuracy, especially the CSS method, where the lack of synchronisation between fluid and structure leads to an error decreasing only linearly for decreasing timestep even with dedicated second-order accurate in time solvers. Furthermore, if the coupling timestep is not sufficiently small, this method might be unstable leading to totally incorrect predictions, as discussed in [3]. Regarding the CPS method, it allows to naturally reduce the total simulation time, provided that the timestep is kept relatively small to assure numerical stability and sufficient accuracy [3]. To improve the deficits of these two schemes, more sophisticated methods have



(a) CSS scheme with sub-cycling of the fluid phase.

(b) CPS scheme.

Figure 2.2: Two of the most common and simple FSI schemes for loosely coupled fluid and structure solvers in time [3].

been developed. For instance, an improvement over the CSS method is the Improved Serial Staggered (ISS) method [3], which can be considered as a mid-point staggered procedure. Besides, the ISS scheme enforces the conservation of both velocity and displacements at the interface, achieving timesteps even five times larger then those required by CSS [21]. Regarding parallel schemes, an enhancement of the CPS is the Improved Parallel Staggered (IPS) [3]. The latter latter was developed to fix the CPS's lack of feedback between fluid and structure within each iteration by exchanging data at the middle of the time step instead of at the end. According to [3], although the IPS returns an improvement of performance over the CPS, it is still not as fast as the ISS.

Sub-cycling variant of staggered schemes

As staggered schemes allow to use different timesteps for the fluid and structure domains, this leads to the sub-cycling variant of staggered schemes, as shown in fig.2.2-(a), for the fluid phase. Typically, the coupling timestep Δt equals to the structural timestep Δt_s, and the fluid time step Δt_f can be either derived by choosing an appropriate number of subcycling iterations n_{sc} or the number of subcycling iterations can be derived from Δt_f, if the latter is prescribed. This approach saves computational time and resources, since the structure is solved at a lower frequency of update and the number of data exchange between the two solvers is reduced [30]. Two important problems are related to subcycling schemes: the first one is the reconstruction and even distribution of the structural displacement (which is provided every Δt_s) within the subcycles of the fluid solver Δt_f, the second one is the determination of which interface pressure to transfer to the structural solver [31]. The first aspect is relatively easy to solve, as either the force value at the last subcycle or, even better, an average over Δt_s can be returned [4]. On the other hand, the second aspect is more difficult to solve, as the entire structure motion should not be applied during the first fluid subcycle, but evenly distributed by means of suitable methods to reconstruct the interface motion [4]. The simplest of these methods is a linear interpolation of the displacements, but this leads to a lack of continuity for the velocity of the reconstructed interface motion at every coupling instant, since the solution will only be continuous. Such velocity and also acceleration discontinuities can negatively impact on the prediction of the unsteady surface fluid pressure [31] and this might lead to the potential propagation of spurious acoustic waves with an over-estimation of noise, especially if the flow solver is compressible. Always according to [31], better interface motion in the fluid solver can be achieved if higher order interpolation methods are employed, for instance spline interpolation. Nevertheless, this could also generate additional oscillations in the reconstructed motion due to the necessity of matching the first and second derivative at every Δt_s. Regarding the thesis, a loose coupling approach with subcycling is required, since the computational timestep of PowerFLOW® for typical high Re number applications can be orders of magnitude smaller than the structure one, while the interface motion reconstruction is handled by means of linear interpolations, as discussed in sec.5.4.1. Finally, as reference for these aspects, [32] and [33] deal with the application of the subcycling approach to LBM-based CFD solvers on Cartesian grids can be considered.

2.1.3. Interface treatment

The interface treatment is a crucial aspect of staggered FSI methods and it typically consists of two aspects: matching the fluid mesh and the structural mesh at the interface, and fluid mesh movement or deformation according to the ALE formulation, see fig.2.3 from [2] and [4]. As these issues are primarily related to 3D flexible structures, which has not been considered in this thesis, here only a short overview is present. The first issue deals with both lack of conformity between the two meshes and different grid-size, see fig.2.4-(a), as well as the treatment of gaps and overlaps shown in fig.2.4-(b). This is related to the properties of energy conservation and interpolation consistency for the match of data on the two sides of the interface respecting the fundamental compatibility and equilibrium interface relations. Among the different methods, the most common ones are the Weighted Residual Method and Radial Basis Function Interpolation [2], [34]. For more advanced methods, [21] and [22] can be considered. Regarding the second aspect, there are many ways of deforming the fluid mesh based on the elastic deformation of the body surface and interface. The most simple one is to recompute a new fluid mesh after each coupling iteration. Although this solution is very robust - as the new mesh is completely uncorrelated to the old one - it is rather undesirable due to the computational cost, update frequency, continuous user interaction, as well as the lack of conservation of physical quantities, unless these are retained by interpolation. Alternatively, automatic and efficient methods are more popular, and most of these fall into two big families, mesh movement approaches for structured meshes (for instance Trans-Finite Interpolation) and those for unstructured meshes. To mention a few: spring analogy, functional-based, Laplace and Biharmonic equation-based methods [2].

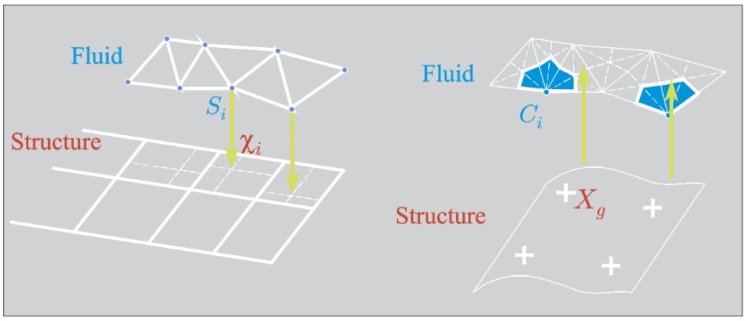
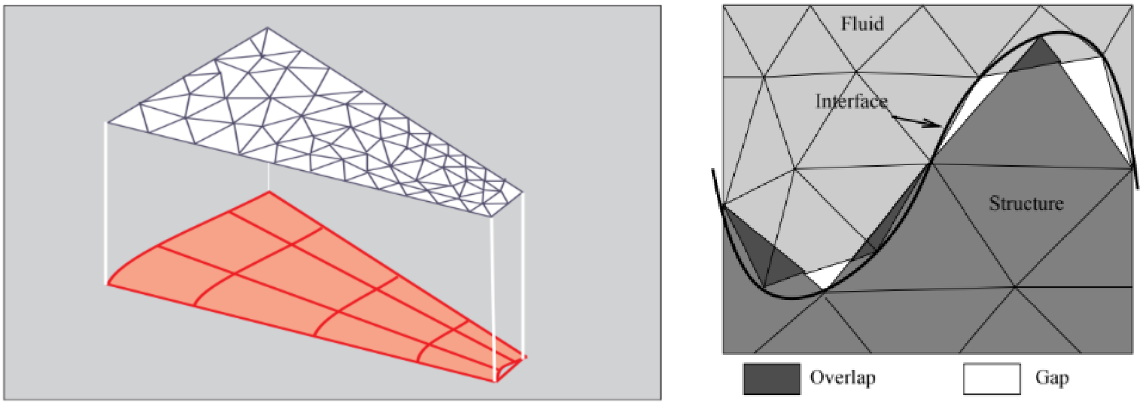


Figure 2.3: Dealing with the equilibrium and consistency BCs at the interface of different fluid and structural meshes, reference [2] and [4].



(a) Different fluid and structure meshes at the interface [2] and [4]

(b) Interface irregularities [2].

Figure 2.4: Geometric and topological issues concerning the matching of different fluid and structural meshes.

2.2. Fluid model and solver

2.2.1. The Lattice Boltzmann Method

PowerFLOW® is an unconventional CFD tool, as it is based on the LBM, contrarily to most of the CFD tools relaying on the Navier Stokes equations. Within this section, the theoretical foundations as well as the most important physical and numerical aspects of such method are explored, in particular the Boltzmann Transport Equation and the way the LBM is derived from it, the lattice models, the main equations and boundary conditions. More details about these topics and on the whole LBM can be found in literature, for instance [35] and [36].

The continuous Boltzmann Transport Equation

If the Navier-Stokes Equations relay on the assumption that the fluid-flow can be modelled as a continuum, the Boltzmann Transport Equation generates from the Kinetic Theory of Gases. According to this model, fluids are considered as collections of a very large amount of discrete particles moving with random motions and exchanging momentum and energy through streaming and billiard-like collisions [37]. Due to the particles number and their highly chaotic behaviour, fluid properties are described statistically, with thermodynamic quantities resulting from an average of particle kinematic states. Within this framework, the Boltzmann Transport Equation in eq.2.5 is a Partial Differential Equation (PDE) describing the transport of the particle distribution function $f = f(\boldsymbol{x}, \boldsymbol{e}, t)$:

$$\frac{\partial f}{\partial t} + \boldsymbol{e} \cdot \frac{\partial f}{\partial \boldsymbol{x}} + \boldsymbol{F} \cdot \frac{\partial f}{\partial \boldsymbol{e}} = \Omega \tag{2.5}$$

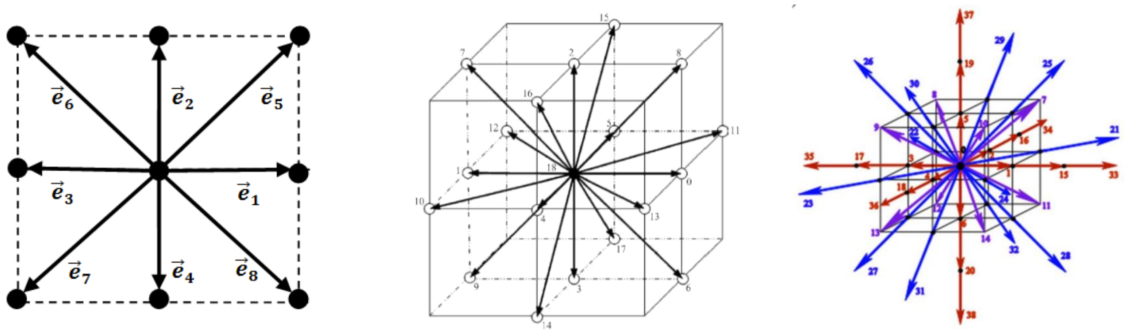
This statistical quantity estimates the probability of an existing particle at point $[\boldsymbol{x}, \boldsymbol{x} + \Delta\boldsymbol{x}]$ and at the continuous microscopic velocity $[\boldsymbol{e}, \boldsymbol{e} + \Delta\boldsymbol{e}]$. In the equation, the space and time evolution of the particle distribution function is influenced by a collision operator Ω accounting for particle interactions and by an external force vector $\boldsymbol{F}$ modelling external influences. The Boltzmann Transport Equation (1872) represents the analytical reference from where the LBM was developed starting from the 70s with the method of Lattice Gas Automata [38]. In its current formulation, the LBM can be seen as a special finite difference discretization of the Boltzmann Transport Equation.

Differences with the Boltzmann Transport Equation

At first, the LBM can be considered as simplification of the Boltzmann model for the dynamic behaviour of gases by reducing the particles number [37] and by introducing an intermediate - called mesoscopic - level between the microscopic world of particle kinematics and the continuum level of macroscopic thermodynamic quantities. Besides, the position and motion of particles in space are limited to a finite set of grid positions, called lattice nodes, which implies a prescribed number of possible directions and values for the microscopic velocity $\boldsymbol{e}_i$. More in detail, the LBM replaces the continuous distribution function $f = f(\boldsymbol{x}, \boldsymbol{e}, t)$ of the Boltzmann Transport Equation with a discretized f_i existing only along the finite directions $\boldsymbol{e}_i$ and depending only on the lattice nodes $\boldsymbol{x}_i$. Therefore, $f_i(\boldsymbol{x}_i, t)$ can be re-interpreted as the probability, for each particle, of streaming along one of the available directions in the lattice.

Lattice Models

The lattice is one of the key components of this method, as it does not only define the allowed positions and directions for particles at their mesoscopic level, but it also embodies the volume mesh where computations are performed to get a flow solution. For a 2D model, particles are restricted to stream



(a) D2Q9 model [37]

(b) D3Q19 model [39]

(c) D3Q39 model [40]

Figure 2.5: Lattice node models with the available directions and stencil points for the LBM.

in 9 possible directions per time step, including the resting conditions. While for a 3D model, a total of 19 directions can be defined. The first is called D2Q9 model, fig.2.5-(a), whereas the second is called D3Q19 model and is shown in fig.2.5-(b). Both models are suitable for low speed application (free-stream $Ma < 0.4$) when flow-compressibility effects are limited. For transonic and low supersonic (free-stream $Ma < 2$) flow regimes of non-isothermal flows, more complex lattice models are required to properly capture the flow physics, like the D3Q39 model with 39 possible particle streaming directions [41], visible in fig.2.5-(c).

Main equations

Given the discrete distribution function $f_i(\boldsymbol{x}_i, t)$, the Boltzmann Transport Equation is frequently discretized by omitting external forces $\boldsymbol{F}$ and by employing a finite difference approximation of the total derivative of f_i. This leads to the main equation of the LBM, shown in eq.2.6, which is a time-explicit advection equation for the the distribution function under the effect of a discrete collision operator Ω_i, which needs to approximated by a specific particle-collision model:

$$f_i(\boldsymbol{x} + \boldsymbol{e}_i \Delta t, t + \Delta t) - f_i(\boldsymbol{x}, t) = \Omega_i \approx -\frac{[f_i(\boldsymbol{x}, t) - f_i^{eq}(\boldsymbol{x}, t)]}{\tau} \tag{2.6}$$

One of the most used models for the collision operator is the Bhatnagar-Gross-Krook (BGK) model [42] shown on the right-hand-side term of the above equation, where an equilibrium distribution function f_i^{eq} and a relaxation time τ towards local thermodynamic equilibrium are employed. The relaxation time is related to the kinematic fluid viscosity ν and temperature T through the relation $\tau = \nu/T + \Delta t/2$, where Δt is the local time-step within the lattice, and it plays an important role in the modelling of the turbulence. Regarding the equilibrium distribution function, Hermite expansions up to a certain order featuring local hydrodynamic properties such as density ρ, velocity $\boldsymbol{u}$ and temperature are usually employed for this quantity [19]. In the case of compressible flows, eq.2.6 is not sufficient and an additional PDE for the entropy is required to satisfy the conservation of energy according to a non-isothermal model [41]. In the LBM, macroscopic fluid variables can be recovered in terms of the microscopic ones by means of discrete integration [19]. For instance, the macroscopic density $\rho(\boldsymbol{x}, t)$ velocity $\boldsymbol{u}(\boldsymbol{x}, t)$ and internal energy $e(\boldsymbol{x}, t)$ field quantities are in general expressed by assuming that the distribution function does not depend on the microscopic velocity, as:

$$\rho(\boldsymbol{x}, t) = \sum_{i=0}^{N} f_i(\boldsymbol{x}, t); \quad \boldsymbol{u}(\boldsymbol{x}, t) = \frac{1}{\rho(\boldsymbol{x}, t)} \sum_{i=0}^{N} f_i \boldsymbol{e}_i; \quad e(\boldsymbol{x}, t) = \frac{D}{2} \frac{T}{\rho(\boldsymbol{x}, t)} = \sum_{i=0}^{N} f_i(\boldsymbol{x}, t) \left| \boldsymbol{e}_i - \boldsymbol{u} \right|^2 \tag{2.7}$$

where Δx is the lattice length and N the number of stencil points for the lattice model considered. The three relations in eq.2.7 are fundamental, as they link the mesoscopic level of the LBM with the macroscopic level of thermodynamics and fluid flow by means of the standard relations for an ideal gas.

Boundary Conditions

As for every numerical method, boundary conditions play a crucial role for the stability and the accuracy of the computed solution [37]. Furthermore, for the LBM, boundary conditions have to respect the true macroscopic behaviour of the fluid. One of the most used conditions is the bounce-back one, which is typically employed to apply no-slip conditions on the boundary. As described in [37], it can be simply explained by saying that "when a fluid particle reaches a boundary node, the particle will scatter back to the fluid along with its incoming direction". Another commonly used condition is the specular reflection one, where only the normal component of the velocity vector is inverted creating a 90° angle between the particle velocities before and after the collision with the wall. A sketch of both boundary conditions is visible in fig.2.6 from [5].

General properties, positive aspects and limitations

Regarding numerical aspects, the LBM offers important advantages when compared to traditional CFD methods based on the Navier-Stokes Equations. First of all, this method returns an intrinsically unsteady and compressible flow solution for low Mach number applications which is richer in flow-details than RANS and with a reduced cost compared to wall-bounded LES simulations. Next, the method is also appreciated for its inherent simplicity, both in the equations, as there are not complex non-linear PDEs to discretize, and in the implementation of the algorithm. Eventually, the LBM is naturally suitable for

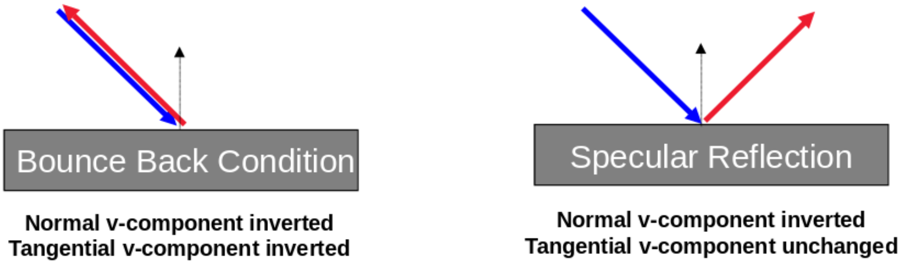

Bounce Back Condition
Normal v-component inverted
Tangential v-component inverted
Specular Reflection
Normal v-component inverted
Tangential v-component unchanged

Figure 2.6: Sketch of the most commonly used BCs for LBM [5].

parallelization, due to the full scalability of the lattice-based approach [43]. Dealing with engineering applications, the LBM is a versatile approach for multiphase/multicomponent flow simulations, flows undergoing heat transfer, chemical reactions, resolved or modelled turbulence in the case of particles or bubbles dynamics, due to the more realistic description of the small-scale physical phenomena than traditional CFD methods [44]. Moreover, due to its low numerical dissipation and dispersion [45], this method can be very effectively applied for Computational Aeroacoutics applications, where the near-field noise can be sampled at a high frequency and employed to compute the far-field noise properties of the source by means of aeroacoustic analogies. When it comes to the downsides of the LBM, the most significant one is related to the simulation of supersonic and hypersonic flow conditions, which are not very well resolved by this method, requiring still fundamental research in the underlying physical and numerical models [46]. Another limitation is the higher computational cost of simulating steady flow-phenomena compared to RANS, due to the very small timestep employed to compute the LBM intrinsic unsteady solution, although the cost of LBM is not necessarily higher than URANS when it comes to true unsteady flows, as long as the large turbulent flow structures are not resolved.

Connection to the Navier-Stokes equations

An important aspect to discuss at the end of this brief introduction and overview of the Lattice Boltzmann Method, is related to its connection to the Navier-Stokes Equations. Therefore, it can be proved that the latter can be retrieved from the Boltzmann Transport Equation by employing the Chapman-Enskog procedure, a multi-scaling expansion technique consisting in a perturbation-series of the Knudsen number with respect to the local equilibrium [47]. This match between such two different models for predicting the behaviour of fluid flows is a fundamental results and it was proven to hold true, and as long as $Ma < 2$ for non-isothermal flows by employing a D3Q39 lattice model due to its high symmetry properties. This result has a fundamental importance, as it shows that there is a link between the microscopic particle description of fluids done by the Boltzmann Transport Equation, the approximate mesoscopic treatment of the Lattice Boltzmann Method and the continuum macroscopic one of the Navier-Stokes Equations.

2.2.2. PowerFLOW®

PowerFLOW® is a software suite for aerodynamic, thermal and aeroacoustic simulations consisting of several programs covering the entirely a typical simulation workflow, namely geometry preparation (PowerDELTA®), case-setup (PowerCASE®), simulation (solver and discretizer) and postprocessing (PowerVIZ®, PowerINSIGHT® and PowerACOUSTICS®). For reasons related to the nature of this thesis project, PowerCASE® is the most employed program of the suite. Therefore, in this section relevant general aspects of PowerCASE® and of the PowerFLOW® solver are described, for instance turbulence and wall modelling, volume and mesh discretization and the handling of body motion, as well as specific functionalities (body force, measurement files and table lookup) that can play a role in the coupling with an external tool.

Turbulence modelling

PowerFLOW® is based on a peculiar implementation of the LBM called Digital Physics® [48] which consists in the eq.2.6 equiped with the BGK approximation of the collision operator. The only difference is in a dedicated extension to turbulent flows. For low Reynolds and Mach numbers, PowerFLOW® performs the equivalent of a Direct Numerical Simulation of the Navier-Stokes Equations compatibly with the lattice model employed (number of discrete particle velocity vectors) and the lattice resolution necessary to resolve the smallest flow structures [49]. As the Reynolds number increases, turbulence modelling is required given the increased number of degrees of freedom and time-space scales. When it comes to turbulence modelling, PowerFLOW® employs an adaptation of the two equation RNG $k-\epsilon$ model from the RANS framework [50] to model the effect of the unresolved turbulent structures on the resolved large eddies [44]. To discriminate between resolved and unresolved scales, the swirl model described in [51] is employed. The overall method is called Very-Large Eddy Simulation (V-LES) where the two partial differential equations of the $k-\epsilon$ RNG model provide the space and time evolution of the turbulent kinetic energy k and dissipation rate ϵ used to compute the turbulent relaxation time τ_{turb} shown on the right relation of eq.2.8, where $C_\mu = 0.09$ and $|\boldsymbol{S}|$ is a measure of the local strain [49]. The turbulent relaxation time accounts for collisions due to the interaction of turbulent structures and is added to the relaxation time, which models instead collisions due to molecular diffusion τ_0, leading to an effective relaxation time τ_{eff} in the BGK collision operator Ω_i, as the right relation in eq.2.8 shows:

$$\tau_{turb} \approx C_\mu \frac{k^2/\epsilon}{T\big(1+(k|\boldsymbol{S}|/\epsilon)^2\big)^{\frac{1}{2}}}; \qquad \tau_{eff} = \tau_{turb} + \tau_0; \qquad \Omega_i \approx -\frac{1}{\tau_{eff}}(f_i - f_i^{eq}); \tag{2.8}$$

Wall model

Due to the tremendous cost of computing flow structures down to the viscous sub-layer and due to the type of volume mesh employed, PowerFLOW® exploits wall models to simulate high Reynolds number turbulent flows around solid surfaces with sufficient accuracy but at a reasonable cost. This is achieved by employing the universal logarithmic law-of-the-wall velocity profile extended to include adverse or favourable pressure gradients by means of a the empirically determined constant A in eq.2.9:

$$u^+ = \frac{u_s}{u_\tau} = \frac{1}{k} ln\left(\frac{y^+}{A}\right) + B; \quad \text{where} \quad y^+ = \frac{y_s u_\tau}{\nu}; \quad \text{and} \quad u_\tau = \sqrt{\frac{\tau_w}{\rho}}; \tag{2.9}$$

In the above equation, u^+ and y^+ are the dimensionless velocity and wall-distance, u_s is the dimensional flow velocity computed at the location y_s, namely the dimensional wall-coordinate of the cell-layer closest to the wall; besides, $k = 0.41$ is the von-Karman constant and $B = 5.0$, as well as u_τ is the friction velocity. The left equation above is iteratively solved starting from the cell-layer closest to the wall to estimate the wall shear stress τ_w required by the boundary conditions prescribed by the LBM [19]. Regarding the constant A, it models the expansion and deceleration of the velocity profile under the effect of an adverse streamwise pressure gradient ($dP/ds > 0$) following the relation $A = 1+\delta|dP/ds|/\tau_w$, where δ is an estimate of the size of the unresolved near wall region, or in case of a favourable one ($dP/ds \leq 0$) $A = 1$.

Volume and surface mesh

PowerFLOW® employs an automatically generated Cartesian volume mesh, the lattice, composed of cubic elements called voxels. The process of domain and body surface discretization is carried out only once at the beginning of the simulation and its outcome can not be modified during the simulation. This limits the employment of traditional FSI approaches based on the ALE formulation and, at the same time, also Immersed Boundary Methods are not available in the solver. The size of voxels can change based on Variable Resolution (VR) regions specified by the user, which act as refinement regions in areas of the computational domain where small fluid structures or large gradients are expected [19]. The spatial refinement is carried out in such a way that two neighbouring VR regions have voxel sizes Δx differing by a factor of two. Furthermore, given an explicit time-marching scheme with a Courant–Friedrichs–Lewy number $CFL = |max(\boldsymbol{e}_i)\Delta t|/\Delta x = 1$, two neighbouring VR regions are characterized by a local timestep which differs by a factor of two. Consequently, in finer areas of the domain the solution is updated at a higher frequency than in coarser regions. Finally, the value of the local timestep is automatically defined in the solver based on the size of voxels, and the discrete particle velocity $\boldsymbol{e}_i$. On the tessellated surface of the body, planar surface elements, called surfels, are automatically generated by the intersection between volume mesh voxels and the body facets. This process allows to capture the geometry details of complex shapes in an accurate and robust manner, up to an order of magnitude difference between the smallest voxel size and the average facets size.

Body motion

As the solver can only handle rigid bodies and not time dependent deformations, due to the mesh generation algorithm, the allowed motions are limited to rotation by means of a truly rotating mesh, namely the "Local Rotating Frame" (LRF) method, and translation by means of a transpiration velocity modelling blowing or suction of fluid-flow, in numerical terms a normal wall boundary conditions. The LRF is employed to simulate truly rotating rigid geometries - both 3D such as fan and wheels and 2D like pitching airfoils - by extending the LBM equations with centrifugal and Coriolis force terms in the lattice nodes inside the rotating region [52]. Besides, this approach allows to accurately capture the flow dynamics also in the case of real systems where installation effects can break flow symmetry, because the body is correctly moving as in real life. On the other hand, the transpiration velocity is a method which can be exploited in PowerFLOW® to simulate the effect of small holes acting as flow or boundary layer control mechanisms or to approximate a true rigid displacement normal to the wall, as long as the amplitude of motion is small compared to the airfoil chord [52]. In the latter case, this method can provide a similar flow behaviour compared to the one of the true motion case, therefore this method is frequently employed to simulate vibrations [52]. When large displacements occur, such equivalence might not be justified as the flow behaviour loses its linearity and complex non-linear phenomena like flow separation can occur. Finally, both the LRF rotating geometry and the transpiration velocity boundary condition are here only introduced, while a more in depth discussion and application to this thesis project is carried out in sec.4.1.4.

Body forces

PowerCASE® allows to include body forces into the LBM formulation shown in eq.2.6 by means of the external force vector $\boldsymbol{F}$. This functionality can be exploited to model the effect of a true rigid or flexible body motion on the flow field by applying such forces to arbitrary regions of the fluid domain near the surface of the body. In PowerCASE®, equations can be defined to compute such forces in space and time, while geometry primitives can be set to prescribe regions of the simulation volume where these forces are applied. In [6], this approach was developed and implemented in the CFD setup for both a rigidly pitching airfoil and an elastic blade affected by a flexible torsional deformation. For the 2D airfoil case, body forces were defined on a x-y coordinate system centered around the quarter chord and applied to two cylindrical volumes surrounding the leading edge (LE) and trailing edge (TE) of the airfoil. Expressions for the dimensional and mass-scaled $\boldsymbol{b}_{LE}$ and $\boldsymbol{b}_{TE}$ body forces at the LE and TE were obtained from approximate expressions of the acceleration vectors of the LE and TE of the airfoil respectively, and are reported below as shown in [6], while more details on the derivation are discussed in the appendix sec.A:

$$\boldsymbol{b}_{LE} = -\dot{\tilde{\phi}}^2 \frac{c}{4}\hat{x} - \ddot{\tilde{\phi}}\frac{c}{4}\hat{y} \qquad\qquad \boldsymbol{b}_{TE} = \dot{\tilde{\phi}}^2 \frac{3c}{4}\hat{x} + \ddot{\tilde{\phi}}\frac{3c}{4}\hat{y} \tag{2.10}$$

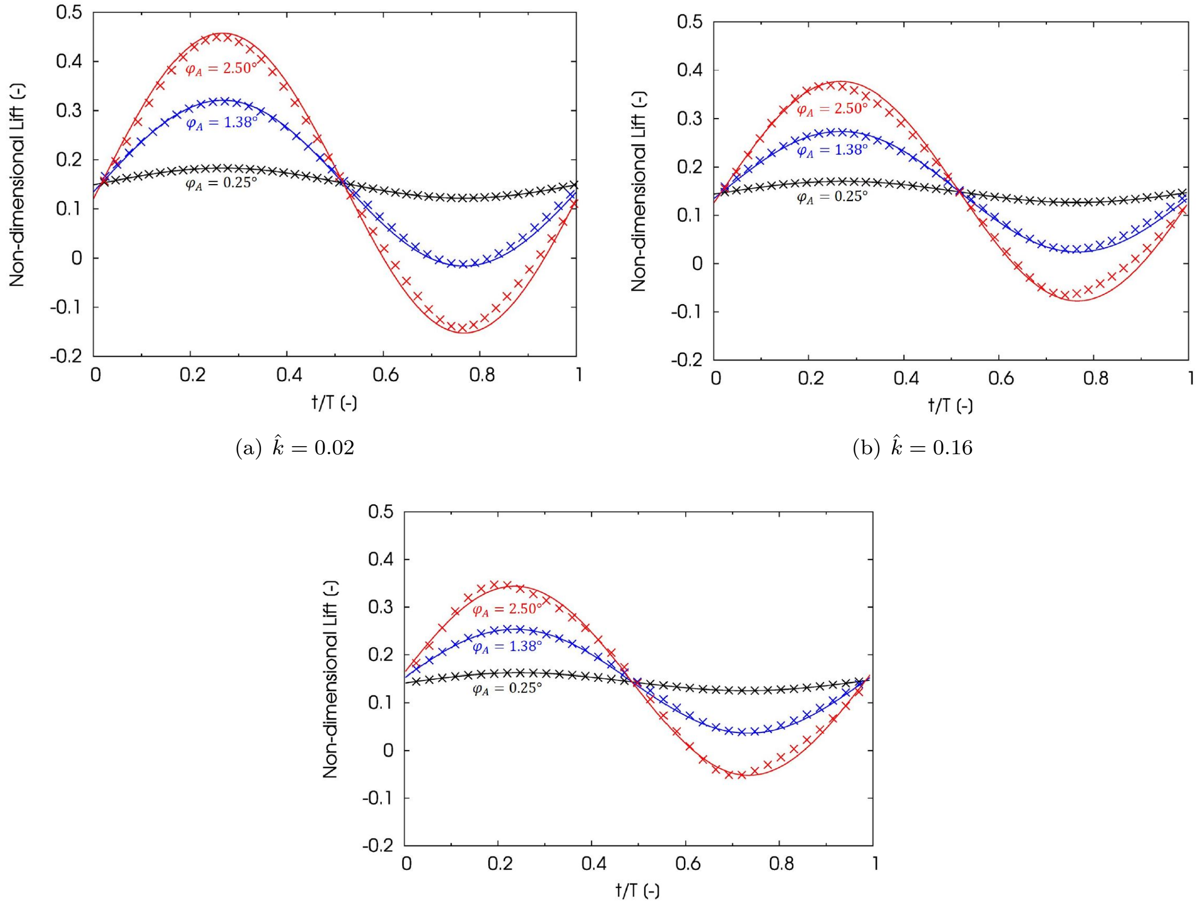
In the above equation, c is the airfoil chord, while $\dot{\tilde{\phi}}$ and $\ddot{\tilde{\phi}}$ are the amplitude modulated and phase shifted pitch velocity and acceleration of the airfoil with prescribed pitch motion:

$$\phi(\hat{t}) = \phi_A \sin\left(k\hat{t}\right) \quad \text{with} \quad k = \omega c/2U_\infty \quad \text{and} \quad \hat{t} = 2U_\infty t/c \tag{2.11}$$

where k is a reduced frequency (defined from the freestream flow speed U_∞ and reduced frequency of pitch motion ω), while $\hat{t}$ is a reduced - the dimensional time t scaled by the convection time $c/2U_\infty$ - and ϕ_A is finally the dimensionless pitch amplitude. The expressions for the previous dimensional pitch velocity and acceleration are shown in eq.2.12, where b is the airfoil semi-chord. The previous quantities require, with respect to the actual pitch velocity and acceleration, amplitude modulation, so that the body force field can modify (up to a certain level) the fluid flow streamlines surrounding the airfoil as those of a real pitching airfoil, and phase shift, as the body force field leads to some initial time delay to replicate the streamline pattern of the rotating geometry.

$$\dot{\tilde{\phi}}(\hat{t}) = \phi_A \frac{kU_\infty}{b}\sqrt{\xi_1}\cos\left(k\hat{t} - 2\pi\xi_2\right) \qquad\qquad \ddot{\tilde{\phi}}(\hat{t}) = -\phi_A \frac{k^2U_\infty^2}{b^2}\xi_1 \sin\left(k\hat{t} - 2\pi\xi_2\right) \tag{2.12}$$

Values for the dimensionless amplitude-modulation coefficient ξ_1 and the dimensionless phase-shifting coefficient ξ_2 need to be calculated for each condition of reduced frequency and pitch amplitude by solving an optimization problem to match the lift generated by the truly pitching airfoil via the rotating



(a) $\hat{k} = 0.02$

(b) $\hat{k} = 0.16$

(c) $\hat{k} = 0.3$

Figure 2.7: Time histories of the lift coefficient over the period of oscillation for the truly pitching airfoil with LRF (solid line) and for the virtually pitching airfoil with body forces (crosses) for three conditions of reduced frequency $\hat{k}$ and three conditions of amplitude ϕ_A [6].

geometry LRF approach [6]. Always in [6], a sensitivity study was carried out to assess the potentialities and limitation of this method for varying ϕ_A and $\hat{k}$. As shown in fig.2.7, a very good match is apparent over the whole range of investigated reduced frequencies (those covered along the blade-span by the HART-II experiment) and up to pitch amplitudes of 2.5°. For $\phi_A = 2.5°$, small deviation start to appear due to the no more limited amplitude of motion. Further validation of this approach is part of the current thesis research project, as it is crucial for the effective application of the coupling methodology to 3D geometries with flexible motion.

Measurement files

To extract useful data from the simulation, measurement files need to be specified via dedicated windows in PowerCASE®. Many possible aerodynamic quantities can be recorded, for instance fluid, surface and porous media data computed in space and time or integrated over volumes/surfaces (*Composite measurements*) as well as point-wise (*Probes*). A very important aspect of measurements files in PowerFLOW® is related to the measurement frames, which are integer multiples of the computational timestep and are used to define the time resolution of these files. For reasons related to the underlying LBM and numerical stability, the solver employs a very small timestep, which, if used directly to sample results, will generate excessively large files. Therefore, according to the purpose and the fluid quantity of interest, the user can choose a sampling period and a corresponding averaging interval to get unsteady results and multiple combinations of these two parameters are available, as described in [52]. The same approach applies also in space, as the spatial resolution of surface or volume results is lower than the one of the finest VR. Also in this case, the user can select an appropriate "linear width of measurement cell", which is usually twice as big as the local lattice length, obtaining one measurement cell from eight voxels in 3D (or four in 2D simulations). If averaged results are required, in PowerCASE®, time averaged results can be prescribed to obtain simulation results comparable to those of a RANS simulation. The same can also obtained form transient data at the end of the simulation by employing the PowerFLOW® Command Line Interface (CLI) tools, as described in [53].

Calculation page and input tables

The calculation page and input tables are functionalities of PowerCASE® allowing customization of the simulation setup. In the calculation page the user can define variables and equations, while external data can be included in the simulation setup by means of input tables. In the calculation page, by defining variables and equations it is possible to provide time and space varying boundary and initial conditions, as well as to parametrize the simulation setup [52]. The latter can be effectively exploited for geometry (VR regions), simulation parameters and measurement files, improving efficiency and reducing mistakes. The calculation page consists of four tabs: a text editor, the commands to evaluate quantities and calculate the equations in space and time, a value feedback pane and position-time specification area. According to [52], table look-up can be primarily used in PowerCASE® to import data from experiments in the simulation setup and reference them in the equations. However, this feature can be exploited also to provide data from another solver, and this can also be done repeatedly during the simulation to achieve a coupling. An import aspect of this tool is that input tables must provide a grid of points where multiple quantities can be defined. Such grid can be uniform or nonuniform, and PowerFLOW® employs tri-linear interpolation to accurately compute values at the locations where data are missing, as long as this occurs only for a few points. In addition to grid points and variables in the datasection, physical units, grid parameters and further instructions for reading the table are pointed out in the header of the table. Rules to write such header and to structure these tables are explained in detail in [52]. If the table is repeatedly loaded, two options are available: table import based on intervals of the solver timestep, or based on the measurements frames of a specific measurement window. In both cases, the first timestep/frame, the interval and the last timestep/frame of import need to be specified. Besides, for frame reading, the user can specify a bash command to be executed by the simulator before the table look up. This can be used to trigger a coupling interface.

2.3. Multi-Body Dynamic theory and solver

2.3.1. Theoretical foundations of the Multi-Body approach

This section offers an overview of the most common models to mathematically describe the dynamic behaviour of a rigid body in terms of dynamic-equilibrium-based and energetic approaches to derive Equations of Motion (EoMs). For more detailed information, the reader can consider [54] and [55] as valuable sources of knowledge.

Newtonian dynamics of point masses

The three fundamental laws of Newtonian dynamics for point masses dm [56] represent the starting point of the theoretical framework of rigid body dynamics. The first and the third law are merely qualitative, with the first defining when a body is at rest. The third law affirms that forces exchanged by bodies always appear in couple, opposite in direction and simultaneously, regardless of the action-reaction principle. On the other hand, the second law, eq.2.13, is much more important for quantitative purposes, as it defines an instantaneous direct proportionality between the force $\boldsymbol{F}(t)$ exerted on a mass and its acceleration $\boldsymbol{a}(t)$:

$$\boldsymbol{F}(t) = dm\,\boldsymbol{a}(t) \tag{2.13}$$

From eq.2.13, many other important equations can be derived. For example, the work-kinetic energy principle in eq.2.14 which affirms that the change in the kinetic energy T of a point mass between times t_1 and t_2 equals the work W done between the two time instants:

$$\Delta T_{2-1} = T_2 - T_1 = W_{2-1} = \int_{t_1}^{t_2} \boldsymbol{F}(t) \cdot d\boldsymbol{r} \tag{2.14}$$

Next, the impulse or (linear) momentum principle defines the equivalence between the impulse (left of eq.2.15 with $\boldsymbol{R}(t)$ the force resultant) and the linear momentum (right), where $\boldsymbol{v}$ is the point mass velocity. This law is useful when forces determine large accelerations over short periods of time, as shown below:

$$\int_{t_1}^{t_2} \boldsymbol{R}(t)\, dt = dm(\boldsymbol{v}_2 - \boldsymbol{v}_1) \tag{2.15}$$

Finally, the principle of conservation of angular momentum - shown in eq.2.16 where $\boldsymbol{dD}$ is the angular momentum for the point mass itself - can be considered as a rotational version of eq.2.13. For a point mass at location P embedded in a mass-less rigid flat body, the below relation considers: a moment M_O about a reference location O of a force acting on the point mass, $\boldsymbol{r} = \boldsymbol{r}_{P-O}$ as the position vector of point P with respect to O, and $\boldsymbol{\omega}$ as the angular velocity of the body.

$$\frac{d}{dt}\big(\boldsymbol{r} \times dm\, \boldsymbol{v}\big) = \boldsymbol{M}_O \qquad \rightarrow \qquad \frac{d}{dt}\big(\boldsymbol{r} \times dm(\boldsymbol{\omega} \times \boldsymbol{r})\big) = \boldsymbol{r} \times \boldsymbol{F} \qquad \rightarrow \qquad \frac{d}{dt} d\boldsymbol{D} = \boldsymbol{r} \times \boldsymbol{F} \tag{2.16}$$

Linear and angular momentum conservation laws for rigid bodies

The extension of Newtonian dynamics for point particles to rigid bodies starts by considering the latter as a clouds of point particles interacting under the effect of external forces $\boldsymbol{F}_i$ and internal forces $\boldsymbol{f}_{ij}$. Therefore, the 2nd law for a single point mass is extended by considering the sum of all internal forces $\sum_{j=1}^{N} \boldsymbol{f}_{ij}$ in eq.2.13 with $j \neq i$ acting within the cloud. Finally, as shown in eq.2.17, this leads to the linear momentum conservation principle for a rigid body where, due to the 3rd Newtonian law, the sum of the only external forces is proportional to the acceleration of the cloud Center of Gravity (CoG) $\boldsymbol{a}_{cg}$ through the total cloud mass m:

$$\sum_{i=1}^{N} \boldsymbol{F}_i + \sum_{i=1}^{N}\sum_{j=1}^{N} \boldsymbol{f}_{ij} = \sum_{i=1}^{N} dm_i\, \boldsymbol{a}_i \quad \text{with } j \neq i \qquad \rightarrow \qquad \sum_{i=1}^{N} \boldsymbol{F}_i = m\, \boldsymbol{a}_{cg} \tag{2.17}$$

In a similar fashion, the angular momentum conservation principle for a rigid body can be derived, as eq.2.18 shows. The final result can be straightforward, if the CoG is taken as reference point for the evaluation of the external forces moment $\boldsymbol{M}$ and the body angular momentum $\boldsymbol{D}$:

$$\sum_{i=1}^{N} \boldsymbol{r}_{i/O} \times \left[\boldsymbol{F}_i + \sum_{j=1}^{N} \boldsymbol{f}_{ij}\right] = \sum_{i=1}^{N} \frac{d}{dt}\Big(\boldsymbol{r}_{i/O} \times dm\big(\boldsymbol{\omega} \times \boldsymbol{r}_{i/O}\big)\Big) \quad \text{with } j \neq i \quad \rightarrow \quad \boldsymbol{M}_{cg} = \frac{d\boldsymbol{D}_{cg}}{dt} \tag{2.18}$$

Deriving a mathematical expression for the angular momentum of a 3D rigid body is not trivial. The derivation starts from the volume ($\mathcal{V}$) integration of the elementary angular momentum for a point mass $d\boldsymbol{D}$ shown in eq.2.16. It can be proved that the final expression of $\boldsymbol{D}$ equals the matrix product between the inertia tensor $\boldsymbol{J}$ and the full angular velocity vector $\boldsymbol{\omega}^T = [p, q, r]$. The inertia tensor features as diagonal components the moments of inertia and as off-diagonal terms the products of inertia. The latter cancel out if the inertia tensor is referred to both the body CoG and a system of principal inertial axis. Both eq.2.17 and eq.2.18 can be used to derive EoMs for rigid bodies in space, which are characterized by six DoF each, in absence of constraints. These six independent parameters can be divided into three translations components u, v, w computed by eq.2.17, and three parameters that define its orientation, for instance the three Euler/Cardan angles α, β, γ, calculated with eq.2.18.

D'Alembert approach and dynamic equilibria

An alternative approach to the conservation of linear and angular momentum principles for a rigid body to get EoMs consists in rewriting eq.2.17 and eq.2.18 in terms of dynamic equilibria [57]. This approach was developed by D'Almbert and allows a parallelism between statics and dynamics, as long as the time derivative of the linear momentum and the angular momentum are considered as apparent forces acting on the system. Therefore, from the linear momentum equation, the inertia force $\boldsymbol{F}_{in}$ is defined as minus the right-hand-side (rhs) of eq.2.17 (see the left relation below), while from the angular momentum relation the inertia moment $\boldsymbol{M}_{cg,in}$ is defined as the rhs of eq.2.18:

$$\boldsymbol{F}_{in} = -m\frac{d\boldsymbol{v}_{cg}}{dt} = -m\boldsymbol{a}_{cg} \qquad \text{and} \qquad \boldsymbol{M}_{cg,in} = -\boldsymbol{J}_{cg} \times \frac{d\boldsymbol{\omega}}{dt}$$

Once forces and moments of inertia are defined, the resultant of all forces acting on the body can be split into the resultant of reacting forces $\boldsymbol{\Lambda}$, namely constraints limiting the number of body DoF, and the resultant of the active forces $\boldsymbol{R}$. The same can be done for the total resultant moment $\boldsymbol{M}_{cg}$ in terms of $\boldsymbol{M}_{cg,\Lambda}$, and $\boldsymbol{M}_{cg,R}$. Finally, the D'Alembert principle affirms that a body is in a condition of dynamic equilibrium for translation if the sum of all the forces (including inertia forces) is zero, and for rotation if the sum of all moments (including inertia moments) is equally zero, as eq.2.19 shows.

$$\sum_{i=1}^{N} \boldsymbol{F}_i + \boldsymbol{F}_{in} = \boldsymbol{R} + \boldsymbol{\Lambda} + \boldsymbol{F}_{in} = 0; \quad \sum_{i=1}^{N} \boldsymbol{M}_i + \boldsymbol{M}_{cg,in} = \boldsymbol{M}_{cg,R} + \boldsymbol{M}_{cg,\Lambda} + \boldsymbol{M}_{cg,in} = 0 \tag{2.19}$$

Theorem of the Kinetic Energy

In addition to Newtonial dynamics and dynamic equilibria, EoMs for a rigid body can be obtained also by means of energetic approaches. If the first allows more physical insight in the understanding of the dynamic behaviour of the system, the latter can be much more convenient in the case of multiple bodies or many DoF [57]. There are two main energetic approaches, the theorem of kinetic energy and the Lagrange equations. Regarding the theorem of kinetic energy, which is shown in eq.2.20, it affirms that the time derivative of the kinetic energy of the system equals the power of the external active forces (Π) (excluding the inertial ones), or alternatively, the variation of kinetic energy equals the work (W) done by external forces.

$$\frac{dT}{dt} = \Pi \qquad \text{integration in time} \qquad \rightarrow \qquad \Delta T = W \tag{2.20}$$

To derive an expression for the kinetic energy of a rigid body, a handy way of expressing the velocity of an arbitrary point of the body itself is needed. This can be done by means of the Charles-Mozzi theorem [58] shown in eq.2.21. This important kinematic law defines the motion of a rigid body as the sum of a translation of its CoG and a rotation around an instantaneous rotation center:

$$\boldsymbol{v}_P = \boldsymbol{v}_{cg} + \boldsymbol{\omega} \times (\boldsymbol{P} - \boldsymbol{G}) \tag{2.21}$$

From the above relation, an expression for the kinetic energy of a rigid body, left of eq.2.22, can be derived as a superposition of a translation and a rotation component. The power of the active forces Π (left of eq.2.22) is simply equal to the product between forces and velocities, as well as moments and angular velocities, namely:

$$T = \frac{1}{2} m \boldsymbol{v}_{cg} \cdot \boldsymbol{v}_{cg} + \frac{1}{2} \boldsymbol{J}_{cg} \cdot \boldsymbol{\omega} \qquad \Pi = \boldsymbol{F} \times \boldsymbol{v}_P + \boldsymbol{M} \times \boldsymbol{\omega} \tag{2.22}$$

Finally, the reason why forces and moments of inertia are not included among the active forces is because the time derivative of the kinetic energy actually returns an inertial force and moment. Therefore, eq.2.20 can be also seen as $\Pi + \Pi_{in} = 0$, consistently with the D'Almebert's approach. The theorem of kinetic energy is not only handy, it also tells a lot about the behaviour of the system, since the sign of Π says if the kinetic energy of the system is increasing or decreasing. This suggests to interpret the inertia of the system as a reservoir of energy that is accumulated during acceleration and released during deceleration. Therefore, eq.2.20 is often employed for systems with 1 DoF as balance of powers. However, its applicability is limited to the actual motion of the system and not to the virtual, preventing its employment for systems with time-dependent constraints [57].

Lagrange formalism

According to this formalism, there are two types of Lagrange equations, those of the second type, the most commonly used for hand derivations, and those of the first type, which are perfect for automatic derivation and solution by means of a computer [57]. The second type Lagrange equations are written in terms of generalized coordinates q_n, namely those coordinates that allow to minimize the number of DoF for the system by implicitly accounting for its constraints [8]. Therefore, by introducing the potential U of conservative forces, many equations as eq.2.23 can be written for every generalized coordinate of the system:

$$\frac{\partial}{\partial t}\left(\frac{\partial T}{\partial \dot{q}_n}\right) - \frac{\partial T}{\partial q_n} + \frac{\partial U}{\partial q_n} = Q_n^{NC} \tag{2.23}$$

In the rhs of the above equation, Q_n^{NC} is the generalized component of the non conservative forces related to the coordinate q_n. It is obtained by deriving the virtual work of the non conservative forces with respect to the virtual displacement δq_n. A virtual displacement is infinitesimal, compatible with constraints and evaluated at fixed time. To be precise, q_n^{NC} should also include dissipative forces, unless they are defined as an independent term in eq.2.23, called dissipation function [8]. When it comes to the first type Lagrange equations, these are written by employing Cartesian coordinates to express the position and orientation of the body. Therefore, all possible DoF for a rigid body are initially considered leading to as many Ordinary Differential Equations (ODEs). Next, algebraic constraint equations ϕ_k are employed to model those physical constraint reducing the number of DOF to the actual number of independent parameters in the system. As described in [57], this approach employs a Lagrangian function $L = T - U$ augmented by a term L', which depends on the constraint relations by means of Lagrangian multipliers λ_k. The value of these coefficients does not affect L, as long as the ϕ_k relations are satisfied. At this point, the Lagrangian formalism in eq.2.23 is applied to $L + L'$ by considering also λ_k as independent variables. The result is a final system of Differential Algebraic Equations (DAE) with as many equations as the sum of Cartesian coordinates and Lagrangian multipliers, and although this method produces much bigger systems than the previous one, it is much easier to implement in a multi-body solver. More details on DAE systems of equation can be found in [59] and [60].

2.3.2. Simpack®

Simpack® is an extensive library of modelling elements with the capability of including flexible bodies in the MBD framework. This allows to analyze the global behaviour of complex multiphysics systems (mechatronic, hydraulic, aerodynamic etc.) with a high level of abstraction and in a very fast and efficient manner.

Relative-coordinate formalism

In Simpack®, the mathematical description of the system dynamics is carried out by employing the relative coordinates approach, which is a convenient and intuitive choice to deal with the kinematics of body chains. This approach significantly reduces the number of kinematic independent parameters by selecting only the minimal DoF that are actually present in the real system. In this way, only one body is referred to the inertial reference frame, while the motion of each one of the other bodies is calculated with respect to the motion of the previous one in the chain. Apart from the minimization of DoF, there are other advantages with this approach. For instance, position vectors remain small and simulation time is significantly reduced when performing time integration [61]. More information on the absolute-coordinate approach applied to MBD simulations can be found in [62].

time/frequency representation of rigid multi-body systems

After discussing kinematic aspects, the modelling of a MBD system continues with the derivation of a set of non-linear DAEs in the general case of constrained systems, similarly to what described in the previous section. Therefore, the full system of equations solved features first order ODEs for the kinematic position and velocity states. Besides, the equation for the velocity states is the Lagrange equation augmented with the Jacobian of constraints by means of the Lagrangian multipliers. The DAE system includes also algebraic equations and additional first order ODEs for the force states. In the case of unconstrained systems, the previous DAE system simplifies to a set of ODEs related to the only kinematic states. In addition to a time representation of the multi-body system, Simpack® can also perform calculations in the frequency domain to compute eigenvalues and, spectral system response. This firstly requires to linearize the system in the neighbourhood of an equilibrium condition. Next, a state-space representation by means of the linear time-invariant approach is used. On top of this formalism the frequency-domain analysis is carried out by means of dedicated solvers.

Time integration of the system

As the main purpose of every MBS tool is to compute the system response in time given initial conditions and under the effect of external forces or excitations, the most important solver is the one performing time-integration, which calculates the solution of the non-linear DAE system by means of either fixed or adaptive timestep numerical integrators. For stiff constrained systems, the *SODASRT2* solver is a Simpack® proprietary adaptation of the *DASSL* scheme[2] and it works very well almost for every situation,

[2] A time integrator to solve non-linear DAE systems by means of Backward Differentiation Formulae (BDF) schemes - implicit linear multi-step methods [63] - and the Newton-Raphson algorithm [64]. More details on this solver can be found in [65].

while for unconstrained systems, the adaptive *LSODE*[3] and a fixed timestep version of the Dormand-Prince couple of explicit Runge-Kutta methods (*DOPRI5* method) can provide good performance and better accuracy than less sophisticated schemes for larger timesteps. Alternatively, traditional basic schemes like classic Runge-Kutta or Euler methods can also be employed for simple problems. In all the cases of adaptive timestep integrators, the user can set appropriate tolerances that the solver tries to respect by changing timestep size and integration order during the simulation.

Most important modelling steps and elements

Simpack®, likewise traditional MBD tools, employs abstract modelling elements, primarily *Bodies*, *Joints*, *Constraints*, *Force Elements* and *Excitations* to model a real system. The main features of these modelling elements are introduced now with reference to the typical simulation setup workflow.

Step 1: Bodies, Primitives and Markers Whenever a new model is set up, the first step is to define *Bodies*, which can be either rigid or flexible. In both cases, mass and inertia properties must be either directly provided or calculated based on the geometric and material properties of the simulation body. The latter properties are specified by means of *Primitive* elements. When mass and inertia properties are calculated from *Primitive* shape and the *Body* material data, these will depend on the facetization of the discrete geometry and the prescribed material. *Primitives* are positioned in the computational space by exploiting *Markers*, which can be seen as both points and coordinate systems at the same time. In Simpack®, there is always an inertial reference system with a related *Marker*, called *M_Isys* and represented by the "ground" in the 2D visualization window of the tool. Besides, in the tool, every *Body* features its own *Body Reference Frame* (*BRF*), which stands for a default reference location defining a local body coordinate system. *Markers* are generally required to define the motion of each *Body* and the locations where *Force Elements* or *Excitations* are applied.

Step 2: Joints and Constraints or Connections *Joints* prescribe the motion of each *Body* by introducing DoF in terms of kinematic states and they should always be defined according to the rule that each *Body* must have one and only one *Joint*. The user can select a specific *Joint type* among those available in the related library, and every one of them is defined by setting a *From Marker* and a *To Marker*, outlining the order of the kinematic chain. When the simulation consists of one *Body*, the *From Marker* is always *M_Isys*. The contrary of *Joints* are *Constraints*, as they suppress the number of DoF in the system by adding algebraic equations of motion to the ODEs for the kinematic states of the system, therefore, they always work in parallel. However, the user can also decide to opt for *Connections*, namely another modelling element which allows to prescribe body-motion by accounting free and constrained motions at the same time. Similarly to *Joints*, both *Constraints* and *Connections* require the specification of the *From* and *To Marker*.

Step 3: Force Elements and Excitations Simpack® provides a very large library of *Force Elements* to model both internal forces, such as spring-damper elements, or external forces spanning a wide range of applications, from wind loads to bushing or gear pair elements consisting of tens of parameters. Also forces are based on the *From Marker* and the *To Marker* according to the Newtonian 3rd law, with the *From Marker* defining the "action" $\boldsymbol{F}$ and the *To Marker* the "reaction" $-\boldsymbol{F}$. On the other hand, *Excitations* are related to time or frequency dependent functions used to provide motion to driven *Markers* or to characterize a specific *Force Element*. In addition, *Expressions* can be employed by the user to arbitrarily define mathematical formulae based on the internal library of Simpack® functions, and *Input Functions* can be defined to generate laws $f(x)$ or $f(x, y)$ by interpolating data sets. Both previous modelling elements can be used to define *Excitations* and *Force Elements*.

Step 4: Solvers Once the model is set-up with its *Bodies*, *Joints*, *Constraints* and *Force Elements*, the user can simulate the behaviour of the system by running a specific solver. In Simpack®, all the available solvers can be used in both an *online* (real-time) or *offline* (in background) version. The first mode doesn't generate any result file and is usually employed to check the correct setup of the system, while the second one primarily returns binary result files which can be visualized and manipulated in the Simpack® Post-processor Graphic User Interface (GUI). In addition to the *Time Integration Solver*, the tool offers several frequency domain integrators, such as the *Eigenvalue Solver*, to run after the *Equilibrium Solver* which finds an equilibrium condition and then linearizes the system around that condition.

[3] The Livermoore Solver for ODEs is a numerical scheme able to solve both stiff and non-stiff systems by employing either Adams (explicit linear multi-step methods) or BDF schemes respectively [66].

Step 5: Post-processing All the previously mentioned modelling elements characterizing the typical simulation setup workflow are carried out in the Simpack® Pre-processor tool. This can be used as either a GUI or as a CLI to run scripts in both Windows and Linux environments. On the other hand, simulation result files are opened and manipulated in the Simpack® Post-processor, which is a handy tool for dynamic analysis and to generate clear reports. Also this tool is available in its GUI or CLI mode on both Linux and Windows.

Dedicated coupling functionalities

Simpack® is a tool that can be extensively exploited for establishing couplings with other tools due to its multi-physics vocation. Some couplings with commercial tools, such as Matlab and Simulink are already implemented in the software via dedicated interfaces and GUIs. Besides, the MBD tool is available coupled with external libraries for specific applications, such as the aerodynamic database AeroDyn [67] exploited to provide data for advanced force elements employed in full wind turbines MBS. Alternatively, the user can also setup an own coupling by exploiting two specific functionalities of the MBD tool: the co-simultion engine as coupling medium or interface and force elements based on user defined routines to apply the data from the external tool into the MBS.

Co-simulation engine The co-simulation engine is a specific setup of the time integration solver exploited to couple Simpack® with an external simulation tool, where either Simpack® or the external solver can be the server or client of the interaction [65]. During such an interaction, both clients run simultaneously and exchange data via *Input* and *Output Vector elements* at a fixed sampling period. These two objects require memory-sharing between the two computer programs and this methodology allows to efficiently exchange data at a very high frequency without introducing the computational cost due file input/output [13]. In a co-simulation engine, each one of the two clients carries out a specific task in the coupled system and is configured within its respective platform, while during the co-simulation, data are exchanged via TCP/IP interface [68].

User defined routines for force elements Regardless of the coupling medium between Simpack® and the external tool, after coupling the two computer programs, the problem of how to apply the data provided by the external tool in the MBD simulation still needs to be addressed. Often, especially in the case of 3D flexible geometries, this issue is solved by implementing dedicated force elements, if those available in the library can not be exploited. This requires the coding of user defined routines in the Fortran 90 programming language. These routines exploit existing Simpack® subroutines (called "Access Functions") to access modelling element, control solvers and manipulate all possible simulation parameters [69]. User routines are very powerful tools that allows for deep customization and extension of the MBD tool capabilities to specific applications and can be used to create new types of all possible modelling elements, not only forces.

Potential coupling functionalities

In addition to the previous coupling features, other Simpack® functionalities can be potentially exploited for couplings. Reasons for considering alternative strategies are primarily related to limitations in the external tool, which might not be suited for memory-sharing-based couplings. Besides, a significant investment of time, resources and expertise are required to effectively and robustly implement co-simulation engines and user-routine-based force elements. This is also a crucial aspect, given the relatively short duration of a master thesis. For these reasons, existing Simpack® features not specifically meant for couplings can also be considered given their robustness, simplicity of use and immediate availability. In this section, two features are discussed: input functions defined from table reading and scripting with the purpose of understanding them and exploring their possibilities.

Table reading The equivalent of PowerFLOW® table import can be achieved in Simpack® by reading *.afs* files to provide to the *Input Functions*, which can then be used to define other modelling elements up to *Force elements* as described previously. These files are text files with a specific format, consisting of datasection and header, and syntax accurately described in [65]. The header provides information for the automatic reading and generation of input functions to apply to the MBD simulation setup. Several possibilities for structuring data in these files are available, but the most used are those defining input functions of the type $f(x)$ or $g(x, y)$, with x for example a time variable and y a position coordinate. These functions can are then be interpolated by the solver during the simulation with linear quadratic or spline interpolations. In the perspective of the coupling with a CFD tool, this feature can be exploited to provide to the MBD solver a force history from the aerodynamic solver, which can than be used to define an excitation-based force element.

Scripting languages Another useful and versatile feature of Simpack® are the scripting languages exploited by Simpack® Pre and Post-processors for automatic case setup, running solvers or to speed up recurrent post-processing steps [70]. The scripting languages employed are QtScript for the Pre-processor and QSA Script for the Post-processor, both based on ECMAScript, which is a standard for the JavaScript language [71]. Simpack® allows to generate these scripts by directly recording operations carried out on the GUI, modifying them with a built-in editor and executing them both via the GUI or in background by means of the CLI version of the tool. Always considering a coupling with a CFD tools, scripting is necessary given the potentially large number of coupling iterations. This requires the full automatization of the MBD process, namely *.afs* table import of the CFD airload, correct application of the airload to the MBD simulation model, execution of the time integration and conversion of binary results to an ASCII file that can then be returned back to PowerFLOW®.

2.4. Unsteady aerodynamics

2.4.1. Overview

Unsteady aerodynamic phenomena are frequent and play a significant role in the overall behaviour of wind turbines and rotorcraft. Therefore, it is firstly necessary to introduce the main sources of unsteadiness in both applications, because, depending on the properties of the physical phenomenon, different approaches are employed to model the unsteady effects that such sources generate. Finally, after introducing the key concept of reduced frequency, the common features and the main analytical models of incompressible unsteady aerodynamics can be discussed

Sources of unsteadiness and prediction approaches

When dealing with the wide range of unsteady phenomena affecting rotorcraft, one of the main sources of unsteadiness are AoA excursions due to blade elastic motion and pitch control inputs; besides, also the induced velocities caused by the trailed and shed vorticity, as well as the concentrated tip vortices, can cause large and fast AoA variations [7]. It is important to mention also that unsteady effects are frequent in the retreating side of the rotor near the blade root, where reverse flow conditions occur during forward flight conditions due to the composition of the freestream and rotation velocity. In the case of a wind turbine, AoA fluctuations are also caused by the elastic deformations of long slender modern blade designs; in addition, the turbulent and stochastic wind inflow conditions, as well as the interference effects due to tower shadow or surrounding turbines in a wind farm facility, play an important role [72]. Furthermore, in both scenarios, unsteady effects can be further split into small and large AoA variations, because this has a significant impact on the models and prediction methods employed to investigate such time-dependent phenomena. When the AoA fluctuations overcome the static stall limit, dynamic stall can occur with a flow field dominated by time-dependent separation resulting in large abrupt amplitude and phase variations (wide hysteresis loops) in the airload that can lead to severe aeroelastic effects [73]. Due to the importance of viscous phenomena for separated flows and the complex non-linear behaviour of dynamic stall, its analysis is often carried out by solving the full Navier-Stokes equations by means of CFD, while predictive models are still an open-end matter in the research community [7]. On the other hand, when small AoA variations (small compared to the static stall angle) and attach flow conditions occur, the resulting phenomena can be considered as limited amplitude and phase oscillations with respect to the quasi-steady aerodynamic forces [7]. Such conditions allow to effectively employ analytical or semi-analytical models with a significantly reduced computational cost compared to CFD. Finally, an important difference in the modelling of unsteady aerodynamic effects between wind turbines and rotorcraft is related to the increased importance of compressibility effects for the latter, especially during high speed forward flights. This makes predictions more complex both for CFD simulations of high AoA dynamic stall and semi-analytical method for attached flows. In the second case, it is very difficult to derive exact analytical solutions for subsonic conditions and for the entire time domain, like in the case of incompressible flows, therefore numerical approximate methods have to be exploited [7].

Reduced frequency

Steady aerodynamic forces can be described by means of only two dimensionless parameters (Reynolds and Mach numbers) for fixed body orientation and shape according to the Buckingham Π-theorem [74].

However, in the case of unsteady airload, an additional dimensionless parameter, called reduced frequency (k), naturally emerges from the dimensional analysis, leading to the below formulation (from [7]):

$$\frac{F}{\rho U_\infty^2 b} = f\left(\frac{\rho U_\infty c}{\nu}, \frac{U_\infty}{a_\infty}, \frac{\omega b}{U_\infty}\right) = f(Re, Ma, k) \tag{2.24}$$

where U_∞, ρ, a_∞ and ν are the free-stream flow velocity, density, sound velocity and kinematic viscosity. The above equation suggests also that the reduced frequency (a ratio between the frequency of the unsteady phenomenon $\omega = 2\pi f$, the airfoil semi-chord b and the inflow velocity U_∞) is a qualitative parameter to estimate the degree of unsteadiness of the airfoil aerodynamics. Therefore, if $k = 0$, the airload is steady, if $0 < k < 0.05$ there are quasi-steady conditions, if $k > 0.05$ proper unsteady flows are considered and highly unsteady behaviour occur when $k > 0.2$. According to [7], the reduced frequency should only be used carefully, as it might lead to ambiguity given the spanwise variation of the local flow velocity in both helicopters and wind turbines. However, this parameter is still used a lot in airfoil applications and is also a fundamental input parameter for several analytical models, especially those in the frequency domain.

Prediction methods for attached flows

Regarding attached unsteady flows under small variations of AoA and featuring incompressible attached flow conditions, elegant analytical models were derived from the Glauert theory for steady thin airfoils providing closed-form solutions in either time or frequency domain. These methods are often known as classical unsteady aerodynamic methods and were developed for aircraft applications back in the first half of the XX century to model the time-varying lift and pitching moment of airfoils undergoing several types of motions. All these methods quantify the unsteady effects as phase lags between the input forcing function and the corresponding response primarily by means of the airfoil shed vorticity and the reduced frequency of the phenomenon. By considering the time domain, an important example is the Wagner model [75] which computes the indicial response of an airfoil due to a step change in AoA. This model is also used to compute the response to a generic excitation by means of the Duhamel integral superposition approach. Besides, the Küssner model ([76] and [77]) can be used to compute an indicial response in the case of an airfoil entering a sharp-edged vertical gust. In the frequency domain, the Theodorsen model [78] is one of the most employed methods, as it describes the unsteady airload due to both a harmonically pitching and plunging motion, where the trailed wake is convected downstream. An improvement over this method for rotor application is the Loewy model ([79] and [80]) which accounts for the returning wake from previous rotations. Finally, in the next sections, the Theodorsen model is analyzed more in detail (in addition to the Quasi-steady and the Steady Glauert theories) as it is directly employed for the verification activity of the MBD simulation setup of Simpack®, as discussed in ch.3.

2.4.2. Steady Glauert

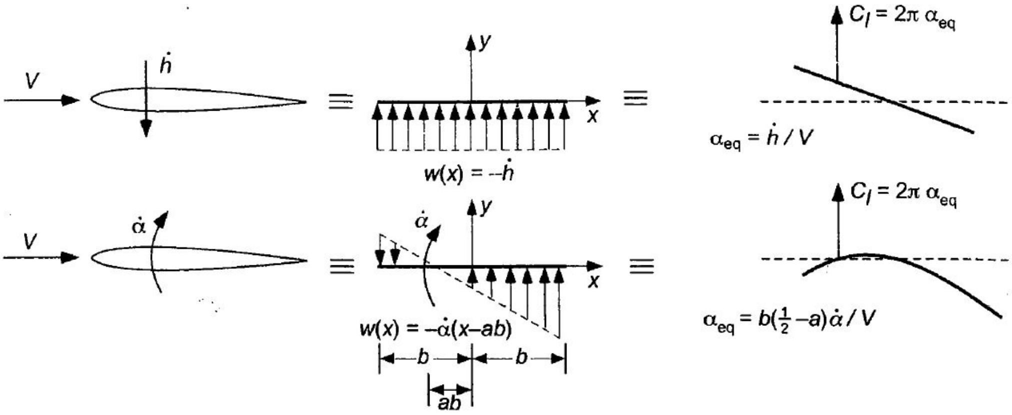
The Glauert model is not an unsteady aerodynamic model, but it serves as introduction for the true time-dependent models, since its discussion is relevant for this thesis project. As already mentioned, the steady Glauert model is based on the potential airflow theory developed by Glauert himself and others around the 1920s [81], and it does not account for viscosity in the flow. However, this is not a big limitation for simple aeroelastic problems with attached flows and small displacements. The steady Glauert model is based on a direct proportionality between the AoA and the airload generated by the airfoil which is summarized by the famous $c_l = 2\pi\alpha$ relation for a symmetric airfoil. In the case of a dimensional lift L and aerodynamic moment about the quarter chord $M_{c/4}$, the following expressions are provided by the model to define the airload:

$$L = \frac{1}{2}\rho U_\infty^2 c\, c_{l_\alpha} \alpha = \frac{1}{2}\rho U_\infty^2\, c 2\pi\, \alpha \qquad \text{and} \qquad M_{c/4} = M_{AC} = 0 \tag{2.25}$$

Other important results of this fundamental theory are the quarter chord location for both the Aerodynamic Center (AC) and pressure center (application point of the resultant of normal pressure stresses) for a symmetric airfoil, which leads to the $M_{c/4} = 0$ expression above, and the aft position of the pressure center with respect to the AC for cambered airfoils, besides to an offset in the lift polar for $\alpha = 0$.

2.4.3. Quasi-steady

The main limitation of the Glauert model has to do with its intrinsic steady nature, which is incompatible with the inherently unsteady properties of most of the aeroelastic phenomena. Therefore, the second aerodynamic model can be considered as an intermediate one between pure steady and unsteady models, namely the quasi-steady one. This model originates from the Glauert theory and it additionally models the unsteadiness of the aerodynamic load by considering the time derivatives of the pitch and plunge motion (h). More in detail, when the airfoil is moving, the equivalent AoA α_{eq} that it experiences is the sum of the geometric AoA (α) and a contribution due to the movement of the airfoil itself. As, shown in fig.2.8, if the motion is a pure vertical translation, the equivalent AoA would be equal to $\alpha + \dot{h}/U_\infty$ and the same for all the points of the airfoil. However, when a rotation is considered, then the equivalent AoA depends on $\dot{\alpha}$ and is not the same everywhere, as it changes linearly with a zero value at the location of the semichord-scaled pitching axis location a in the figure. Even though the quasi-steady model is a more sophisticated model than the steady Glauert one, as it includes unsteadiness in both the translation and the rotation DoF, it does not take into account the dynamics of the near shed vorticity. The latter is an important quantity as it plays a crucial role on the loads acting on the airfoil due to the induction velocity that the shed vorticity determines. To account for the wake effect, even more sophisticated models are required, like the Theodorsen one considered in detail next. By employing the Chasles theorem [58], the velocity v_P (position x_P) of the pitching axis of the airfoil (point P) can be defined as: $\dot{v}_P = \dot{h} + \dot{\alpha} x_P$. This leads to the following expression for the equivalent AoA for a pitching and plunging airfoil:



V
$\dot{h}$
$\equiv$
y
x
$\equiv$
$C_l = 2\pi\,\alpha_{eq}$
$w(x) = -\dot{h}$
$\alpha_{eq} = \dot{h}/V$
V
$\dot{\alpha}$
$\equiv$
$\dot{\alpha}$
y
x
$\equiv$
$C_l = 2\pi\,\alpha_{eq}$
$w(x) = -\dot{\alpha}(x-ab)$
b
b
ab
$\alpha_{eq} = b(\frac{1}{2}-a)\dot{\alpha}/V$

Figure 2.8: Equivalent AoA for a purely plunging (top) and pitching (bottom) airfoil modelled as a velocity perturbation and an induced camber respectively [7].

$$\alpha_{eq} = \alpha + \frac{\dot{h}}{U_\infty} + \frac{\dot{\alpha} x_P}{U_\infty} \tag{2.26}$$

With the above expression for the equivalent AoA, the only difference between the Glauert and the quasi-steady model is the replacement of α with an α_{eq} containing also the dynamic states of the airfoil motion, leading to:

$$L = \frac{1}{2}\rho U_\infty^2 c\, c_{l_\alpha} \alpha_{eq} = \frac{1}{2}\rho U_\infty^2 c\, 2\pi \left(\alpha + \frac{\dot{h}}{U_\infty} + \frac{\dot{\alpha} x_P}{U_\infty} \right) \quad \text{and} \quad M_{AC} = 0 \tag{2.27}$$

2.4.4. Theodorsen

The third unsteady aerodynamic model considered is the Theodorsen one. Its description starts by the modelling of the airfoil and its wake in terms of vortex sheets - due to the small magnitude of pitching and plunging motion considered - with the wake extending from the trailing edge up to infinity downstream, as shown in fig.2.9. The wake consists of counter-rotating vortices - due to either the upstroke or downstroke phases of airfoil motion - being convected with the free-stream velocity. Besides, the planar wake sheet is also force-free, which means that no net pressure jump is allowed over the sheet, and the value of wake vorticity has to be minus the bound vorticity to satisfy the Kalvin's theorem at every time. The bound vorticity over the airfoil is instead allowed to provide a pressure difference, leading to a lift force.

The goal of the Theodorsen model is to compute the bound vorticity by solving an integral equation, where the former is combined with the shed vorticity providing the downwash effect acting over the airfoil surface. This equation requires the Kutta condition $\gamma_b(x = c, t) = 0$ to prescribe zero bound vorticity

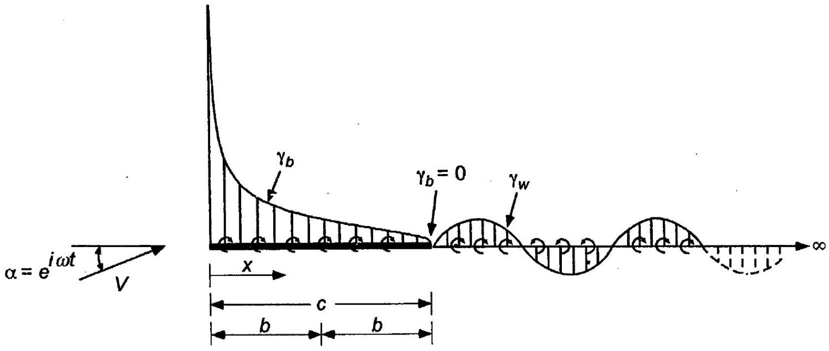


Figure 2.9: Modelling of the airfoil bound circulation γ_b and of the wake shed vorticity γ_w for a harmonically oscillating airfoil $\alpha = e^{i\omega t}$ according to the Theodorsen model [7].

at the trailing edge - this is necessary to make sure that the flow remains attached at that location - and the Kelvin's theorem for the conservation of circulation. The solution of this equation was found by Theodorsen by considering harmonic motion for the pitching and plunging DoFs, and it consists of a transfer function between the input motion and the output aerodynamic load. The result is shown below for the lift and the aerodynamic moment about the quarter chord in the case of generic semichord-scaled pitching axis location $\tilde{a}$[4]: [78].

$$L = L_{nc} + L_c = \pi\rho b^2\Big(\ddot{h} - \tilde{a}b\ddot{\alpha} + U_\infty\dot{\alpha}\Big) + 2\pi\rho U_\infty b\,C(k)\Big(\dot{h} + b\Big(\frac{1}{2} - \tilde{a}\Big)\dot{\alpha} + U_\infty\alpha\Big) \tag{2.28}$$

$$M_{AC} = M_{AC}^{(nc)} = -\pi\rho b^3\Big(\frac{\ddot{h}}{2} + \frac{b}{2}\Big(\frac{1}{4} - \tilde{a}\Big)\ddot{\alpha} + U_\infty\dot{\alpha}\Big)$$

The non-circulatory ("nc" terms above) or apparent-mass-effect terms are quite similar for the Theodorsen and the quasi-steady model, as both take into account the forces required to accelerate the flow near the airfoil surface. On the other hand, the significant difference between the two models is in the wake vorticity effect, which is neglected by the quasi-steady one. As the circulation over the airfoil changes in time, also the shed vorticity changes. This determines a downwash velocity modifying the airfoil surface loads according to the information generated by the airfoil itself in the past. Such memory-effect of the shed vorticity is accounted by the circulatory component ("c" term above) of the force by means of the lift-deficiency function $C(k)$. The name lift-deficiency can be motivated by saying that the main effect of the wake induction is that of delaying the build-up processes of the airload [82], and this is clearly visible from the increasing size of hysteresis loops for increasing k occurring in the dynamic lift-polar of a purely pitching airfoils. The term $C(k)$ is also known as Theodorsen function and is a complex function of the reduced frequency $k = \omega b/U_\infty$ with the following expression:

$$C(k) = \frac{H_1^{(2)}(k)}{H_1^{(2)}(k) + jH_0^{(2)}(k)} \tag{2.29}$$

where the Hankel functions of second kind and order n $H_n^{(2)}$ with $n = 0, 1$ are employed. In addition, this specific type of Hankel functions depend on the Bessel functions of the first (J_n) and second (Y_n) kind of order n, namely $H_n^{(2)} = J_n - jY_n$.

According to [83], the Theodorsen model is appreciated for its derivation from fundamental principles by means of clear assumptions and the fact that it is composed of terms which have a clear physical correspondence, like the added-mass, the quasi-steady term multiplying the Theodorsen function and the latter modelling the wake-effect. However, it might not be the best approach for low Re number flows (micro aerial vehicles) and its intrinsic frequency-dependent nature limits its applicability to time-domain control and aeroelasticity. Therefore, the appendix sec.B.1 discusses general methods to approximate the generalized Theodorsen function $C(k)$ in eq.2.29 and applies that to derive a complete time domain formulation of an aeroelastic system, as discussed also in ch.3.

[4] The quantity $\tilde{a}$ measures the pitching axis location from from the mid-chord position, therefore, a pitching axis location at the quarter chord leads to a value for this quantity $\tilde{a} = -1/2$.

3

Simpack® MBD setup verification

The first major part of the thesis project has been devoted to the understanding and usage of the MBD simulation tool Simpack® on simple 2D aeroelastic problems. Such physical systems involve airfoils undertaking plunge and pitch motions under the effect of lumped elastic forces/torques and prescribed airloads from analytical unsteady aerodynamic models. As shown in fig.3.1, the main hypothesis of work of this activity is to assess the degree of accuracy of the MBD tool in computing kinematic states - $x(t)$, $\dot{x}(t)$ and $\ddot{x}(t)$ - compared against those returned by full aeroelastic analytical models. As the airload - $F(t)$ - provided to Simpack® is the same as the one employed by the aeroelastic system, it is possible to focus only on the solution of the structural dynamic part of the simulation. This verification is fundamental in the process of developing of a coupling between PowerFLOW® and Simpack®, as the simulation setup of both tools needs to be validated separately to understand their intrinsic limitations. Being said this, the structure of this chapter consists in four main section, starting with the description of the derivation of the reference full aeroelastic analytical model in sec.3.1. Next, the numerical solutions of the aeroelastic systems related to the different unsteady aerodynamic models employed is discussed in sec.3.2. This section is followed by a detailed description of the the MBD simulation setup in sec.3.3. Finally, in sec.3.4, the results of the Simpack® solutions are compared with the reference aeroelastic model and discussed in terms of their accuracy, description of their physical properties and in terms of the main findings in view of the coupling with PowerFLOW®.

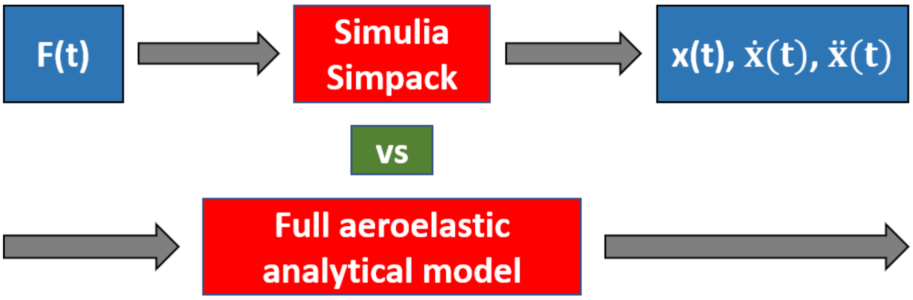


Figure 3.1: Diagram showing the main approach followed in this verification activity of the Simpack® MBD simulation setup for the investigation of 2D aeroelastic problems involving plunging and pitching airfoils.

3.1. Analytical aeroelastic model

3.1.1. Problem definition

First of all, it is necessary to point out that the aeroelastic problem considered and the derivation of the equations for all the unsteady aerodynamic models employed is based on the work of [84]. The problem investigated consists of a 2D semi-rigid airfoil with lumped parameters. Two degrees of freedom are considered[1]: plunge translation h_O - which is defined as positive downwards - and pitch rotation α, which is considered as positive clockwise. The calculated rotation and translation motions are those of

[1] The horizontal degree of freedom can be neglected, at a first instance, since blades and wings are typically much stiffer along the chordwise direction compared to the chord-normal direction. This is a frequent assumption in preliminary modeling of aeroelastic effects for a 2D airfoil.

point O, which is the origin of the airfoil reference system. Regarding the geometry of the body, the airfoil is actually a flat plate, modelled as a line in the analytical model and a zero-thickness plate in Simpack®. The elasticity of the airfoil section is concentrated into the rotational (k_α) and the translation (k_h) spring stiffness coefficients acting on the two allowed directions of motion. The CoG is point G and lies in the middle of the plate, the elastic center is point E and represents the location where both springs are located[2]. Finally, the AC is defined at the quarter-chord of the airfoil (point A), where aerodynamic forces are applied. Such airload can be further split into a vertical lift force F and a moment M_A. As the horizontal motion is locked, the horizontal component of the aerodynamic force (i.e. drag) is not considered. For the sake of reducing the analytical complexity in the formulation of the aerodynamic forces - especially in the case of the Theodorsen model, the pitching axis of the airfoil is considered at the quarter chord location leading to O≡A and $\tilde{a} = -1/2$ in eq.2.28. In the derivation of the EoMs, a general reference location O is still employed.

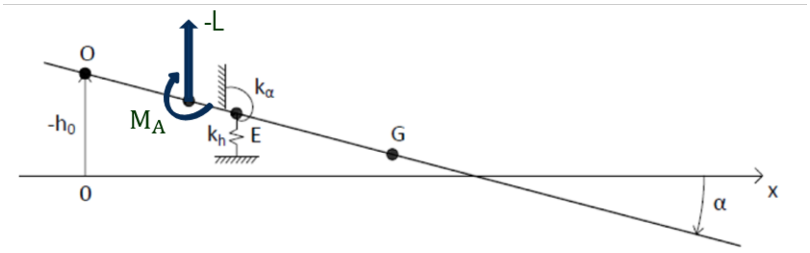


Figure 3.2: Representation of the analytical model for the 2 DoFs aeroelastic airfoil problem considered in this investigation.

3.1.2. Derivation of the equations of motion

The problem investigated deals with a straight line airfoil which can vertically translate (plunge motion h_O) and rotate (pitch motion α) under the action of elastic and aerodynamic forces. Besides, the angle α is assumed to be small, this way the sine and cosine functions resulting from the mathematical description of its kinematics can be linearized. In this section, the goal is to derive the equations of motion of the aeroelastic system. The derivation begins by describing the kinematics of the system in terms of equations. Therefore, by applying the Chasles theorem [58], the velocity of the center of gravity $\boldsymbol{v}_G$ can be computed with respect to the velocity $\boldsymbol{v}_O$ of the reference point O. The derivation is the following:

$$\boldsymbol{v}_G = \boldsymbol{v}_O + \dot{\boldsymbol{\alpha}} \times (\boldsymbol{G} - \boldsymbol{O}) = -\dot{h}_O \hat{\boldsymbol{y}} + \dot{\boldsymbol{\alpha}} \times x_G\big(-\sin(\alpha)\hat{\boldsymbol{x}} - \cos(\alpha)\hat{\boldsymbol{y}}\big) \approx -(\dot{h}_O + x_G\dot{\alpha})\hat{\boldsymbol{y}} \tag{3.1}$$

where $\boldsymbol{G}$ and $\boldsymbol{O}$ are the position vectors of points G and O with respect to the inertial reference system, whose unit versors of x- and y-axis are $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}$; besides, x_G is the distance between both points when measured from point O on the airfoil. Once the kinematics is defined, the equations of motion of the system can be derived by employing the Lagrangian formalism, as it is the most convenient method for hand derivations of systems with n multiple DoFs. This leads to:

$$\frac{\partial}{\partial t}\left(\frac{\partial T}{\partial \dot{q}_n}\right) - \frac{\partial T}{\partial q_n} + \frac{\partial U}{\partial q_n} = Q_n \tag{3.2}$$

In the above equation, q_n is the n-th generalized coordinate, Q_n is the n-th component of the non-conservative generalized forces, T is the kinetic energy of the system and U is the potential one. By exploiting the previously computed velocity of the center of gravity, the kinetic energy is equal to:

$$T = \frac{1}{2}m\ \boldsymbol{v}_G \cdot \boldsymbol{v}_G + \frac{1}{2}J_G\,\dot{\boldsymbol{\alpha}} \cdot \dot{\boldsymbol{\alpha}} = \frac{1}{2}m\big(\dot{h}_O + \dot{\alpha}x_G\big)^2 + \frac{1}{2}J_G\dot{\alpha}^2 \tag{3.3}$$

where J_G is the moment of inertia of the airfoil around an axis perpendicular to the x-y plane and passing through point G. Regarding the potential energy, this equals to:

$$U = \frac{1}{2}k_\alpha\alpha^2 + \frac{1}{2}k_h\big(h_O + \alpha x_E\big)^2 \tag{3.4}$$

[2] In general, when the CoG and the AC are different, this returns a full mass matrix and coupled torsional-bending modes, because of the interaction between inertial and torsional stiffness moments.

The generalized components of the non conservative forces for both h_O (Q_1) and α (Q_2) coordinates can be computed, always with respect to point O, based on the Virtual Work Principle as:

$$Q_n = \boldsymbol{F} \cdot \delta \boldsymbol{h_O} + (\boldsymbol{M_A} + x_A \boldsymbol{F}) \cdot \delta \boldsymbol{\alpha} \qquad \rightarrow \qquad Q_1 = F \qquad Q_2 = M_A + x_A F \tag{3.5}$$

with $\boldsymbol{F}$ the force vector along the y-direction and $\boldsymbol{M}_A$ the aerodynamic moment about point A and perpendicular to the plane. After applying the derivatives involved by the Lagrangian formalism to the previously defined quantities, the following system of equations is obtained:

$$\begin{cases} m\ddot{h}_O + mx_G\ddot{\alpha} + k_h h_O + x_E k_h \alpha = F \\ mx_G\ddot{h}_O + (mx_G^2 + J_G)\ddot{\alpha} + x_E k_h h_O + (k_\alpha + k_h x_E^2)\,\alpha = M_A + x_A F \end{cases}$$

Finally, the above system can be re-written in a more handy matrix notation as:

$$\begin{bmatrix} m & mx_G \\ mx_G & J_O \end{bmatrix} \begin{bmatrix} \ddot{h}_O \\ \ddot{\alpha} \end{bmatrix} + \begin{bmatrix} k_h & x_E k_h \\ x_E k_h & k_\alpha + k_h x_E^2 \end{bmatrix} \begin{bmatrix} h_O \\ \alpha \end{bmatrix} = \begin{bmatrix} F \\ M_A + x_A F \end{bmatrix} \qquad \rightarrow \qquad \boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{K}\boldsymbol{x} = \boldsymbol{f} \tag{3.6}$$

where $J_O = mx_G^2 + J_G$ according to the Huygens-Steiner's theorem, $\boldsymbol{M}$ is the mass matrix, $\boldsymbol{K}$ is the stiffness matrix, $\boldsymbol{f}$ is the aerodynamic force vector and $\boldsymbol{x}$ is the vector of kinematic states. Eq.3.6 represents the analytical model for the structural part of the aeroelastic problem where the airload is considered as an external forcing term, and therefore as a rhs term.

3.1.3. Unsteady aerodynamic models employed

To model the effect of the aerodynamic forces on the previously discussed structural system, the three unsteady aerodynamic models are considered with an increasing complexity and accuracy in capturing the real aerodynamic behaviour of the airfoil undergoing pitch and plunge motions.

Steady Glauert

When the steady airload according to the Glauert model in sec.2.4.2 is considered, the following aerodynamic force vector is obtained:

$$F = -L = -\frac{1}{2}\rho U_\infty^2 2bc_{l_\alpha}\alpha \qquad \text{and} \qquad M_A = 0 \qquad \rightarrow \qquad \boldsymbol{f} = \begin{bmatrix} -\frac{1}{2}\rho U_\infty^2 2bc_{l_\alpha}\alpha \\ -x_A\frac{1}{2}\rho U_\infty^2 2bc_{l_\alpha}\alpha \end{bmatrix} \tag{3.7}$$

By including the above formulation for $\boldsymbol{f}$ into the structural system in eq.3.6, the following steady aeroelastic problem can be defined. Due to the formulation of the airload with a lift force dependent on the pitch position, $\boldsymbol{f}$ is included directly in the structural left-hand-side (lhs) part of the system, leading to a homogeneous system of equations. Besides, the moment of inertia J_O is referred to point E always by employing the Huygens Steiner's theorem.

$$\begin{bmatrix} m & mx_G \\ mx_G & J_E - mx_E^2 + 2mx_E x_G \end{bmatrix} \begin{bmatrix} \ddot{h}_O \\ \ddot{\alpha} \end{bmatrix} + \begin{bmatrix} k_h & x_E k_h \\ x_E k_h & k_\alpha + k_h x_E^2 \end{bmatrix} \begin{bmatrix} h_O \\ \alpha \end{bmatrix} + \begin{bmatrix} 0 & \frac{1}{2}\rho U_\infty^2 2bc_{l_\alpha} \\ 0 & x_A\frac{1}{2}\rho U_\infty^2 2bc_{l_\alpha} \end{bmatrix} \begin{bmatrix} h_O \\ \alpha \end{bmatrix} = \boldsymbol{0} \tag{3.8}$$

At this point, it is possible to proceed to the non-dimensionalization of the equations of motion by considering the following scaling rules:

$$h = \frac{h_O}{b}; \quad \hat{t} = \omega_\alpha t; \quad a = \frac{m}{\rho b^2 c_{l_\alpha}}; \quad \hat{U} = \frac{U_\infty}{b\omega_\alpha}; \quad \Omega^2 = \frac{\omega_h^2}{\omega_\alpha^2} \quad \text{with} \quad \omega_h^2 = \frac{k_h}{m};$$

$$\xi_E = \frac{x_E}{b}; \quad \xi_G = \frac{x_G}{b}; \quad \xi_A = \frac{x_A}{b}; \quad r_\alpha^2 = \frac{k_\alpha}{mb^2\omega_\alpha} \tag{3.9}$$

where h is the dimensionless pitch motion, $\hat{t}$ the dimensionless time, ω_α the pitch angular velocity, a is a dimensionless mass, $cl_\alpha = 2\pi$ is the aerodynamic lift derivative as introduced already in sec.2.4.2, $\hat{U}$ is dimensionless freestream velocity, Ω is the ratio between the pitch and the plunge (ω_h) angular velocities, ξ_A, ξ_G and ξ_E are the dimensionless x-coordinates of the AC, CoG and elastic center and r_α is finally a dimensionless rotational stiffness. When such scaling rules in eq.3.9 are applied to the previous steady dimensional system in eq.3.8, they return:

$$\begin{bmatrix} m & m\xi_G b \\ m\xi_G b & r_\alpha^2 mb^2 - m\xi_E^2 b^2 + 2m\xi_E\xi_G b^2 \end{bmatrix} \begin{bmatrix} \ddot{h} b\omega_\alpha^2 \\ \ddot{\alpha}\omega_\alpha^2 \end{bmatrix} + \begin{bmatrix} \Omega^2 m\omega_\alpha^2 & \xi_E b\Omega^2 m\omega_\alpha^2 \\ \xi_E b\Omega^2 m\omega_\alpha^2 & r_\alpha^2 mb^2\omega_\alpha^2 + \Omega^2 m\omega_\alpha^2 \xi_E^2 b^2 \end{bmatrix} \begin{bmatrix} hb \\ \alpha \end{bmatrix} +$$

$$+ \begin{bmatrix} 0 & \frac{1}{2}\rho\hat{U}^2 b^2\omega_\alpha^2 2bc_{l_\alpha} \\ 0 & b\xi_A \frac{1}{2}\rho\hat{U}^2 b^2\omega_\alpha^2 2bc_{l_\alpha} \end{bmatrix} \begin{bmatrix} hb \\ \alpha \end{bmatrix} = \mathbf{0}$$

Next, by dividing the first equation by $mb\omega_\alpha^2$ and the second one by $mb^2\omega_\alpha^2$, the final non-dimensional system for the steady Glauert model is obtained:

$$\begin{bmatrix} 1 & \xi_G \\ \xi_G & r_\alpha^2 - \xi_E^2 + 2\xi_E\xi_G \end{bmatrix} \begin{bmatrix} \ddot{h} \\ \ddot{\alpha} \end{bmatrix} + \Omega^2 \begin{bmatrix} 1 & \xi_E \\ \xi_E & \frac{r_\alpha^2}{\Omega^2} + \xi_E^2 \end{bmatrix} \begin{bmatrix} h \\ \alpha \end{bmatrix} + \hat{U}^2 \begin{bmatrix} 0 & \frac{1}{a} \\ 0 & \frac{\xi_A}{a} \end{bmatrix} \begin{bmatrix} h \\ \alpha \end{bmatrix} = \mathbf{0} \tag{3.10}$$

Finally, the above homogeneous system can be rewritten in a compact form in terms of the dimensionless structural mass ($\boldsymbol{M^*}$) and stiffness ($\boldsymbol{K^*}$) matrices, the vector of the non-dimensional DoFs $\boldsymbol{\hat{x}}$ and a dimensionless aerodynamic mass matrix $\boldsymbol{K_a^*}$ corresponding to the matrix in the third product of the above equation:

$$\boldsymbol{M^*}\ddot{\boldsymbol{\hat{x}}} \;+\; \Omega^2\boldsymbol{K^*}\boldsymbol{\hat{x}} \;+\; \hat{U}^2\boldsymbol{K_a^*}\boldsymbol{\hat{x}} \;=\; \mathbf{0} \tag{3.11}$$

For reasons related to the calculation of both the analytical and the numerical solution of the problem, it is required to convert the second-order ODEs for h and α into a system of four first-order differential equations in time for the position and velocity states of the plunging and pitching DoFs. This type of formulation of the system is called state-space-model and can be obtained by introducing the velocity state vector $\boldsymbol{y}$:

$$\boldsymbol{y} = \dot{\boldsymbol{\hat{x}}} \;\rightarrow\; \begin{cases} \boldsymbol{M^*}\dot{\boldsymbol{y}} + \Omega^2\boldsymbol{K^*}\boldsymbol{\hat{x}} + \hat{U}^2\boldsymbol{K_a^*}\boldsymbol{\hat{x}} = \mathbf{0} \\ \dot{\boldsymbol{\hat{x}}} - \boldsymbol{y} = \mathbf{0} \end{cases} \quad\rightarrow\quad \begin{bmatrix} \boldsymbol{M^*} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{y}} \\ \dot{\boldsymbol{\hat{x}}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \Omega^2\boldsymbol{K^*} + \hat{U}^2\boldsymbol{K_a^*} \\ -\boldsymbol{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{\hat{x}} \end{bmatrix} = \mathbf{0}$$

Which can be re-written again in a more compact form as:

$$\boldsymbol{M_{tot}^*}\dot{\boldsymbol{z}} + \boldsymbol{K_{tot}^*}\boldsymbol{z} = \mathbf{0} \quad\rightarrow\quad \dot{\boldsymbol{z}} = \left(-\boldsymbol{M_{tot}^*}^{-1}\boldsymbol{K_{tot}^*}\right)\boldsymbol{z} = \boldsymbol{A_s}\,\boldsymbol{z} \tag{3.12}$$

It is apparent that the solution in time for the position and velocity states of the aeroelastic system consists in computing a final four-by-four matrix $\boldsymbol{A_s}$ equal to the matrix product between the inverse of the total mass matrix $\boldsymbol{M_{tot}^*}$ and the total stiffness matrix $\boldsymbol{K_{tot}^*}$.

Quasi-steady

The derivation of the second aeroelastic model, starts by recalling the quasi-steady formulation of the airload in sec.2.4.3 leading to the following aerodynamic force vector $\boldsymbol{f}$:

$$F = -L = -\frac{1}{2}\rho U_\infty^2 2bc_{l_\alpha}\alpha_{eff} \quad \text{and} \quad M_A = 0 \quad\rightarrow\quad \boldsymbol{f} = \begin{bmatrix} -\frac{1}{2}\rho U_\infty^2 2bc_{l_\alpha}\left(\alpha + \frac{\dot{h}_O}{U_\infty} + \frac{\dot{\alpha}x_P}{U_\infty}\right) \\ -x_A\frac{1}{2}\rho U_\infty^2 2bc_{l_\alpha}\left(\alpha + \frac{\dot{h}_O}{U_\infty} + \frac{\dot{\alpha}x_P}{U_\infty}\right) \end{bmatrix} \tag{3.13}$$

By plugging in the expression for the airload vector $\boldsymbol{f}$ into the structural system, the following dimensional system for the quasi-steady aeroelastic problem is obtained:

$$\begin{bmatrix} m & mx_G \\ mx_G & J_E - mx_E^2 + 2mx_Ex_G \end{bmatrix} \begin{bmatrix} \ddot{h}_O \\ \ddot{\alpha} \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\rho U_\infty^2 2bc_{l_\alpha}\frac{1}{U_\infty} & \frac{1}{2}\rho U_\infty^2 2bc_{l_\alpha}\frac{x_P}{U_\infty} \\ x_A\frac{1}{2}\rho U_\infty^2 2bc_{l_\alpha}\frac{1}{U_\infty} & x_A\frac{1}{2}\rho U_\infty^2 2bc_{l_\alpha}\frac{x_P}{U_\infty} \end{bmatrix} \begin{bmatrix} \dot{h}_O \\ \dot{\alpha} \end{bmatrix} +$$

$$+ \begin{bmatrix} k_h & x_Ek_h \\ x_Ek_h & k_\alpha + k_hx_E^2 \end{bmatrix} \begin{bmatrix} h_O \\ \alpha \end{bmatrix} + \begin{bmatrix} 0 & \frac{1}{2}\rho U_\infty^2 2bc_{l_\alpha} \\ 0 & x_A\frac{1}{2}\rho U_\infty^2 2bc_{l_\alpha} \end{bmatrix} \begin{bmatrix} h_O \\ \alpha \end{bmatrix} = \mathbf{0} \tag{3.14}$$

The corresponding non-dimensional version of the above system can be obtained similarly as before by exploiting the scaling rules in eq.3.9 and by dividing the first equation by $mb\omega_\alpha^2$ and the second by $mb^2\omega_\alpha^2$:

$$\begin{bmatrix} 1 & \xi_G \\ \xi_G & r_\alpha^2 - \xi_E^2 + 2\xi_E\xi_G \end{bmatrix} \begin{bmatrix} \ddot{h} \\ \ddot{\alpha} \end{bmatrix} + \hat{U} \begin{bmatrix} \frac{1}{a} & \frac{\xi_P}{a} \\ \frac{\xi_A}{a} & \frac{\xi_A \xi_P}{a} \end{bmatrix} \begin{bmatrix} \dot{h} \\ \dot{\alpha} \end{bmatrix} + \Omega^2 \begin{bmatrix} 1 & \xi_E \\ \xi_E & \frac{r_\alpha^2}{\Omega^2} + \xi_E^2 \end{bmatrix} \begin{bmatrix} h \\ \alpha \end{bmatrix} + \hat{U}^2 \begin{bmatrix} 0 & \frac{1}{a} \\ 0 & \frac{\xi_A}{a} \end{bmatrix} \begin{bmatrix} h \\ \alpha \end{bmatrix} = \mathbf{0} \quad (3.15)$$

Besides, a compact expression of the dimensionless system above can be written also in this case by adding a non-dimensional aerodynamic damping matrix $\boldsymbol{D_a^*}$ (the matrix in the second product of the above equation) to the system in eq.3.11:

$$\boldsymbol{M}^* \ddot{\hat{\boldsymbol{x}}} \ + \ \hat{U} \boldsymbol{D_a^*} \dot{\hat{\boldsymbol{x}}} \ + \ \Omega^2 \boldsymbol{K}^* \hat{\boldsymbol{x}} \ + \ \hat{U}^2 \boldsymbol{K_a^*} \hat{\boldsymbol{x}} \ = \ \mathbf{0} \qquad \text{with} \quad \boldsymbol{D_a^*} = \begin{bmatrix} \frac{1}{a} & \frac{\xi_P}{a} \\ \frac{\xi_A}{a} & \frac{\xi_A \xi_P}{a} \end{bmatrix} \quad (3.16)$$

Finally, the following 4-by-four system of first-order ODEs can be derived for the state-space model of the quasi-steady problem:

$$\begin{cases} \boldsymbol{M}^* \dot{\boldsymbol{y}} + \hat{U} \boldsymbol{D_a^*} \boldsymbol{y} + \Omega^2 \boldsymbol{K}^* \hat{\boldsymbol{x}} + \hat{U}^2 \boldsymbol{K_a^*} \hat{\boldsymbol{x}} = \mathbf{0} \\ \dot{\hat{\boldsymbol{x}}} - \boldsymbol{y} = \mathbf{0} \end{cases} \quad \rightarrow \quad \begin{bmatrix} \boldsymbol{M}^* & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{y}} \\ \dot{\hat{\boldsymbol{x}}} \end{bmatrix} + \begin{bmatrix} \hat{U} \boldsymbol{D_a^*} & \Omega^2 \boldsymbol{K}^* + \hat{U}^2 \boldsymbol{K_a^*} \\ -\boldsymbol{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{y} \\ \hat{\boldsymbol{x}} \end{bmatrix} = \mathbf{0}$$

For the quasi-steady model, the final matrix $\boldsymbol{A_{qs}}$ results from the inversion of the same total mass matrix of the Glauert model, but this time multiplied by a different total stiffness matrix $\boldsymbol{\tilde{K}_{tot}^*}$, due to the presence of the aerodynamic damping operator:

$$\boldsymbol{M_{tot}^*} \dot{\boldsymbol{z}} + \boldsymbol{\tilde{K}_{tot}^*} \boldsymbol{z} = \mathbf{0} \quad \rightarrow \quad \dot{\boldsymbol{z}} = \big(-\boldsymbol{M_{tot}^*}^{-1} \boldsymbol{\tilde{K}_{tot}^*}\big) \boldsymbol{z} = \boldsymbol{A_{qs}}\, \boldsymbol{z} \quad (3.17)$$

General time/frequency-domain Theodorsen

Under a quarter-chord pitching axis location ($\tilde{a} = -1/2$ in eq.2.28), the general equations of the Theodorsen unsteady aerodynamic model for lift and moment shown in eq.3.6 simplify to:

$$L = L_{nc} + L_c = \pi \rho b^2 \left(\ddot{h}_O + \frac{1}{2} b \ddot{\alpha} + U_\infty \dot{\alpha} \right) + 2\pi \rho U_\infty b\, C(k) \left(\dot{h}_O + b\dot{\alpha} + U_\infty \alpha \right)$$

$$M_A = M_A^{(nc)} + \underbrace{M_A^{(c)}}_{=0} = -\pi \rho b^3 \left(\frac{1}{2} \ddot{h}_O + \frac{3}{8} b \ddot{\alpha} + U_\infty \dot{\alpha} \right) \qquad \text{with} \qquad v_n^{(3/4)} = \dot{h}_O + b\dot{\alpha} + U_\infty \alpha \quad (3.18)$$

with the moment only consisting of a non-circulatory component. In the circulatory lift term, the quantity that follows the Theodorsen function is often label with the symbol $v_n^{(3/4)}$ and it represents the aerodynamic input on which the Theodorsen transfer function $C(k)$ applies the delay resulting in the lift-deficiency effect. To employ the Theodorsen model in the EoMs of the structural system in eq.3.6, it is necessary to first employ an approximation for the generalized Theodorsen function $C(k)$ - this is shown in the appendix sec.B.1 - which allows then to obtain pure time-domain expressions for the airload. The latter can be obtained by means of two methods from the same approximate Theodorsen function $\tilde{C}(k)$, with the first one leading to a convolution integral and the second one to additional ODEs for the retained aerodynamic lag-states. The first approach is discussed in the appendix sec.B.2, while the second in the appendix sec.B.4. In this section, only the time domain airloads resulting from the two methods are applied to the structural system in eq.3.6 to get the full aeroelastic formulation of the problem.

Time-domain Theodorsen: convolution integral method

From the approximate Padè-expression of the Theodorsen function $\tilde{C}(k)$ in eq.B.2 of the appendix sec.B.1, it is possible to obtain a pure time domain formulation of the Theodorsen airload by applying the inverse Laplace transform to $\tilde{C(k)}$ and by subsequent time integration including the aerodynamic input. All these steps are described in detail in the appendix sec.B.2 and lead the following expressions for the time-domain approximate lift and aerodynamic moment, when $v_n^{(3/4)}$ is defined as in eq.3.18:

$$\tilde{L} = \pi\rho b^2\Big(\ddot{h}_O + \frac{1}{2}b\ddot{\alpha} + U_\infty\dot{\alpha}\Big) + 2\pi\rho U_\infty b\left[\Big(\dot{h}_O + b\dot{\alpha} + U_\infty\alpha\Big)\Big|_{t=0} W(t) + \int_0^t W(t-\hat{\tau})\Big(\ddot{h}_O + b\ddot{\alpha} + U_\infty\dot{\alpha}\Big)\,d\hat{\tau}\right]$$

$$\tilde{M}_A = -\pi\rho b^3\Big(\frac{1}{2}\ddot{h}_O + \frac{3}{8}b\ddot{\alpha} + U_\infty\dot{\alpha}\Big) \qquad \rightarrow \qquad \boldsymbol{f} = \begin{bmatrix} -\tilde{L} \\ \tilde{M}_A - x_A\tilde{L}\end{bmatrix} \tag{3.19}$$

Based on the above expressions, the derivation of the convolution-integral version of the Theodorsen aeroelastic problem can start. Therefore, by proceeding as in the case of the two previous aerodynamic models, the following dimensional system of equations can be obtained:

$$\begin{bmatrix} m & mx_G \\ mx_G & J_E - mx_E^2 + 2mx_Ex_G \end{bmatrix}\begin{bmatrix}\ddot{h}_O \\ \ddot{\alpha}\end{bmatrix} + \pi\rho b^2\begin{bmatrix} 1 & \frac{b}{2} \\ \frac{b}{2} + x_A & \frac{3}{8}b^2 + x_A\frac{b}{2}\end{bmatrix}\begin{bmatrix}\ddot{h}_O \\ \ddot{\alpha}\end{bmatrix} +$$

$$+\pi\rho b^2 U_\infty\begin{bmatrix} 0 & 1 \\ 0 & b + x_A\end{bmatrix}\begin{bmatrix}\dot{h}_O \\ \dot{\alpha}\end{bmatrix} + \begin{bmatrix} k_h & x_Ek_h \\ x_Ek_h & k_\alpha + k_hx_E^2\end{bmatrix}\begin{bmatrix} h_O \\ \alpha\end{bmatrix} =$$

$$= -2\pi\rho U_\infty b\begin{bmatrix} (\dot{h}_O + b\dot{\alpha} + U_\infty\alpha)\big|_{t=0}W(t) + \int_0^t W(t-\hat{\tau})(\ddot{h}_O + b\ddot{\alpha} + U_\infty\dot{\alpha})\,d\hat{\tau} \\ x_A\big\{(\dot{h}_O + b\dot{\alpha} + U_\infty\alpha)\big|_{t=0}W(t) + \int_0^t W(t-\hat{\tau})(\ddot{h}_O + b\ddot{\alpha} + U_\infty\dot{\alpha})\,d\hat{\tau}\big\}\end{bmatrix} \tag{3.20}$$

Next, to get the non dimensional form of this system, it is firstly required to obtain a non dimensional version of the Padè-approximated Wagner function $\tilde{W}(\tilde{t})$ in terms of the dimensionless time $\hat{t}$ employed in the aeroelastic systems derivation. By recalling the scaling rules of eq.3.9, it is possible to re-state the dimensionless time $\hat{\tau}$[3] in terms of $\hat{U}$ and $\hat{t}$, namely $t = \hat{U}\hat{t}$. Therefore $\tilde{W}(\hat{t})$ becomes:

$$\tilde{W}(\hat{t}) = \frac{1}{2}\Big(1.9956 - 0.6114e^{-0.0965\hat{U}\hat{t}} - 0.3842e^{-0.4555\hat{U}\hat{t}}\Big) \tag{3.21}$$

Finally, by following the same procedure employed for the two previous models the non-dimansional version of the aeroelastic problem with airloads according to the Padè-approximated Theodorsen model can be obtained:

$$\begin{bmatrix} 1 & \xi_G \\ \xi_G & r_\alpha^2 - \xi_E^2 + 2\xi_E\xi_G\end{bmatrix}\begin{bmatrix}\ddot{h} \\ \ddot{\alpha}\end{bmatrix} + \begin{bmatrix}\frac{1}{2a} & \frac{1}{4a} \\ \frac{1}{4a} + \frac{\xi_A}{2a} & \frac{3}{16a} + \frac{\xi_A}{4a}\end{bmatrix}\begin{bmatrix}\ddot{h} \\ \ddot{\alpha}\end{bmatrix} + \hat{U}\begin{bmatrix} 0 & \frac{1}{2a} \\ 0 & \frac{1}{2a} + \frac{\xi_A}{2a}\end{bmatrix}\begin{bmatrix}\dot{h} \\ \dot{\alpha}\end{bmatrix}$$

$$+\,\Omega^2\begin{bmatrix} 1 & \xi_E \\ \xi_E & \frac{r_\alpha^2}{\Omega^2} + \xi_E^2\end{bmatrix}\begin{bmatrix} h \\ \alpha\end{bmatrix} = -\frac{\hat{U}}{a}\begin{bmatrix}(\dot{h} + \dot{\alpha} + \hat{U}\alpha)\big|_{t=0}\hat{W}(\hat{t}) + \int_0^{\hat{t}}\hat{W}(\hat{t}-\hat{\tau})(\ddot{h} + \ddot{\alpha} + \hat{U}\dot{\alpha})\,d\hat{\tau} \\ \xi_A\big\{(\dot{h} + \dot{\alpha} + \hat{U}\alpha)\big|_{t=0}\hat{W}(\hat{t}) + \int_0^{\hat{t}}\hat{W}(\hat{t}-\hat{\tau})(\ddot{h} + \ddot{\alpha} + \hat{U}\dot{\alpha})\,d\hat{\tau}\big\}\end{bmatrix} \tag{3.22}$$

The above system can be then expressed in a compact matrix notation similarly to the two previous models by adding a dimensionless aerodynamic mass matrix $\boldsymbol{M_a^*}$, introducing a different aerodynamic damping matrix $\boldsymbol{\tilde{D}_a^*}$ and by replacing the $\boldsymbol{K_a^*}$ with the dimensionless force vector $\boldsymbol{f}^*(\hat{t})$. This leads to:

$$\boldsymbol{M}^*\ddot{\hat{\boldsymbol{x}}} + \boldsymbol{M_a^*}\,\ddot{\hat{\boldsymbol{x}}} + \hat{U}\boldsymbol{\tilde{D}_a^*}\,\dot{\hat{\boldsymbol{x}}} + \Omega^2\boldsymbol{K}^*\hat{\boldsymbol{x}} = \boldsymbol{f}^*(\hat{t}) \tag{3.23}$$

Regarding the state-formulation of this system in terms of four first-order ODEs, this can be derived similarly as before, obtaining:

$$\begin{cases}\boldsymbol{M}^*\dot{\boldsymbol{y}} + \boldsymbol{M_a^*}\,\dot{\boldsymbol{y}} + \hat{U}\boldsymbol{\tilde{D}_a^*}\,\boldsymbol{y} + \Omega^2\boldsymbol{K}^*\hat{\boldsymbol{x}} = \boldsymbol{f}^*(\hat{t}) \\ \dot{\hat{\boldsymbol{x}}} - \boldsymbol{y} = \boldsymbol{0}\end{cases} \qquad \rightarrow$$

[3] This time variable is related to the reduced frequency (k) and has to do with the transformation of the Theodorsen model from the frequency to the time domain carried out in the appendix sec.B.2.

$$\rightarrow \qquad \begin{bmatrix} \boldsymbol{M}^* + \boldsymbol{M}_{\boldsymbol{a}}^* & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{y}} \\ \dot{\hat{\boldsymbol{x}}} \end{bmatrix} + \begin{bmatrix} \hat{U}\boldsymbol{D}_{\boldsymbol{a}}^* & \Omega^2 \boldsymbol{K}^* \\ -\boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{y} \\ \hat{\boldsymbol{x}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}^*(\hat{t}\,) \\ \boldsymbol{0} \end{bmatrix}$$

In the end, the following compact notation where matrix $\boldsymbol{A_{tci}}$ is frozen in time, while vector $\boldsymbol{b}_{tci}(\hat{t}\,)$ changes for each timestep, as it contains the force vector related to the convolution integral:

$$\tilde{\boldsymbol{M}}_{\boldsymbol{tot}}^* \dot{\boldsymbol{z}} + \hat{\boldsymbol{K}}_{\boldsymbol{tot}}^* \boldsymbol{z} = \boldsymbol{F}^*(\hat{t}\,) \quad \rightarrow \quad \dot{\boldsymbol{z}} = \big(- \tilde{\boldsymbol{M}}_{\boldsymbol{tot}}^{*\;-1} \hat{\boldsymbol{K}}_{\boldsymbol{tot}}^* \big) \boldsymbol{z} + \tilde{\boldsymbol{M}}_{\boldsymbol{tot}}^{*\;-1} \boldsymbol{F}^*(\hat{t}\,) = \boldsymbol{A_{tci}}\, \boldsymbol{z} + \boldsymbol{b}_{tci}(\hat{t}\,) \qquad (3.24)$$

Time-domain Theodorsen: aerodynamic lag-states method

Shortly, to derive the lag-states formulation of the time-domain approximated Theodorsen airload, the inverse of the Laplace transformation is applied to the whole lift expression. This leads to two second-order ODEs for the two aerodynamic lag-states approximating the memory effect of the wake and with the structural $v_n^{*(3/4)}$ term on the lhs. The derivation is in discussed in detail in the appendix sec.B.4 and leads to the following dimensionless lift and moment expressions:

$$L^* = \frac{1}{2a}\Big(\ddot{h} + \frac{1}{2}\ddot{\alpha} + \hat{U}\dot{\alpha}\Big) + \frac{\hat{U}}{a}\Big(\ddot{r} + \hat{U}(n_1 + n_2)\dot{r} + \hat{U}^2 n_1 n_2 r\Big); \qquad M_A^* = \frac{1}{2a}\left(\frac{\ddot{h}}{2} + \frac{3}{8}\ddot{\alpha} + \hat{U}\dot{\alpha}\right)$$

When the above equations are applied to the structural system in eq.3.6, the below aeroelastic system is obtained:

$$\begin{cases} \boldsymbol{M}^* \ddot{\boldsymbol{x}} + \boldsymbol{K}^* \boldsymbol{x} = \begin{bmatrix} -L^* \\ M_A^* - \xi_A L^* \end{bmatrix} \\ \ddot{r} + \hat{U}(d_1 + d_2)\dot{r} + \hat{U}^2 d_1 d_2 r = v_n^{*(3/4)} \end{cases} \qquad \text{with} \qquad v_n^{*(3/4)} = \dot{h} + \dot{\alpha} + U_\infty \alpha \qquad (3.25)$$

which consists in three second-order ODEs in $\hat{t}$. Besides, the latter system can be written-down in terms of the following matrices and solution vector:

$$\begin{bmatrix} 1 & \xi_G & 0 \\ \xi_G & r_\alpha^2 - \xi_E^2 + 2\xi_E\xi_G & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \ddot{h} \\ \ddot{\alpha} \\ \ddot{r} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \hat{U}(d_1 + d_2) \end{bmatrix} \begin{bmatrix} \dot{h} \\ \dot{\alpha} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} \Omega^2 & \Omega^2 \xi_E & 0 \\ \Omega^2 \xi_E & r_\alpha^2 + \Omega^2 \xi_E^2 & 0 \\ 0 & 0 & \hat{U} d_1 d_2 \end{bmatrix} \begin{bmatrix} h \\ \alpha \\ r \end{bmatrix} =$$

$$= - \begin{bmatrix} \frac{1}{2a} & \frac{1}{4a} & \frac{\hat{U}}{2a} \\ \frac{1}{4a} + \frac{\xi_A}{2a} & \frac{3}{16a} + \frac{\xi_A}{4a} & \frac{\xi_A \hat{U}}{2a} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \ddot{h} \\ \ddot{\alpha} \\ \ddot{r} \end{bmatrix} - \begin{bmatrix} 0 & \frac{\hat{U}}{2a} & \frac{\hat{U}^2}{2a}(n_1 + n_2) \\ 0 & \frac{\hat{U}}{2a} + \frac{\hat{U}\xi_A}{2a} & \frac{\hat{U}^2}{2a}\xi_A(n_1 + n_2) \\ -1 & -1 & 0 \end{bmatrix} \begin{bmatrix} \dot{h} \\ \dot{\alpha} \\ \dot{r} \end{bmatrix} - \begin{bmatrix} 0 & 0 & \frac{\hat{U}^3}{2a} n_1 n_2 \\ 0 & 0 & \frac{\hat{U}^3}{2a}\xi_A n_1 n_2 \\ 0 & -\hat{U} & 0 \end{bmatrix} \begin{bmatrix} h \\ \alpha \\ r \end{bmatrix}$$

The above system can be made more compact leading to the equation shown below:

$$\boldsymbol{M_T^*} \ddot{\boldsymbol{w}} \; + \; \boldsymbol{R_T^*} \dot{\boldsymbol{w}} \; + \; \boldsymbol{K_T^*} \boldsymbol{w} \; = \; - \boldsymbol{E_1^*} \ddot{\boldsymbol{w}} \; - \; \boldsymbol{E_2^*} \dot{\boldsymbol{w}} \; - \; \boldsymbol{E_3^*} \boldsymbol{w}$$

Next, the state-space model of the above system leads to six first-order ODEs for the four structure states and the two aerodynamic lag-states corresponding to the two-pole two-zero Padè approximation of the Theodorsen function:

$$\begin{cases} (\boldsymbol{M_T^*} + \boldsymbol{E_1^*})\dot{\boldsymbol{y}} + (\boldsymbol{R_T^*} + \boldsymbol{E_2^*})\boldsymbol{y} + (\boldsymbol{K_T^*} + \boldsymbol{E_3^*})\boldsymbol{w} = \boldsymbol{0} \\ \dot{\boldsymbol{w}} - \boldsymbol{y} = \boldsymbol{0} \end{cases} \qquad \rightarrow$$

$$\rightarrow \qquad \begin{bmatrix} \boldsymbol{M_T^*}{+}\boldsymbol{E_1^*} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{y}} \\ \dot{\boldsymbol{w}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{R_T^*}{+}\boldsymbol{E_2^*} & \boldsymbol{K_T^*}{+}\boldsymbol{E_3^*} \\ -\boldsymbol{I} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{w} \end{bmatrix} = \boldsymbol{0}$$

Finally, the final system is homogeneous with a single six-by-six matrix $\boldsymbol{A_{tls}}$:

$$\hat{\boldsymbol{M}}_{\boldsymbol{tot}}^* \dot{\boldsymbol{z}} + \bar{\boldsymbol{K}}_{\boldsymbol{tot}}^* \boldsymbol{z} = \boldsymbol{0} \quad \rightarrow \quad \dot{\boldsymbol{z}} = \big(- \hat{\boldsymbol{M}}_{\boldsymbol{tot}}^{*\;-1} \bar{\boldsymbol{K}}_{\boldsymbol{tot}}^* \big) \boldsymbol{z} = \boldsymbol{A_{tls}}\, \boldsymbol{z} \qquad (3.26)$$

3.1.4. Analytical solution

The dimensionless aeroelastic systems with the steady and quasi-steady airload can be directly solved analytically, while regarding the Theodorsen system, this is possible for the case of the aerodynamic lag-states method. This can be done because, in these three situations, the final systems are homogeneous and, therefore, it is possible to exploit the eigenvalues and eigenvectors of the final matrices $\boldsymbol{A_s}$, $\boldsymbol{A_{qs}}$ and $\boldsymbol{A_{tls}}$ to compute analytical solutions based on a linear combination of exponential functions. This is carried out by considering the set of values for the dimensionless parameters in tab.3.1. The first step consists in the calculation of the eigenvectors $\boldsymbol{\tilde{z}_i}$ and eigenvalues λ_i of the generic final matrix $\boldsymbol{A}$ of each system. The number of such eigenvalues/vectors is equal to four for the Glauert and quasi-steady cases while it is six for the Theodorsen problem due to the two additional aerodynamic lag-states. The relation between the eigenvalues and eigenvectors of matrix $\boldsymbol{A}$ and the solution of the system $\boldsymbol{z}(t)$ is the following:

$\xi_A = 0.0$	$\xi_E = 0.1$	$\xi_G = 0.5$	$a = 1.5$	$\Omega = 0.8$	$r_\alpha = 0.5$

Table 3.1: Values for the dimensionless parameters characterizing the structural side of the problem.

$$\boldsymbol{z}(t) = \sum_{i=1}^{4} c_i \boldsymbol{\tilde{z}_i} e^{\lambda_i t} \quad \text{and} \quad \boldsymbol{c} = \boldsymbol{V}^T \boldsymbol{z}(0) \tag{3.27}$$

where $\boldsymbol{c}$ is a vector taking into account the prescribed initial conditions of the problem $\boldsymbol{z}(0)$ by means of matrix $\boldsymbol{V}^T$, which is the left-eigenvector matrix of $\boldsymbol{A}$. Such expression for $\boldsymbol{z}(t)$, if plugged into the final homogeneous systems, leads to:

$$\sum \lambda_i c_i \boldsymbol{\tilde{z}_i} e^{\lambda_i t} = \sum c_i \boldsymbol{A} \boldsymbol{\tilde{z}_i} e^{\lambda_i t} \quad \rightarrow \quad (\lambda_i \boldsymbol{I} - \boldsymbol{A})\boldsymbol{\tilde{z}_i} = 0 \quad \text{with i=1,2,3,4}$$

where the linear independence of the $e^{\lambda_i t}$ objects is exploited to simplify the left part of the above equation. Regarding the right part, this can provide relevant solutions (all solutions different from $\boldsymbol{\tilde{z}_i} = 0$) if it is possible to solve the problem $det(\lambda_i \boldsymbol{I} - \boldsymbol{A}) = 0$. This means that, to find the non-zero eigenvectors $\boldsymbol{\tilde{z}_i} = 0$, it is necessary to find the roots of the characteristic polynomial of $\boldsymbol{A}$, which are, at the same time, also the eigenvalues (λ_i) of $\boldsymbol{A}$. Finally, due to the mathematical properties of the final matrix $\boldsymbol{A}$, which is non-symmetric in all of the three problems, the eigenvalues λ_i are two couples of complex conjugate numbers with values changing according to the non-dimensional speed parameter $\hat{U}$.

Eigenvalue dependence on flow velocity

The dependence of the eigenvectors $\boldsymbol{\tilde{z}_i}$ and eigenvalues λ_i of $\boldsymbol{A}$ to $\hat{U}$ is a very important property of every aeroelastic system, as the sign of the real part of λ_i - $Re(\lambda_i)$ - is used to described the dynamic stability properties of the system, namely if this is in a condition below or above flutter. When the sign of $Re(\lambda_i)$ becomes positive, the flutter boundary is overcome and the system is no more dynamically stable. This means that a further increase in speed will determine a more and more divergent behaviour of the solution with large values for the position states reached. If in reality this is associated to damage and mechanical failure of the system, in the case of the problem considered, such behaviour is still to be avoided, as it violates the small displacement assumption employed to linearize the kinematics of the system. For this reason, before proceeding with the calculation of the analytical solution of the system for the three aeroelastic problems, it is necessary to find the flutter boundary of each one of them. Only after this step, a condition for $\hat{U}$ can be set below the flutter boundary to have small undamped or damped oscillations according to the aerodynamic model employed.

Flutter boundary determination

The determination of the flutter boundary for the steady, quasi-steady and Theodorsen aeroelastic problems is determined by repeatedly calculating $Re(\lambda_i)$ for increasing $\hat{U}$ (step 0.001) until at least one of the eigenvalues returns a positive real part. The result of this investigation is shown in tab.3.2, where in the case of the quasi-steady model the location of the pitching axis x_P corresponds to the quarter-chord point x_A, as the latter is chosen as reference point for the evaluation of the kinematics. Based on the results from the flutter boundary analysis, flow conditions close to such threshold are considered, namely $\hat{U} = 0.9\hat{U}_f$ for each of the three aeroelastic problems, for the calculation of the analytical solution with the method of the eigenvalues.

Model	$\hat{U}_f$
Steady Glauert	0.761
Quasi-Steady	0.686 for $\xi_P = \xi_A$
Theodorsen (aerodynamic lag-states)	1.194

Table 3.2: Results of the flutter boundary (dimensionless inflow velocity) investigation for the three aeroelastic problems.

3.2. Numerical modelling of the problem

The four aeroelastic problems can be synthetically described in terms of the corresponding linear systems of first-order ODEs shown in tab.3.3. Such systems contain four equations for the steady, quasi-steady and Theodorsen models with convolution integral method describing the time evolution of the structural kinematic states, while in the case of the Theodorsen model with aerodynamic lag-states method there are six equations due to the two aerodynamic lag-states introduced. The numerical solution of the above systems can be obtained by applying a specific time marching technique able to compute the value of the solution vector $\boldsymbol{z}_{n+1}$ at timestep $\hat{t}_{n+1}$ based on the knowledge of solution at the previous timesteps.

Model	System of ordinary differential equations
Steady Glauert	$\dot{\boldsymbol{z}} = \big(-\boldsymbol{M}^{*}_{\boldsymbol{tot}}{}^{-1}\boldsymbol{K}^{*}_{\boldsymbol{tot}}\big)\boldsymbol{z} = \boldsymbol{A_s}\,\boldsymbol{z}$
Quasi-Steady	$\dot{\boldsymbol{z}} = \big(-\boldsymbol{M}^{*}_{\boldsymbol{tot}}{}^{-1}\tilde{\boldsymbol{K}}^{*}_{\boldsymbol{tot}}\big)\boldsymbol{z} = \boldsymbol{A_{qs}}\,\boldsymbol{z}$
Theodorsen aerodynamic lag-states	$\dot{\boldsymbol{z}} = \big(-\hat{\boldsymbol{M}}^{*}_{\boldsymbol{tot}}{}^{-1}\bar{\boldsymbol{K}}^{*}_{\boldsymbol{tot}}\big)\boldsymbol{z} = \boldsymbol{A_{tls}}\,\boldsymbol{z}$
Theodorsen convolution integral	$\dot{\boldsymbol{z}} = \big(-\tilde{\boldsymbol{M}}^{*}_{\boldsymbol{tot}}{}^{-1}\hat{\boldsymbol{K}}^{*}_{\boldsymbol{tot}}\big)\boldsymbol{z} + \tilde{\boldsymbol{M}}^{*}_{\boldsymbol{tot}}{}^{-1}\boldsymbol{F}^{*}(\hat{t}) = \boldsymbol{A_{tci}}\,\boldsymbol{z} + \boldsymbol{b}_{tci}(\hat{t})$

Table 3.3: Systems of ODEs for the four aeroelastic systems.

3.2.1. Time marching methods

For the steady, quasi-steady and Theodorsen with lag-states methods, due to the simplicity of the final system - which is homogeneous and does not contain the convolution integral - the Crank-Nicolson method is applied. In the case of the Theodorsen system with convolution integral, the simpler forward Euler integration scheme is implemented. In the subsequent paragraphs, both methods are applied to a uniformly-spaced time array to discretize the analytical aeroelastic problems in tab.3.3 and then a few aspects of their numerical properties are presented.

Crank-Nicolson method

In the case of a generic homogeneous system of first-order ODEs $\dot{\boldsymbol{z}} = \boldsymbol{A}\boldsymbol{z}$ the Crank Nicolson method consists in approximating the time derivative of the solution vector $\dot{\boldsymbol{z}}$ by means of a forward finite difference in time, given a time difference $\Delta\hat{t} = \hat{t}_{n+1} - \hat{t}_n$ between consecutive timesteps, and by replacing the solution vector $\dot{\boldsymbol{z}}$ with an average between its value at timestep $\hat{t}_n$ and $\hat{t}_{n+1}$. This leads to:

$$\dot{\boldsymbol{z}} = \boldsymbol{A}\boldsymbol{z} \quad \approx \quad \frac{\boldsymbol{z}_{n+1} - \boldsymbol{z}_n}{\Delta\hat{t}} = \boldsymbol{A}\frac{\boldsymbol{z}_{n+1} + \boldsymbol{z}_n}{2} \quad \rightarrow \quad \boldsymbol{z}_{n+1} = \left(\frac{\boldsymbol{I}}{\Delta\hat{t}} - \frac{\boldsymbol{A}}{2}\right)^{-1}\left(\frac{\boldsymbol{I}}{\Delta\hat{t}} + \frac{\boldsymbol{A}}{2}\right)\boldsymbol{z}_n \tag{3.28}$$

Therefore, the above equation can be applied to the generic final $\boldsymbol{A}$ matrix for the steady, quasi-steady and Theodorsen with lag-states. This integration scheme is implicit, as the right-hand-side of the system is approximated with values at the timestep $\hat{t}_{n+1}$ and single step, since values no older than n are employed backwards. This single-step property and the evaluation method employed to discretize the rhs term, is a property that the Crank-Nicolson method shares with the family of Runge-Kutta time-integration methods, where even more sophisticated evaluation methods are available to improve the accuracy of the solution. Regarding the accuracy of the Crank-Nicolson method, it is a consistent second order method, as the global discretization error decreases quadratically when $\Delta\hat{t}$ is reduced. Eventually, this method is unconditionally stable on a potentially unbounded time interval, which means that even if $\Delta\hat{t}$ is large, the method will still be stable.

Forward Euler method

The forward Euler method is the most simple time marching scheme that can be used to solve ODEs. When applied to the final system of the Theodorsen problem with convolution integral, the lhs time derivative of the solution vector is discretized with a forward finite difference similarly to the Crank-Nicolson method, while the rhs is evaluated at the previous n timestep. This leads to:

$$\dot{\boldsymbol{z}} = \boldsymbol{A_{tci}}\,\boldsymbol{z} + \boldsymbol{b}_{tci}(\hat{t}\,) \quad \approx \quad \frac{\boldsymbol{z}_{n+1} - \boldsymbol{z}_n}{\Delta\hat{t}} = \boldsymbol{A_{tci}}\,\boldsymbol{z}_n + \boldsymbol{b}_{tci}(\hat{t}_n)$$

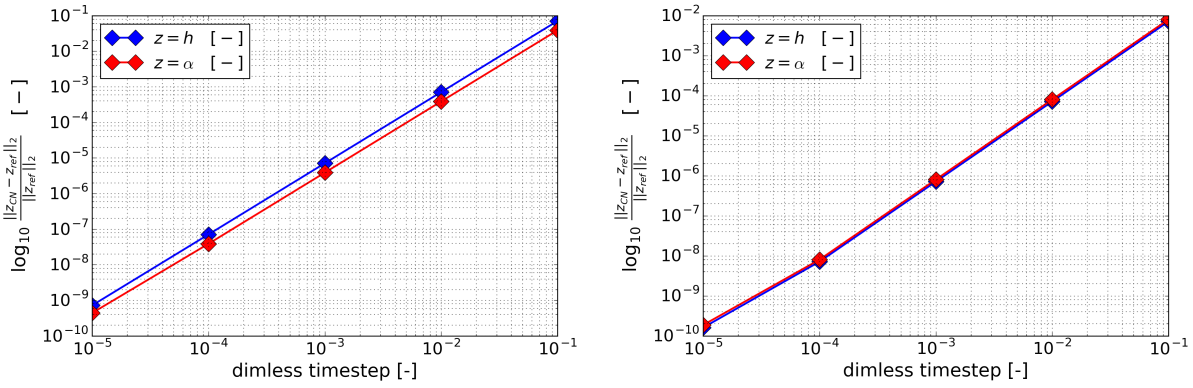
and to the following discretized system:

$$\boldsymbol{z}_{n+1} = (\boldsymbol{I} + \boldsymbol{A_{tci}}\Delta\hat{t})\boldsymbol{z}_n + \boldsymbol{b}_{tci}(\hat{t}_n\,)\Delta\hat{t} \quad \text{with} \quad \boldsymbol{b}_{tci}(\hat{t}_n) = \boldsymbol{\tilde{M}^*_{tot}}^{-1}\boldsymbol{F}^*(\hat{t}_n) \tag{3.29}$$

where the discretization of the $\boldsymbol{F}^*(\hat{t}_n)$ term containing the convolution integral is carried out in detail in the appendix sec.B.3. The forward Euler method, although simple and straightfoward in the implementation, can lead to numerical instability, if the timestep value $\Delta\hat{t}$ is too large. Besides, even when $\Delta\hat{t}$ is sufficiently small to return a stable behaviour of the numerical solution, the accuracy is lower than the Crank Nicolson method, as the discretization error decreases only linearly with the timestep. Regardless of the forward Euler method, the numerical solution of the Theodorsen system with convolution integral is expected to require a higher computational cost with respect to the Theodorsen method with lag-states, due to the presence of the convolution integral. As shown in the appendix sec.B.3, the computation of the convolution integral requires an internal loop per each time-integration timestep, this is because of the computation of wake memory-effect requires all the previous values of the solution up to the initial conditions.

3.2.2. Consistency analysis

In this section, the numerical solutions (consisting in the position and velocity kinematic states) for the four aeroelastic problems is computed for various timesteps and compared with the corresponding analytical ones calculated previously. Given the availability of the analytical solution show in eq.3.27, the behaviour of the four systems is known, therefore, the main purpose of this step is to verify the consistency of the numerical method employed and to find an optimal value of the timestep in terms of both accuracy and cost of the simulation. Both the numerical and the analytical solutions are computed for the values of the dimensionless problem parameters shown in tab.3.1, with initial conditions consisting in only a perturbation of plunge velocity equal to the 1% of the inflow velocity, which is set to the 90% of the flutter value prediction shown in tab.3.2 for each one of three aerodynamic models.

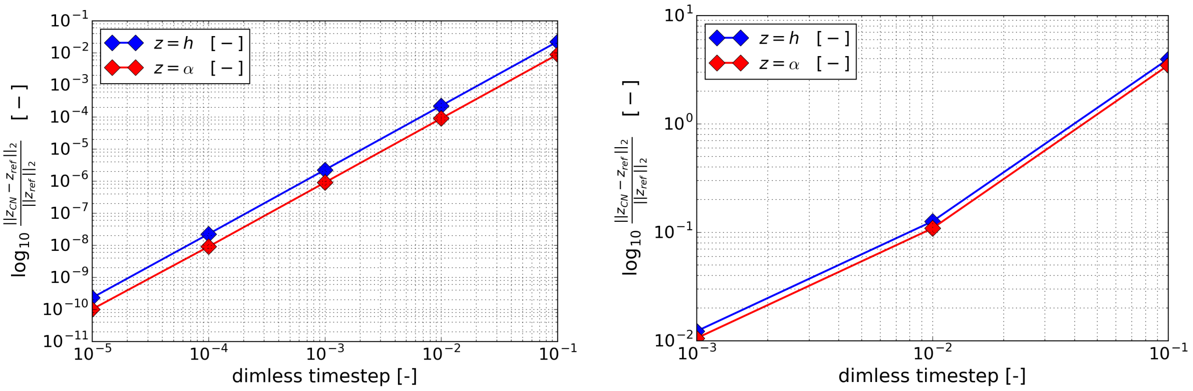


(a) Steady Glauert

(b) Quasi-steady

Figure 3.3: Discretization error versus computational timestep related to the steady (left) and quasi-steady (right) aeroelastic systems when both are compared with the analytical solution obtained with the method of eigenvalues.

The discretization error - defined as the L2 norm of the difference between the numerical and analytical solution for each position and velocity state divided by the same for only the analytical solution - for the steady problem is shown in fig.3.3-(a), while the same for the quasi-steady system in fig.3.3-(b). Both plots confirm the second-order accuracy of the Crank-Nicolson method as error decreases by eight orders of magnitude when the timestep is decreased by four orders of magnitude. Regarding the cost of the simulations, the steady ones require around one-half of the calculation time than the quasi-steady ones, most likely due to the simpler mathematical systems. In both methods, the cost increases by one order of magnitude when the timestep is reduced by a factor of ten going from $\approx 10^{-3}s$ for $\Delta\hat{t} = 0.1$ to ≈ 10 when $\Delta\hat{t} = 10^{-5}$ on a common laptop.



(a) Aerodyanmic lag-states

(b) Convolution integral

Figure 3.4: Discretization error versus computational timestep related to the lag-states (left) and convolution integral (right) approximation methods of the Theodorsen model when both are compared with the analytical solution obtained with the method of eigenvalues.

In figure 3.4 the discretization error previously defined is shown for the two methods employed to solve the Theodorsen aeroelastic system. Fig.3.4-(a) is related to the kinematic states calculated by the aerodyanmic lag-states method and in this case the solution is obtained by employing the Crank-Nicolson integration scheme. Such a plot shows a behaviour very similar to the two previous cases in terms of a quadratic convergence trend with a decreasing timestep. Besides, also the calculation times are of the same order of the steady and quasi-steady problems, with an increase of around 50% per timestep calculation point. In the case of the convolution integral method, due to the increased cost related to the convolution integral itself, only timesteps 0.1, 0.01 and 0.001 are considered. Indeed, the simulation time is around four orders of magnitude greater than the one required by the other methods, and, in the case of the smallest timestep, it took more than one hour to compute the solution vector. Besides, due to the simpler integration scheme employed, also the global error shows a worse convergence trend, which is close to the expected linear trend. Consequently, due to such reasons of significantly higher simulation cost and lower accuracy, the convolution integral method is not considered as airload model for the Simpack® simulation of the Theodorsen aeroelastic problem. In addition, both time-domain approximation methods of the Theodorsen model tend to converge to the same solution for a decreasing timestep.

Given the previously described results for the steady, quasi-steady and Theodorsen with aerodynamic lag-states problems, the numerical solutions obtained with a dimensionless timestep of $\Delta\hat{t} = 10^{-4}$ is selected for the subsequent verification activity of the MBD simulation setup.

3.3. Simpack® simulation setup

In this section, the setup of the Simpack® simulations related to the three aeroelastic systems previously derived (steady, quasi-steady and Theodorsen with lag-states method) is described by following the general workflow outlined in sec.2.3.2. In doing so, aspects related to global simulation parameters, body, motion, elastic forces and time integration are common for the three simulations, while the modelling of the aerodynamic forces significantly differs.

Global parameters and Subvars

Before discussing the proper modelling steps and options employed, it is necessary to modify some default setups among the *Global Parameters*, such as the switching-off of the gravity force by setting all its components to zero. Besides, to speed-up and parameterize the simulation setup, *subvars* are defined for all the required simulation parameters. These quantities can be obtained by rewriting the scaling rules in eq.3.9 to get expressions for the dimensional simulation parameters (airfoil geometry, aerodynamic and structural quantities) in terms of the dimensionless ones used for the derivation of the aeroelastic systems in sec.3.1.3. Next, values for the dimensional parameters can be obtained from the values of the dimensionless ones shown tab.3.1. Both aspects are covered in the appendix sec.C, where also additional parameters (fluid-flow density, airfoil chord and depth, as well as pitching angular velocity) are prescribed to determine the values for the dimensional Simpack® simulation parameters finally shown in tab.3.4.

Name	Symbol	Formula	Value	Unit	*Subvar*
airfoil semi-chord	b	-	1.0	[m]	$_b
airfoil depth	d	-	1.0	[m]	$_depth
angular pitch frequency	ω_α	-	1.0	[rad/s]	$_oma
point E x-coordinate	x_E	$\xi_E b$	0.1	[m]	$_xe
point G x-coordinate	x_G	$\xi_G b$	0.5	[m]	$_xg
flow density	ρ	-	1.225	[kg/m^3]	$_rho_air
lift aerodynamic derivative	c_{l_α}	-	2π	[-]	$_cla
flow velocity [S; QS]	U	$0.9\hat{U}_f b\omega_\alpha$	0.6849; 0.6174	[m/s]	$_uinf
airfoil mass	m	$1.5\rho b^2 c_{l_\alpha} d$	11.5453	[kg]	$_m
translation stiffness	k_h	$0.8^2 m\omega_\alpha^2$	7.389	[N/m]	$_kh
rotation stiffness	k_α	$0.5^2 mb^2\omega_\alpha^2$	2.8863	[Nm/rad]	$_ka
moment of inertia	J_E	$0.5^2 mb^2$	2.8863	[kgm^2]	$_JE

Table 3.4: Dimensional parameters employed for the Simpack® simulations by means of *subvars*. The letters S and QS are used to define the two different inflow velocity values employed to model the Steady and Quasi-Steady airload.

Body definition: geometric primitive

The definition of the single simulation body related to the flat-plate airfoil starts by prescribing a shape, which corresponds to the *Primitive* shown in fig.3.5. Therefore, in the model tree of the Simpack® Pre-Processor GUI, a *Cuboid Primitive type* is selected. The default *BRF Marker* settings are kept without prescribing any position or orientation angle obtaining a cuboid centered at the *BRF* location. Regarding the cuboid physical dimensions, these are defined based on two times the *subvar* "$_b" along the chordwise direction and "$_depth" for the spanwise one (with both *subvars* indicated in tab.3.4) in addition to a zero thickness along the z-direction, as the mass properties of the airfoil are known from the analytical model[4]. Furthermore, to maintain the correspondence between the 2D analytical model and the Simpack® intrinsic 3D simulation, the value for the body depth/span chosen is 1m.

Body definition: Markers

For the subsequent definition of the *Joint* and *Force elements*, *Markers* (namely local coordinate systems) of the *Cardan Angles type* are defined on top of the body surface. As the reference point for the evaluation of the body motion is the AC or point A, such point is defined for first at a distance equal to half the airfoil semi-chord from the BRF towards the flat-plate LE. In this fashion, the location of *Marker* A perfectly matches the quarter chord at the middle of the airfoil span. Once the A *Marker* is defined, the *Markers* E and G are obtained by employing the *subvars* "$_xe" and "$_xg" shown in tab.3.4 along the chordwise direction of the MBD computational domain. According to what mentioned before for the *Primitive* setup with respect to the *BRF*, the position of point G is exactly the same as the *BRF* one.

Body definition: Body Properties

After prescribing a shape and local frames, also mass and inertial properties need to be provided to completely define the simulation body. Therefore, in the *Body Properties* window, the user has to specify the *Body type* between *rigid* and the multiple options available for flexible structures. In the problem of interest, the option *rigid* is selected and the data input option is set to *Manual*, as mass

[4] Alternatively, the solver can automatically compute the mass properties of the body based on the shape of its primitives, but this option requires that all the primitive dimensions have non-zero values.

("$_m") and inertial properties ("$_JE") of the body are known. In the same window, the position of the CoG (*Marker* G) and the body mass, as shown in tab.3.4, are first set. Next, the inertia tensor is prescribed relative to *Marker* E and filled with the J_E simulation parameter only for the J_{xx} diagonal moment, while the other moments are left to the default unitary value (no rotation is set about the y- and z-axes) and the off-diagonal products are left equal to zero due to the plain rotation considered.

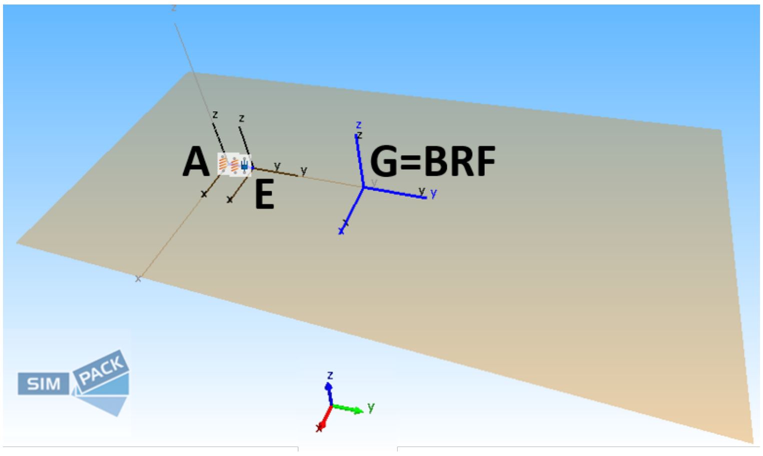


Figure 3.5: Body *Primitive*, reference frames and *Markers* for the Simpack® MBD simulation.

Joint definition

To model the body motion, the combination *Joints/Constraints* is employed instead of only *Connections*, but the two options are conceptually identical and expected to return very similar results. To have the same airfoil roto-translation of the analytical model, a *User Defined Joint type* is chosen with point A as the *To Marker* of the kinematic chain, without introducing any *Constraint*. In the *States* section of the *Joint* window, the initial conditions are defined consistently with the numerical/analytical reference solutions (initial plunge velocity perturbation) and the other options are left unchanged with their default setups. In the *Parameters* section, the DoFs that need to be activated to get the desired motion - namely rotation about the x-axis and translation along the z-axis - are prescribed and then the *Trans-rot sequence*[5] is set to *rotation first*, as it provided the best phase match with the reference solution. Finally, it is relevant to point out that the position of *M_Isys* is automatically shifted to the *To Marker* location (point A) of the *Joint*. In this way, point A becomes also the origin of the inertial reference system, and the kinematic states, as well as the expressions for the aerodynamic force, are computed with respect to this point, consistently with the analytical model.

Springs Force Element

Similarly to *Joints*, also the modelling of the translation and rotational springs is done by considering the very general *Bushing Cmp Force Element type* to have the freedom to define arbitrary spring-damper forces/moments in all the possible six spatial directions. Next, to exactly replicate the system drawing in fig.3.2, where the spring force $k_h h_O$ has an opposite direction with respect to the plunge DoF and the spring torque $k_\alpha \alpha$ is counter-rotating with respect to the pitch motion, the *To Marker* is set to Marker E. Another important parameter to set is the *Mode of angle calculation*, which needs to be set to the option *All angles* $<$ *10 [deg]*[6] to be consistent with the linear formulation of the reference solution. Finally, by providing the *subvar* for the translational "$_kh" and rotational "$_ka" stiffnesses respectively along the z-direction and for the α rotation, all the required steps to define the lumped elasticity of the body are completed.

[5] This *Joint* parameter deals with the order between single rotations and translations employed by the solver to assemble the transformation matrix, which transforms a vector from one coordinate system to the other [65].

[6] This option deals with the full non linear expressions of the angles between the *From Marker* and the *To Marker* of the modelling element due to trigonometric functions. With the option *All angles* $<$ *10 [deg]*, the expressions are linearized for all the angles at stake. This increases the computation speed and provides still accurate results, as long as angles are significantly below 10° [65].

Time Integration Solver

To be entirely consistent, the setup of *Time Integration Solver* in Simpack® has to be as close as possible to the time integration methods employed to get the numerical solutions. Besides, a physical timestep of $\Delta t = 0.0001s$ and a simulation duration of 50s are employed, which correspond to a simulation duration of 50 units of dimensionless time and a dimensionless timestep $\Delta\hat{t} = 0.0001$ as well, do to the dimensionalization approach outlined in the appendix sec.C. Regarding the specific integration scheme, as the steady, quasi-steady and Theodorsen with lag-states numerical reference solutions are obtained with the Crank-Nicolson method, fixed timestep integrators need to be employed in the MBD simulation. Among those available, the *Linear Implicit "one-leg" method* with a default value for the *Node c* parameter equal to 0.5 is selected. This choice is motivated by the fact that this Simpack® integrator returns the best match and because its formulation is the closest one to the Crank-Nicolson method, given the common properties of being single-step, implicit and linear methods (for linear problems, like the one at stake). Finally, no further advanced options are changed from the default ones.

Airload Force Element for the steady and quasi-steady problems

Due to the different level of complexity in the analytical expressions for the aerodynamic forces, different methods between the steady and quasi-stead problems on the one hand and the Theodorsen one on the other hand are employed to model the airload in the Simpack® simulation setup. In the case of the steady and quasi-steady problems, *Expressions-based Force Elements* are defined to model only the lift, as there is no aerodynamic moment for both models. For the steady Glauert model, an expression for the lift is defined according to eq.3.7 by employing the required *subvars* from tab.3.4 and a Simpack® internally defined function to access the required kinematic state (pitch position) during the time integration. The same procedure is exploited also for the quasi-steady model from the lift formulation of eq.3.13. At this point, by selecting the *Force-Torque Expression Cmp* option as *Force Element type*, the previously defined expressions for the lift force can be recalled for the setup of the two simulated problems. Finally, important parameters to set are the *To Marker* as the airfoil *Marker A* (this is because the expressions are positive defined) and the *Reference Marker for calculation* as *Marker M_Isys*, because of what explained in the *Joint* paragraph.

Airload Force Element for the Theodorsen problem

Regarding the Theodorsen model, it is not possible to directly include the analytical formulations for the lift and aerodynamic moment according the lag-states method in the MBD simulation by means of *Expressions*, due to the presence of the additional ODEs for the lag-states. Alternatively, it could be possible to set a *User-defined Force Element* implementing the equations of the model, but this is not considered due to the considerable investment of time that this requires. Therefore, a solution is found by first externally computing the complete time-history of the aerodynamic force components from the full aeroelastic model by exploiting the equations in the appendix sec.B.5.2. Then, such time-histories are provided to the MBD simulation to compute the corresponding kinematic states for the same time range. This can be considered as a one-way coupling occurring only once and at the a level of the simulation setup, differently from the two-way coupling on a timestep-per-timestep basis discussed in ch.5. To implement this strategy, *Input Functions* are employed, because they allow to define functions from input data-tables *.afs files*[7] which can be subsequently employed for several purposes. In order to import *.afs files*, first of all it is necessary to define a search path among the *Global Parameters* where Simpack® looks for the required file and automatically loads it when the simulation file is opened. Once the input functions for the aerodynamic loads are defined, *Excitations from Input Functions* can be defined by employing a unitary scaling factor. Next, directly from the *Excitation Element* window, *u-Vector Elements* can be prescribed from the zero-derivative of the *Excitation*. Such *u-Vector* quantities are eventually supplied as input parameter to the *Force Element type "Force/Torque by u(t) Cmp"* to prescribe the z-force and the x-torque to apply to the *From Marker* A of the airfoil. The *From Marker* is set to A because the force values are generated externally with the negative sign, as shown in the appendix sec.B.5.2.

[7] A more in depth explanation of this tool can be found in the appendix sec.D.4.3 with reference to its setup and structure in the context of the coupling.

3.4. Presentation and analysis of results

This chapter concludes with the presentation and discussion of the results of the MBD simulations (red lines in fig.3.6, fig.3.7 and fig.3.8) compared with the numerical reference solutions (black dots in the same figures) of the same aeroelastic problems, where the aerodynamic models employed are the steady Glauert, quasi-steady and the time-approximated Theodorsen with the lag-states method. This comparison is expected to return very little differences between the Simpack® and the reference solution computed by a python routine, because their mathematical and numerical formulations are similar and potential error sources due to the MBD setup are minimized. More in detail, both solutions are obtained by modelling the physics of the problem in terms of lumped-parameters for the structural domain and analytical expressions based on the body motion for the fluid dynamics, leading to the compatible mathematical description consisting in systems of ODEs. Regarding discretization, as already pointed out in sec.3.3, compatibility between the two solutions can also be considered for this aspect, since a fixed-timestep numerical integration based on a linear implicit discretization is carried out. However, it is not possible to affirm that the numerical schemes are the same at the level of the implementation, because the Simpack® source code is not available and no further details are given in [65] regarding the properties and algorithm of the *Time Integration Solver* chosen, although this aspect is very likely to provide negligible effects. Finally, also some parameters of the Simpack® simulation setup, which play a role comparable to tuning parameters (for instance the *trans/rot sequence* in the *User Defined Joint type* or the *Mode of angle calculation* in the *Bushing Cmp Force Element type*), are set to those options returning the best match between the two solutions, and thus their impact can also be considered as negligible.

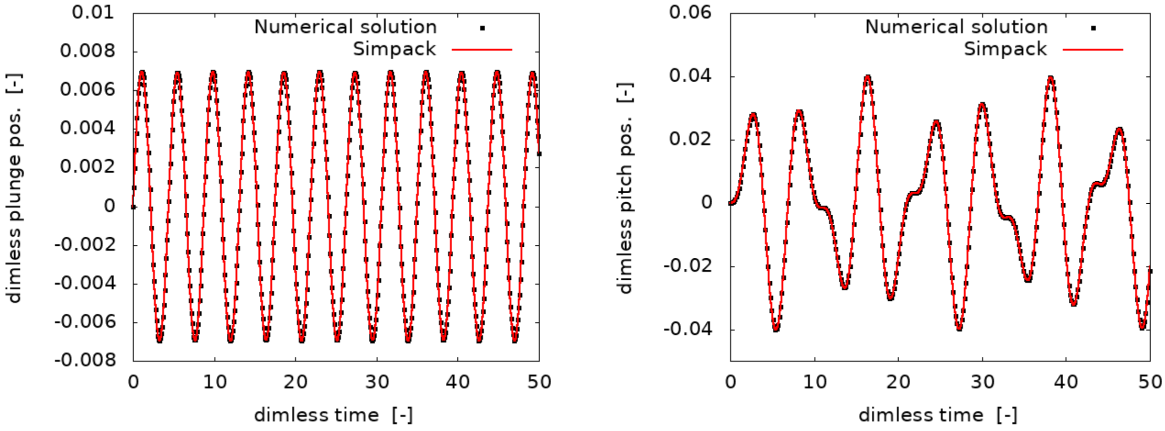


Figure 3.6: Comparison between the numerical and Simpack® dimensionless pitch and plunge positions time-histories for the steady aeroelastic problem computed by employing a dimensionless freestream velocity $\hat{U} = 0.6849$.

After motivating why a basically perfect comparison is expected, as already mentioned, both the numerical and the Simpack® solutions for the three problems are computed by considering the same integration timestep (0.0001s or units of dimensionless time due to the dimensionalization approach described in the appendix sec.C) and over the same time range (50s or units of dimensionless time). As shown in the global discretization error plots in sec.3.2.2, such a timestep is sufficiently small to affirm that the numerical solution is very close to the analytical one, leading to an equivalent comparison between the Simpack® solution and the analytical one. Moreover, both solutions employ the same initial condition (plunge velocity perturbation) in all the three problems and the inflow speed is set to 90% of the flutter velocity prediction returned by each aerodyanmic model shown in tab.3.2, leading to three different values of dimensionless inflow speed, namely $\hat{U} = 0.6849$ for the steady, $\hat{U} = 0.6174$ for the quasi-steady and $\hat{U} = 1.0746$ for the Theodorsen problem. Being said this, the expected perfect match between the Simpack® and the numerical reference solutions is actually observed in fig.3.6, fig.3.7 and fig.3.8 for the three aeroelastic problems considered and in the pitch and plunge position states over the entire simulation duration. The same can be observed also for the velocity and acceleration states shown in the appendix fig.F.1, fig.F.2 and fig.F.3. Furthermore. such a perfect match is kept also during intervals of the solution time-history characterized by rapid changes, or small time-scales. This

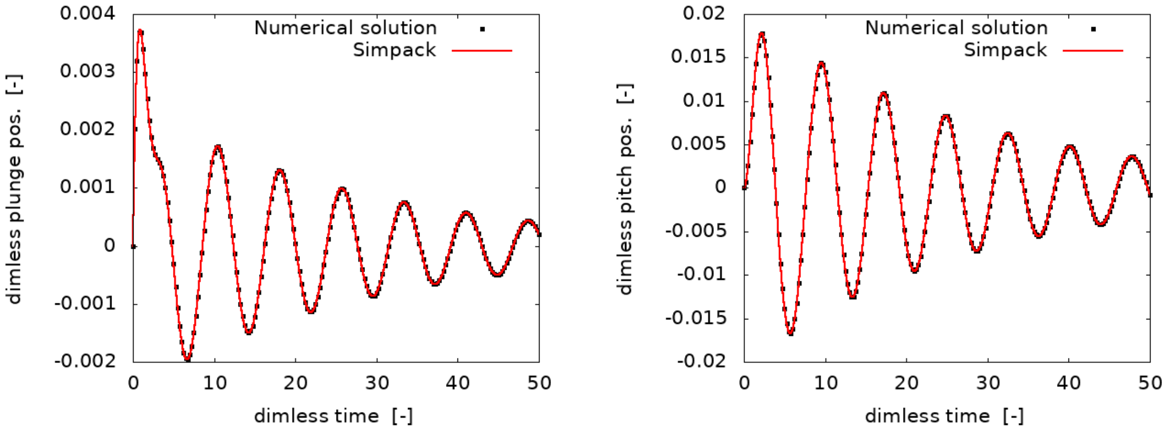


Figure 3.7: Comparison between the numerical and Simpack® dimensionless pitch and plunge positions time-histories for the quasi-steady aeroelastic problem computed by employing a dimensionless freestream velocity $\hat{U}$=0.6174.

is the case of the pitch kinematic states of the steady problem and in the initial transient for the plunge states of the quasi-steady problem. Therefore, these results show that Simpack® is able to provide accurate computations of the kinematic states of a rigid body under the effect of aerodynamic forces from analytical models.

If the results of such three aeroelastic problems are investigated from a physical point of view, a different dynamic response of the airfoil can be noticed. This is the direct consequence of the aerodynamic models employed and the aeroelastic effects that they trigger on the same structure shared between the three problems. On the one hand, the steady Glauert aerodynamic model in fig.3.6 returns an undamped response in both motion states with a single-component harmonic oscillation for the plunge motion, while the pitch motion is the result of a summation of several components. On the other hand, both the quasi-steady in fig.3.7 and Theodorsen in fig.3.8 with lag-states models provide a damped response in all the kinematic states considered. Concerning the Glauert model, this does not introduce any damping of aerodynamic nature in the system, which continues to oscillate at the limit of stability. In other words, the amount of work done by the aerodynamic forces onto the airfoil during an oscillation cycle equals the one returned by the body to the flow. In this way, the energy content of the airfoil does not change and oscillations do not reduce in amplitude. However, it is important to point out that such a limit-of-stability behaviour is not related to a flutter condition and it is just a consequence of the inadequacy of the Glauert model, which does not include both the inertial and the wake-memory effects. In reality, the limit-of-stability does not occur for the flow velocity considered with respect to the flutter boundary, and the oscillations are damped-out for the initial condition employed with pitch and plunge motions asymptotically tending to zero. Concerning both the quasi-steady and the Theodorsen models, they both include aerodynamic damping in the equations, leading to a dynamic response that, for a large time, correctly tends to zero. However, the response of both systems is still different, especially in terms of frequency of oscillation (almost double for the quasi-steady solution), damping of the response (the Theodorsen response appears to be more damped) and in the different initial transient behaviour of the plunge state. These differences are due to the fact that the Theodorsen model is more able to capture the complexity of the flow physics, by including the lift-deficiency effect caused by the wake memory effect - even though here only two aerodynamic states are considered. Besides, the Theodorsen model employs higher order derivatives to model the inertial effects on the flow due to the airfoil motion.

The assessment of the MBD simulation setup is relevant for the overall thesis project for multiple reasons, not just to prove that it can accurately compute kinematic states for prescribed aerodynamic forces. First of all, by understanding the working principle of the tool on simple problems, fundamental experience is developed and expertise is gained as well. In particular, all the tests and attempts carried out to improve the match between the two solutions allowed to better understand the working principle of the tool and to explore more in detail its functionalities, in particular in terms of *Time Integration Solver*, *Force Element* modelling and *Joint types* properties. Such knowledge is essential for the prosecution of the project, not only limited to this thesis, but also to the development of the coupling for 3D flexible

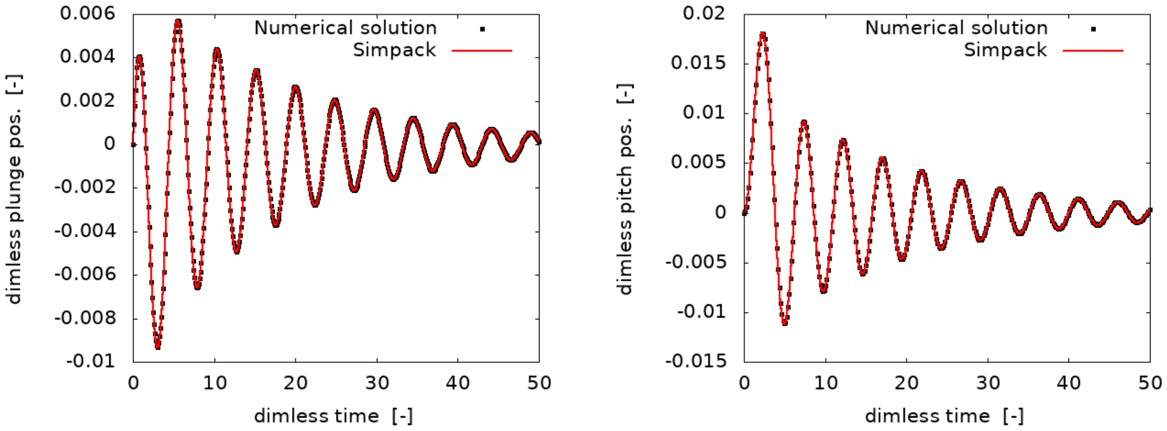


Figure 3.8: Comparison between the numerical and Simpack® dimensionless pitch and plunge positions time-histories for the aeroelastic problem featuring the time-domain approximated Theodorsen model with lag-states method. Both solutions are computed by employing a dimensionless freestream velocity $\hat{U}$=1.0746.

helicopter or wind turbine blades, which is the end goal of the research. Then, from a practical/technical viewpoint, this activity shows that a coupling methodology between Simpack® and an external tool, which does not involve user-defined force routines or co-simulation engines, can be still developed by exploiting existing and more straightforward features of the tool. The most important of these are *.afs input tables* to import external time-histories of aerodynamic forces and *Excitation-based Force Elements* to apply such airload time-histories to the simulation. As discussed in ch.5, this process is extended and automated to load data in a coupling loop iteration between Simpack® and an external routine which acts as a low-order surrogate of a CFD code.

4

PowerFLOW® CFD setup verification

The second step of this thesis project deals with an extensive verification activity of the PowerFLOW® CFD setup by assessing the accuracy of the tool in computing the aerodynamic forces resulting from prescribed pitching and plunging motions. As shown in the diagram of fig.4.1, the CFD simulation receives as inputs harmonic pitch - $\dot{\alpha}(t)$ - and plunge - $\dot{h}(t)$ - motion states and computes aerodynamic forces - $F(t)$ - which are then compared against the Theodorsen analytical unsteady aerodynamic model. Such comparison allows to evaluate the overall simulation setup and in particular the methods employed to model the airfoil pitching and plunging motions, namely a truly rotating mesh (LRF-approach) flow blowing/suction (transpiration wall velocity BC approach) and the body forces approach. As the flow conditions employed in this activity are based on low Mach and high Reynolds numbers attached flows, similarly to the experiments of [85] and the numerical simulations of [86], they are compatible with the assumptions of the potential flow theory. This justifies the employment of the Theodorsen model as reference solution for the lift and moment coefficient generated by the CFD simulations. Finally, this activity can be useful also to explain the results of the coupling between PowerFLOW® and Simpack® shown in ch.6.

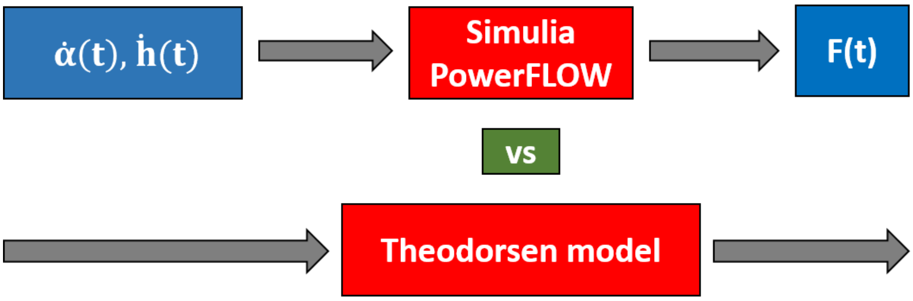


Figure 4.1: Diagram showing the main approach followed in this verification activity of the PowerFLOW® CFD simulation setup for the investigation of the aerodynamic forces generated by pitching and plunging airfoils.

This chapter start in sec.4.1 with a detailed description of the PowerCASE® simulation setup. Next sec.4.2 contains the test matrices employed for the pitch-LRF, wall-velocity-plunge and body-forces-pitch simulations. In sec.4.3, a short resolution study of the CFD setup with pitch-LRF is presented. Finally, the remaining three sections deal with the comments and possible explanations for the aerodynamic results of the three investigations for all the conditions outlined in the previous test matrices, namely pitch-LRF in sec.4.4, plunge-wall-velocity in sec.4.5 and pitch-body-forces in sec.4.6.

4.1. Description of the aerodynamic Simulation setup

The setup of the CFD simulation is developed from the one employed in [6] for the tuning of the body-force approach on 2D pitching airfoils, with modifications primarily in terms of body, refinement regions and boundary conditions to model the vertical plunge motion. The following subsections cover all the aspects related to the setup of the simulations carried out in this aerodynamic verification activity and to a good extent also in the coupling case. Regarding the latter case, the reader is referenced to ch.6.

4.1.1. Simulation options

The first aspect to take care when setting-up CFD simulations in PowerCASE® is to prescribe *Simulation Options*, which are very high-level parameters directly related to the PowerFLOW® solver. As this thesis is focused on 2D simulations of aeroelastic airfoils, the *2D option* is selected as *Dimensionality*. As discussed in the geometry subsection, PowerCASE® always needs a three dimensional geometry when setting up a simulation case, both with respect to the simulation body and primitives. When the 2D option is activated, the PowerFLOW® discretizer generates a lattice consisting in only one voxel along the spanwise third dimension, and computations are performed on a x-y plane at the location of z=z_{min}[52]. Besides, when dealing with 2D simulations of airfoils, it is always a good practice to employ a length size for the squared simulation volume considered much bigger than the airfoil chord length (a ratio of 500 is here employed). Moreover, given the cartesian mesh, it is possible to cover a large volume with a are relatively low number of additional voxels and without increasing significantly the simulation cost (if the near field VR regions are unchanged). This is done to minimize the effects of disturbance waves due to the simulation startup and boundary conditions on the airfoil surface loads. Other options are related to the flow conditions, such as *External Flow Type*, *Turbulence Model* as *Simulation Method* and *Low* ($Ma < 0.5$) *Mach Regime*, since the topic of this thesis deals with low speed incompressible flow conditions at high Reynolds number. Regarding the *Boundary Layer Transition Model*, this is enabled with the default option set to *Fully Turbulent*. Finally, the *Simulation Fluid* is based on air as an *Ideal Gas without Heat Transfer* and the *Solver* is set to work in (default) *Single Precision*.

4.1.2. Case variables

In PowerCASE® the most important variables defining the simulation case can be defined on a separate table were they can be stored in terms of name, unit, value and further attributes. This functionality of the tool is exploited to define quantities like the the free-stream pressure, Mach number and temperature, as well as airfoil chord, as shown in tab.4.1. Furthermore, also the pitch/plunge reduced frequency and amplitude of motion are defined as Case Variables, the parameters ξ_1 and ξ_2 used to tune the body forces and t_1 in the case of the wall velocity boundary conditions, as well as the default turbulence intensity and length-scale are required to compute in the calculation the free-stream turbulent conditions (kinetic energy, dissipation rate and eddy viscosity) employed by the solver for the turbulence model.

Name	Symbol	Value	Unit
Reynolds number	Re	$1.0181 \cdot 10^6$	[-]
Mach number	Ma	0.1	[-]
airfoil chord	c	0.46	[m]
free-stream velocity	U_∞	34.7223	[m/s]
free-stream pressure	P_∞	101325	[Pa]
free-stream temperature	T_∞	300	[K]
air kinematic viscosity	ν	$1.57 \cdot 10^{-5}$	[m^2/s]
default turbulence intensity	$I_{0_{turb}}$	0.1% local velocity	[m/s]
default turbulence length-Scale	$l_{0_{turb}}$	1	[mm]

Table 4.1: Main global simulation parameters employed for the CFD setup aerodynamic validation activity. Some of them (chord, temperature, viscosity and turbulence quantities) and also used for the aeroelastic Simpack®-PowerFLOW® coupled simulations.

4.1.3. Geometry

Airfoil simulation body

All the PowerFLOW® simulations carried out feature as body geometry a NACA 0003 symmetric airfoil shown in fig.4.2. This airfoil is chosen due to its geometric and flow dynamic similarity with the flat-plate on which the Theodorsen theory is based. Other airfoils are also tested, such as a NACA 0012 and a NACA 0006, but the NACA 0003 returned the best agreement in terms of lift coefficient in the case of periodic pitch oscillations modelled by the truly rotating mesh approach due to the larger thickness and the more rounded LE shape of the other two geometries. Therefore, this geometry is employed also for the simulations with periodic plunge motion and pitch motion modelled by body forces. Also thinner airfoils could be considered, but the range of AoA and reduced frequencies able to return attached flow conditions without LE separation would be too narrow to observe tangible amplitude or frequency effects. As already mentioned, the simulation body geometry has to be 3D, therefore a wing-airfoil with a given span needs to be imported in PowerCASE® as a ".stl" mesh file, which can be generated by a common CAD tool from the contour-points coordinates of the airfoil widely available in open literature, such as [81]. Besides, a thin TE is considered in order to avoid possible vortex shedding from the wake.

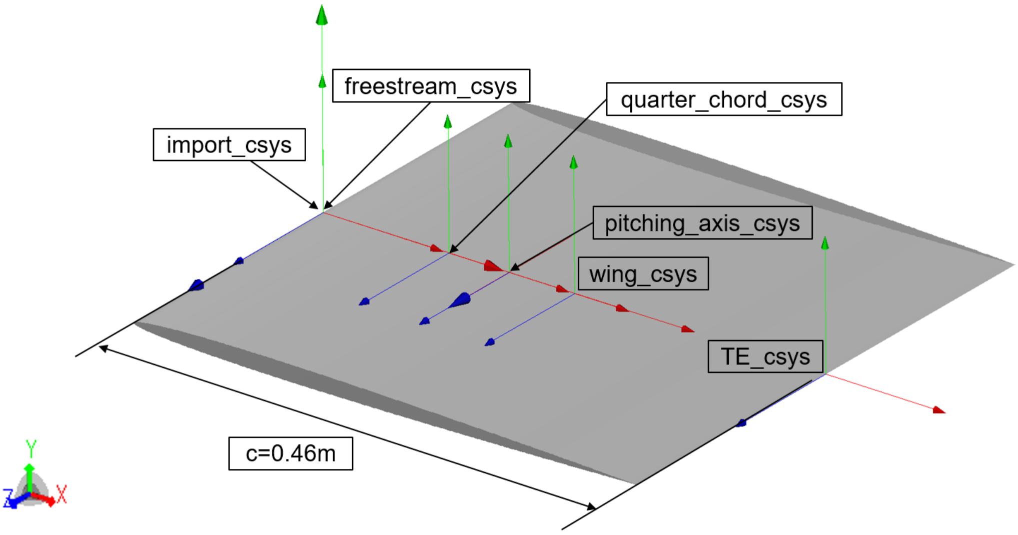


Figure 4.2: Airfoil NACA 0003 employed as CFD simulation body geometry with the defined coordinate systems.

Coordinate systems

For the simulation setup it is important to define coordinate systems for the positioning of the airfoil, geometry primitives, free-stream flow conditions and boundary conditions inside the computational domain. The coordinate systems employed are shown in fig.4.2 and are all are labeled with "_csys" at the end of their name. A rotation of the "freestream_csy", with respect to the z-axis normal to the simulation plane, defines an AoA in between the airfoil reference system and the wind-axis frame. The location of "freestream_csy" is the same as the "import_csys" (x position of the airfoil LE), which is instead used to position the airfoil when it is imported in PowerCASE®. The "quarter_chord_csys" (at the 25% of the airfoil chord) defines the location of the automatically generated VR regions (namely VR01, 2, 3 and 4), while the "pitching_axis_csys" defines the location of the LRF that models the airfoil pitch motion and its position in all the simulations, differently from what shown in fig.4.2, is the same as the "quarter_chord_csys". Finally, the "wing_csys" (at the airfoil center) is used to define the boundaries of the simulation volume (*SimVol*) and is located at the center of the wing, while the "TE_csys" helps in the positioning of the wake VR regions.

Primitives

Primitives such as boxes, cylinders, offsets, polylines, points, axis and symmetry planes can be defined in PowerCASE® for multiple reasons. In this setup, primarily boxes, cylinders, offsets and points are employed to define VRs, the LRF and the volumes for the body forces. Body forces to simulate the effect of a pitch motion are prescribed on fluid volumes enclosed by cylinders at the location of the LE and TE, similarly, the LRF requires an axis-symmetric primitive enclosing the whole geometry that is expected to rotate - in this case the entire airfoil simulation body - without intersecting more than two VR regions in order to avoid discretization errors [52]. Regarding the user-defined VR regions, boxes and cylinders (7, 6 and 5 are offsets of the LRF) are used to define refinement regions in the far field, as shown in fig.4.3, while offsets of the airfoil geometry (8, 9, 10 and 11) are used in the near field, see fig.4.4. On the other hand, boxes are employed in the case of the wake VRs extending downstream from the TE. Boxes are used also to set the boundaries of the computational domain, namely bottom, top, inlet and outlet shown in fig.4.3, and the simulation volume itself. Finally, points are defined to highlight

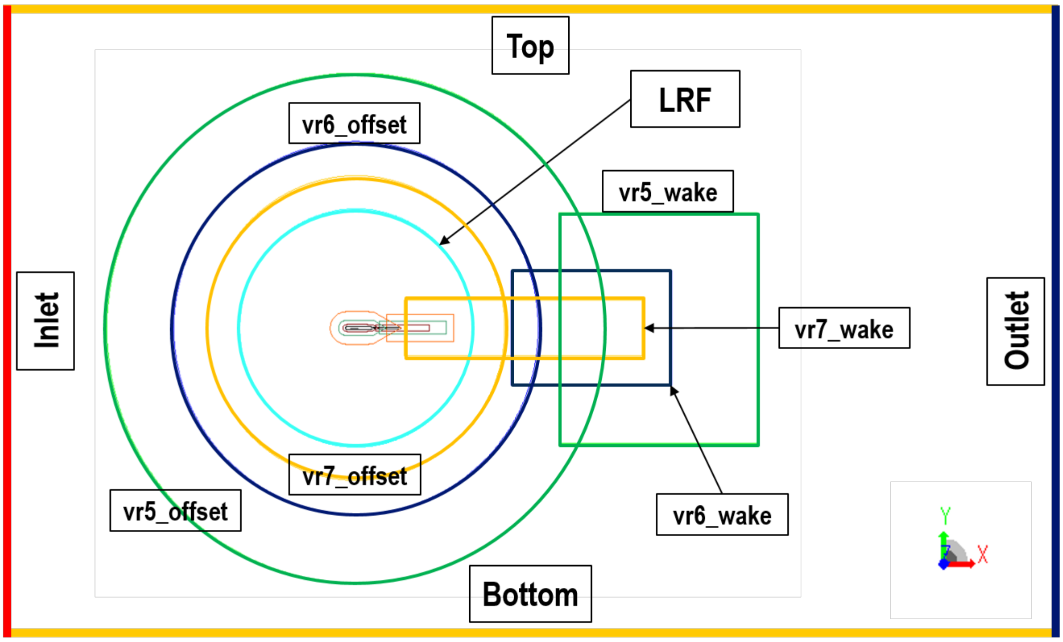


Figure 4.3: SimVol with boundary conditions and far-field VR regions for the fine simulation, image not to scale.

the location of the key airfoil points, such as the quarter-chord, the pitching axis and the LE and TE, but they do not have any other significant propose.

4.1.4. PowerCASE® parameters tab

This subsection contains all the most important aspects of the setup of a PowerFLOW® simulation and, in general, these are specified in the *Parameters tab* of the PowerCASE® GUI.

Reference frames: sliding mesh (LRF)

As already introduced in sec.2.2.2, to simulate the real pitching behaviour of an airfoil, a sliding mesh object can be defined in PowerCASE® with a truly rotating mesh and forces consistent with rotation (Coriolis and Centrifugal) applied to the fluid enclosed by the LRF volume [52]. To define a rotating mesh volume, a cylinder primitive is defined with a radius of 2 meters at the location of the pitching axis coordinate system. This radius is sufficient to enclose the airfoil, all the near-field airfoil-offset VRs and some of the wake VRs near the TE. After defining the geometric properties of the sliding mesh, an angular pitch velocity $\dot{A}(t)$ is prescribed for the LRF volume in the PowerCASE® calculation page. Contrarily to what carried out in [6], a combination of a cosinusoidal pitch motion of amplitude A_α and a rotation of the "freestream_csys" coordinate system with respect to the "default_csys" frame in fig.4.3 are employed. The first aspect is to minimize the spurious effects in the surface airload due to the LRF start-up (the initial LRF angular velocity is zero in this way), while the second one is to set a non-zero initial condition for the AoA[1]. Therefore, by setting an angle for "freestream_csys" equal to $-A_\alpha$ acting as CFD initial condition, the lift signal gets a zero mean value and AoA oscillations are in the range $-A_\alpha \leq A(t) \leq A_\alpha$ (similarly to [6]) with an initial value $\alpha_0 = -A_\alpha$. This leads to the following analytical expression for the pitch angular velocity of the rotating mesh in the PowerFLOW® "default_csys":

$$\dot{A}(t) = -A_\alpha \omega_\alpha \sin(\omega_\alpha t) \tag{4.1}$$

where ω_α is the angular frequency depending on the prescribed reduced frequency of pitch. Therefore, the corresponding time-variations of the pitch angle $A(t)$ and the AoA $\alpha(t)$ are:

$$A(t) = A_\alpha \cos(\omega_\alpha t) - A_\alpha; \qquad \text{and} \qquad \alpha(t) = A_\alpha \cos(\omega_\alpha t); \tag{4.2}$$

[1] This is required because the default initial value of the rotating mesh position, or pitch angle $A(t)$, is always zero, and leads to a pitch oscillation in the range $-2A_\alpha \leq A(t) \leq 0$ for a pitch motion of amplitude A_α.

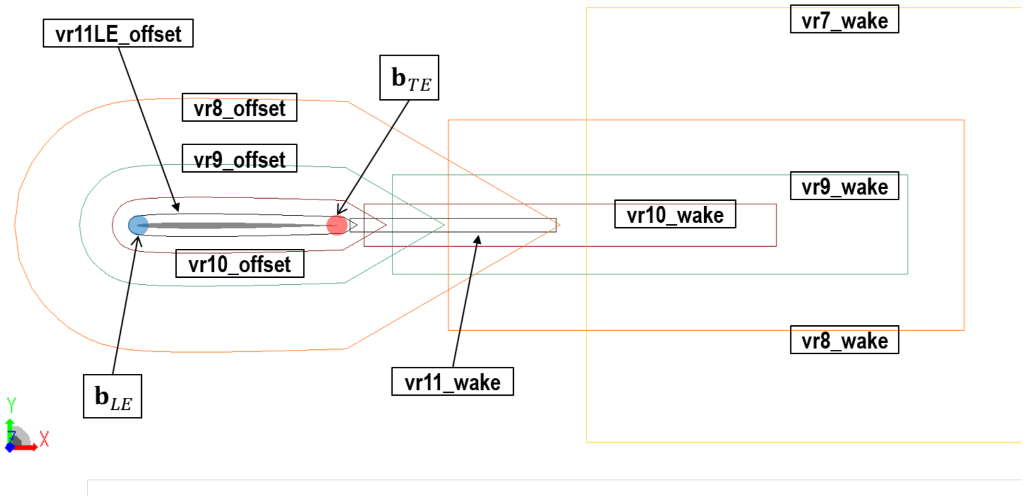


Figure 4.4: Near-field airfoil-offset and wake VR regions for the fine simulation geometry.

where the difference is due to the effect of the $-A_\alpha$ orientation of "freestream_csys", which is required to set a non-zero initial value for the AoA oscillation, namely $\alpha_0 = -A_\alpha$. The above expression for $A(t)$ is evaluated timestep-per-timestep by the PowerFLOW® solver together with all the other equations in the calculation page necessary for the CFD simulation setup, while for the CFD simulations with body forces the LRF angular velocity is simply set to zero.

Body forces applied to fluid regions

As already discussed in sec.2.2.2, a pitching airfoil can be simulated also by applying body forces within fluid volumes surrounding the surface of the body. This approach can replicate, up to certain conditions, the effect of a true airfoil pitching motion on the near-field fluid-flow by means of body forces applied to fluid volumes at the airfoil LE and TE (as shown by the $\mathbf{b}_{LE}$ and $\mathbf{b}_{TE}$ labels in fig.4.4). For this reason, as shown by [6], this method is expected to have limitations and to be effective only for limited values of the amplitude and frequency of motion and only after tuning the model to match the lift signal generated by the truly pitching airfoil. Without repeating the details in sec.2.2.2, here the focus is on the expressions of the pitch velocity and acceleration employed in this activity, which differ from the ones employed in [6] only in the harmonic function, given the previously defined pitch motion carried out by the sliding mesh LRF in eq.4.1. Consequently, a sinusoidal $\dot{\tilde{\phi}}(t)$ and a co-sinusoidal $\ddot{\tilde{\phi}}(t)$ amplitude modulated and phase shifted pitch velocity and acceleration are employed based on the following expressions directly implemented in the PowerCASE® calculation page and applied to the expression for the LE and TE body forces shown in eq.2.10:

$$\dot{\tilde{\phi}}(t) = -\xi_1 A_\alpha \omega_\alpha \sin\Big[\omega_\alpha\big(t - \xi_2 T_\alpha\big)\Big] \qquad \text{and} \qquad \ddot{\tilde{\phi}}(t) = -\xi_1 A_\alpha \omega_\alpha^2 \cos\Big[\omega_\alpha\big(t - \xi_2 T_\alpha\big)\Big] \tag{4.3}$$

where T_α is the period of pitch motion $T_\alpha = 2\pi/\omega_\alpha$ obtained from the prescribed reduced frequency. In the case of the body-force approach for pitch motion, there is no need to set an initial condition for the AoA ($\alpha_0 \neq 0$) by rotating the "freestream_csys", as the this model returns a cosine oscillation around an average zero AoA for the above prescribed $\dot{\tilde{\phi}}(t)$ and $\ddot{\tilde{\phi}}(t)$ already providing the desired initial condition for the angle of attack ($\alpha_0 = -A_\alpha$). Besides, spurious oscillations due to the motion start up are much smaller than those due to the sliding mesh. The only inconvenience is due to the increased cost of this method, as the above equations need to be calculated for each voxel within the cylinders surrounding the LE and TE and for each internal timestep of the CFD solver. Therefore, a rather small radius for such cylinders - one 16-th of the chord - is employed. In the case of LRF-simulations, body forces are switched-off by setting both $\dot{\tilde{\phi}}$ and $\ddot{\tilde{\phi}}$ to zero.

VR strategy

The VR regions shown in fig.4.3 and fig.4.4 are obtained by employing a set of parameters defined via equations in the calculation page. This is the case of the boxes for the wake VRs, whose size and position are determined based on multiples of the airfoil chord and the corresponding VR offset. Each VR region features a minimum offset distance - in other words VR thickness - of at least 20 times the local voxel size. As the smallest voxel size is the one of VR11, its offset is the smallest, while the largest offset is the one of VR5. This explains why the space between two consecutive VR regions that are offset of the airfoil or of the LRF increases from VR11 to VR5.

Fluid/solid and initial conditions

In this subtab of the PowerCASE® *Parameters tab*, the user specifies which geometric features are solid or fluid in order to instruct the PowerFLOW® discretizer where fluid voxels are required, where not and where to generate surfels for collecting data to generate surface measurement files. The airfoil and the boundary block (inlet, outlet, top and bottom) are the solid bodies in the simulation and within these volumes the discretizer will not generate fluid voxels. More in detail, the discetizer firstly fills the whole computational domain with voxels from the coarsest VR (VR0), then it generates the smaller voxels by splitting the coarser ones based on VR regions and finally, all the voxels inside solid regions are removed and surfels are created over solid surfaces. Regarding fluid regions, these are related to the SimVol and the body-force cylinders. In the first, no body forces are applied, while in the second the body forces simulating the effect of the pitch motion are activated by prescribing accelerations along the x and y directions. In both fluid regions, initial conditions for the pressure and velocity in the x-direction are given (P_∞ and U_∞ in tab.4.1) as well as initial conditions for Turbulence are provided based on the Intensity and Length Scale defined in the previous subsection.

Boundary conditions

Regarding boundary conditions, the simulation setup employs the *Pressure* and *Velocity Inlet type* which prescribes U_∞ along the only x-direction of the "freestream_csys" and P_∞ for the whole simulation duration. Besides, the same initial conditions for *Turbulence Intensity* and *Length Scale* discussed for the fluid regions are applied as boundary conditions. When it comes to the body surface, in the case of pure pitch simulations, the *Wall type* is set to *Standard Wall*, while in the case of pure plunge motion simulations, a *Velocity Wall* is considered where the velocity applied as BC is the one of the wall/body itself. The latter option allows, as already mentioned in sec.2.2.2, to simulate the effect of a rigid plunge displacement and it is the scope of this activity to exploit the possibilities and limitations of such method for this second purpose. When the *Velocity Wall* option is activated, a reference frame needs to be defined to prescribe the initial velocity to apply to the wall, and the LRF one is chosen so that the pitch and plunge motions can be simulated at the same time. Next, the "quarter_chord_csys" coordinate system is selected, for convenience, to specify the wall's velocity and, given the small amplitude of the pitching motions considered, only a y-component is prescribed based on a cosine velocity law:

$$\dot{h} = t_1 A_h \omega_h \cos(\omega_h t) \tag{4.4}$$

where A_h is the plunge amplitude, ω_h is the plunge angular frequency and t_1 is a coefficient required to tune the model to match a reference force signal. Considering only a y-wall velocity is an approximation acceptable for small AoA, as the coordinate system "quarter_chord_csys" does not take into account the rotation of the airfoil due to the pitching motion, but it would be more correct to include also a (very small) x-velocity, or at least scale the y-velocity by a $\cos(\alpha)$ factor. All the simulations, regardless of the wall type, feature the *Automatic* option for the *Boundary Layer Type* so that the transition from a laminar to a turbulent boundary layer is automatically computed by the aerodynamic solver based on its wall model and no surface roughness is applied to the airfoil boundary conditions.

Measurement files

Two measurement files are defined to sample fluid data. The first one is related is a *Composite Measurement by Face* for the airfoil surface airload. This file computes a surface airload over the surfels discretizing the simulation body by integrating the distribution of normal pressure stresses and tangential viscous stresses acting over the body. This is done for the whole simulation duration with a sampling interval corresponding to the local timestep of VR5 (64 computational timesteps corresponding to a range between roughly two and four thousand frames per period of motion depending on the motion frequency) and with a time averaging interval of the same duration. This leads to a high resolution in time, with an expected sampling frequency of several kHz depending on the resolution level. Finally. the aerodynamic moment is defined with respect to the pitching axis location. The other file is a *Standard Measurement by Region* collecting fluid variables within the LRF volume and surface quantities over the airfoil. Given the much larger amount of data to sample per each frame, the sampling period and the averaging interval are both set to a much larger time value than the previous measurement file, with a ratio between the motion-period and the frame time of 36 (or 36 samples per period).

Simulation duration

As the PowerFLOW® simulations are affected by an initial transient, like all CFD solvers, all the simulations carried out in for this verification activity of the aerodynamic setup have a duration of three periods of either pitching or plunging motion, depending on the type of phenomenon investigated. This is in order to reach a sufficiently converged periodic solution where the spurious oscillations due to the initial transient and boundary conditions do not significantly affect the aerodynamic forces.

4.2. Verification approach and test matrix

Once the main details of the CFD simulation setup are described (in particular rotating mesh, wall velocity BC and body forces), in this section, the approach employed in the verification of such setup is discussed, with focus on the test matrices. As already mentioned, the choice of considering low-speed, high-Reynolds number and attached flow conditions allows to employ the Theodorsen model as a reference solution for the airload computed by the CFD simulations. More in detail, the Theodorsen solution is therefore employed as reference for the simulations with pitch motion modeled by LRF and the plunge motion modelled by wall-velocity BC. Next, the pitch-LRF solution is employed as reference for the for the simulations with pitch motion modelled by body-forces.

	$k = 0.11$	$k = 0.17$	$k = 0.22$
$A_\alpha = 1°$	#1: c_l & $c_{m_{AC}}$	#2: c_l & $c_{m_{AC}}$	#3: c_l & $c_{m_{AC}}$
$A_\alpha = 2°$	#4: c_l & $c_{m_{AC}}$	#5: c_l & $c_{m_{AC}}$	#6: c_l & $c_{m_{AC}}$

Table 4.2: Test matrix for the pitch-LRF aerodynamic simulations with the corresponding variables considered.

After an initial resolution study carried out for the pitch-LRF simulation at a certain amplitude and reduced frequency condition, several simulations are performed for a range of different amplitude and reduced frequency conditions, as shown in tab.4.2, tab.4.3 and tab.4.4. The ranges considered cover the whole spectrum of reduced frequencies and amplitudes reached by both the 1 and 2 DoFs simulations of the PowerFLOW®-Simpack® coupling shown in ch.6. In the case of the pitch-LRF investigation, three values of the reduced frequency ($k = 0.11$, $k = 0.17$ and $k = 0.22$) and two values of amplitude ($A_\alpha = 1°$ and $A_\alpha = 2°$) are considered, leading to the 3-by-2 test matrix in tab.4.2. These reduced frequency and amplitude values are related to the typical elastic torsional deformations of helicopter rotors, as shown in [6]. For the plunge-wall-velocity verification, the same previous reduced frequencies corresponding to moderately and highly unsteady flow phenomena, are coupled with the two amplitude values $A_h = 0.5\%h/c$ and $A_h = 5\%h/c$, which can be considered as respectively very small and small compared to the airfoil chord leading to the other 3-by-2 test matrix in tab.4.3. Concerning the pitch-body-force, only a 2-by-2 test matrix (tab.4.4) is considered without the $k = 0.17$ cases of the LRF simulations. When it comes to the aerodynamic quantities considered, for the pitch-LRF and plunge-wall-velocity simulations, both the lift and the moment coefficient are considered, as both quantities are required to investigate the results of the 1 DoF (airfoil with free plunging and prescribed pitching motion as described in sec.5.3.1) and 2 DoFs (the same free pitch and plunge airfoil problem in ch.3) versions of

	$k = 0.11$	$k = 0.17$	$k = 0.22$
$A_h = 0.5\%h/c$	#1: c_l & $c_{m_{AC}}$	#2: c_l & $c_{m_{AC}}$	#3: c_l & $c_{m_{AC}}$
$A_h = 5\%h/c$	#4: c_l & $c_{m_{AC}}$	#5: c_l & $c_{m_{AC}}$	#6: c_l & $c_{m_{AC}}$

Table 4.3: Test matrix for the plunge-wall-velocity aerodynamic simulations with the corresponding variables considered. The parameter "c" stands for the airfoil chord $c = 0.46$ m.

	$k = 0.11$	$k = 0.22$
$A_\alpha = 1°$	#1: c_l	#2: c_l
$A_\alpha = 2°$	#3: c_l	#4: c_l

Table 4.4: Test matrix for the pitch-body-forces aerodynamic simulation with the corresponding variables considered.

the PowerFLOW®-Simpack® coupling. Conversely, in the case of the pitch-body-force simulations, only the lift signal is considered, since this approach is only employed in the 1 DoF coupling, where only the lift aerodynamic force plays a role in the aeroelastic formulation of the problem.

4.3. Resolution study

Before running all the simulation cases required to evaluate the capabilities and limitations of the LRF, wall velocity BC and body forces approaches in modelling pitching and plunging airfoils, a short resolution study of the CFD setup for pitch-LRF motion is carried-out to assess the effect of the mesh resolution on the accuracy of the airload computation. Three meshes are considered, with their data shown in tab.4.5 in terms of resolution, number of fine equivalent (fe[2]) voxels and surfels, as well as the total number of voxels and CPU hours. The motion conditions considered are those of the test case #5 in the LRF-simulations test matrix shown in tab.4.2, namely amplitude $A_\alpha = 2°$ and reduced frequency $k = 0.11$ applied to the expression for the LRF angular velocity in eq.4.1, which returns the pitching motion and the AoA variation in time shown in eq.4.2. The results of this analysis can be visualized in fig.4.5 for the lift and moment coefficients plotted against the AoA. To obtain both c_l and c_m, these simulations, and all the following ones, employ as scaling factors some of the reference conditions indicated in tab.4.1, namely the airfoil chord $c = 0.46$m, the free-stream velocity $U_\infty = 34.7223$ m/s and the free-stream density $\rho = 1.1766$kg/m^3.

Mesh	resolution	fe voxels	fe surfels	total voxels	CPU hours
Coarse	128	26.88k	0.79k	1,073k	8.3
Medium	256	36.03k	0.96k	1,128k	16.9
Fine	512	63.54k	1.59k	1,263k	45.3

Table 4.5: Essential data describing the features of the three meshes employed for this resolution study, with resolution defined as the number of voxels per chord (c=0.46m) in the finest VR region.

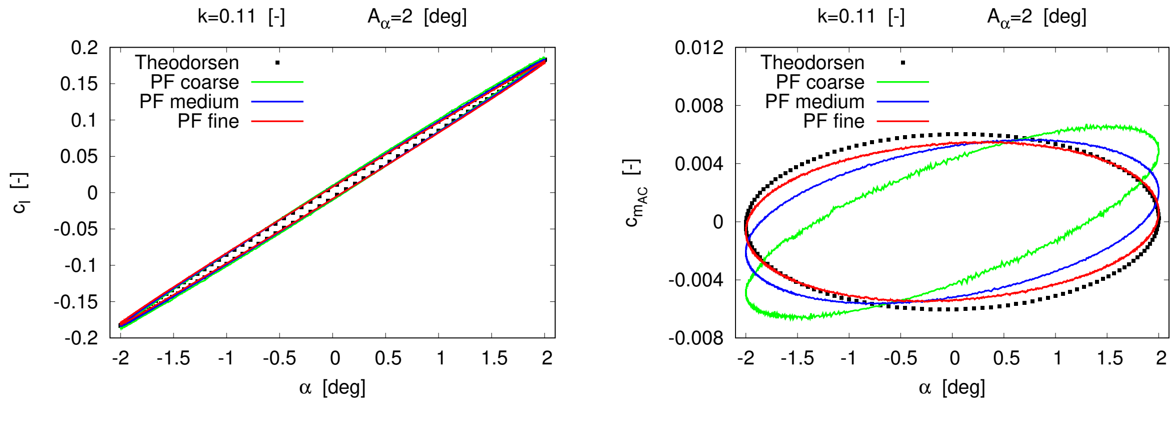


Figure 4.5: Mesh resolution effect on the lift and moment coefficients returned by the CFD simulations of a pitching airfoil modelled by sliding mesh (LRF).

Regarding the results for the c_l, it is quite clear that already a coarse resolution (green curve) is able to capture very well the shape of the hysteresis loop computed by the Theodorsen model, both in phase

[2] This is a weighted average of total number of voxels in the lattice which takes into account the resolution level of the VR regions employed in the simulation case. Information on the calculation of this quantity can be found in [87].

and in amplitude, at the locations of the maximum and minimum values. In the case of the $c_{m_{AC}}$, this coefficient requires the fine resolution (red curve) to be captured sufficiently well, although a certain mismatch is still visible and the possible reasons for such a discrepancy are given in sec.4.4. On the other hand, the medium and even more the coarse simulations show significant mismatches both in the phase and shape of the hysteresis loop. Always in the moment-coefficient plot, the convergence of the airload is monotonic with a bigger change from the coarse to the medium (blue curve) than from the medium to the fine resolutions. Given these results, all the CFD simulations shown in this chapter and the coupled aeroelastic simulations shown in ch.6 are carried out with the fine resolution. This can be motivated by the much better prediction of the c_m and the still reasonable cost (around 45 CPU hours for $k = 0.11$) of the 2D simulation with such a fine mesh. Regarding the simulations with wall velocity BC and body forces, the same fine resolution with 512 voxels per chord is employed before the two approaches are tuned.

4.4. Pitch motion modelled by sliding mesh (LRF)

After assessing the better adequacy of the fine resolution in capturing the aerodynamic forces (in particular the moment coefficient) generated by a truly pitching airfoil for a single pair of reduced-frequency and amplitude conditions, this mesh is here employed to carry out a sensitivity study of the same pitch-LRF approach for the range of reduced frequencies and amplitudes reported in tab.4.2. The aerodynamic lift and moment obtained by this investigation are shown in the plots of fig.4.6 up to fig.4.10 for all the test cases shown in tab.4.2, apart from test #4, which is instead already shown in fig.4.5 by the red line.

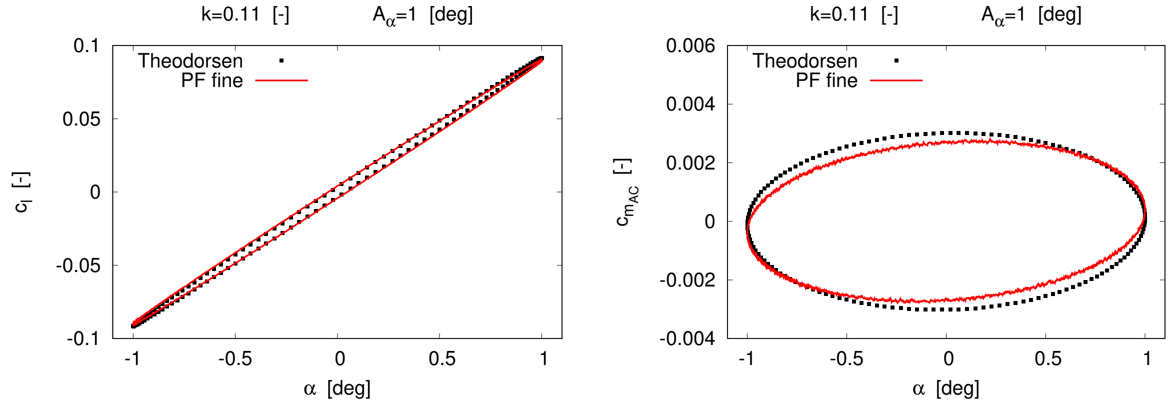


Figure 4.6: Lift and moment coefficients against the AoA for the first #1 pitch-LRF PowerFLOW® simulation.

Before discussing the aerodynamic results of this and the following investigations (transpiration velocity and body forces), it is necessary to point out that the physical and mathematical modelling of the flow dynamic behaviour employed by the reference Theodorsen model and the PowerFLOW® computation substantially differ. The comparisons in the investigations of this chapter are different from the comparisons in sec.3.4, where the Simpack® and reference solutions are based on similar physical models and mathematical formulations in the context of rigid body dynamics. In these investigations, on the one hand, the Theodorsen model is based on a simplified physical and mathematical description of the fluid-flow (the linear potential flow theory) which can account for the lift deficiency effect of the wake, but does not include any boundary layer modelling with transition from the laminar to the turbulent regime, which still occurs in the real pitching airfoil. On the other hand, the PowerFLOW® solution employs a high-fidelity physical description of the flow dynamics (the mesoscopic LBM) with more complex PDEs, especially in the case of turbulence and wall modeling. This allows to model the boundary layer and to provide a better wake description (as viscous phenomena are included) which can lead to a better estimation of the lift deficiency effect than the Theodorsen model. Nevertheless, it is important to mention that, for the flow conditions employed, the CFD solution is in the range of applicability of the Theodorsen theory and the two solutions should be therefore similar, but still some differences can occur and a perfect match cannot be expected. Furthermore, important differences are also in the simulation body geometry, with an elementary flat-plate shape employed by the Theodorsen model, in contrast to

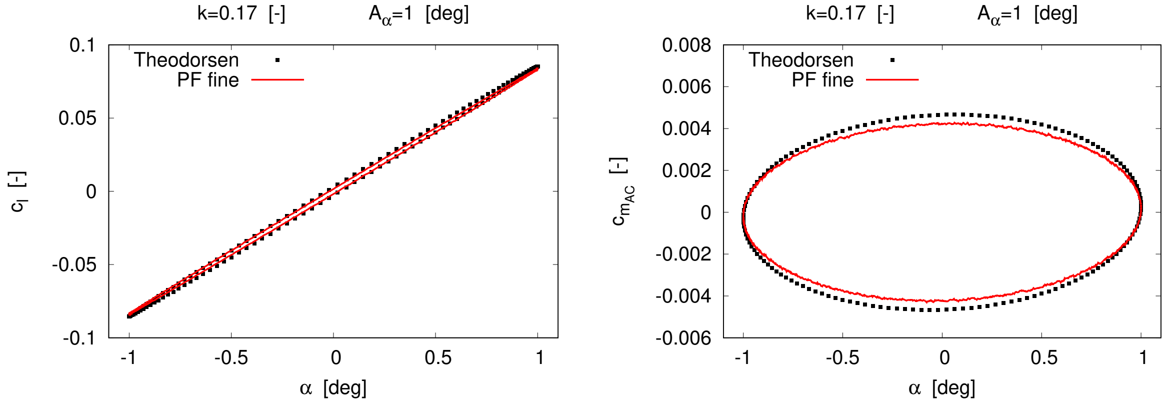


Figure 4.7: Lift and moment coefficients against the AoA for the #2 pitch-LRF PowerFLOW® simulation.

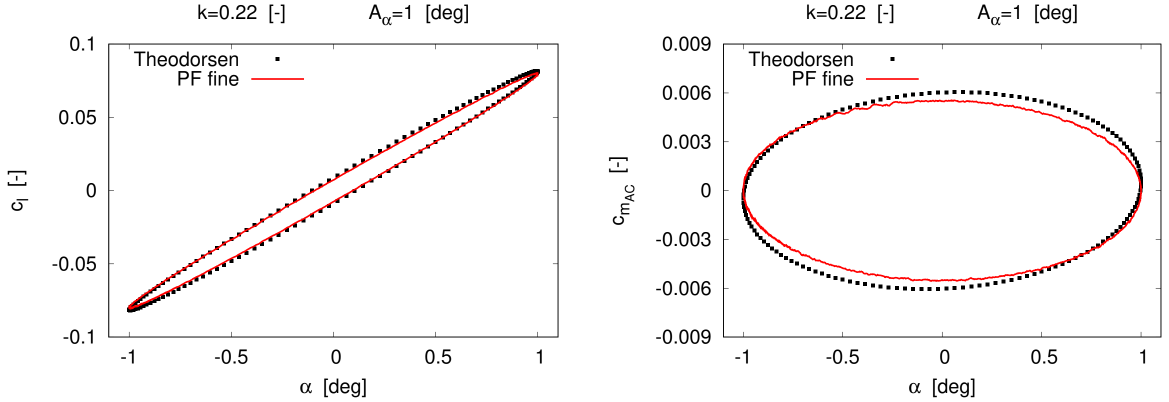


Figure 4.8: Lift and moment coefficients against the AoA for the #3 pitch-LRF PowerFLOW® simulation.

a realistic tessellated airfoil geometry based on CAD files employed by the CFD solver.

Being said this, when considering the lift coefficient results of all the six test cases, the most apparent observation is that all the PowerFLOW® simulations are able to capture very well the hysteresis loops, both in terms of maximum/minimum values of the c_l and width of the loop around the $\alpha = 0°$, where the difference between the upstroke and the downstroke airload is the largest. This very good match is achieved for all the reduced frequency-amplitude k-A_α pairs considered. Moreover, the CFD solutions correctly capture the decrease in the absolute value of $c_{l_{max}}$ and $c_{l_{min}}$ as k increases, while the effect of raising the amplitude from 1° to 2° is that of doubling the values of the lift coefficient without changing the shape of the hysteresis loop. A more careful observation of the plots reveals that, as k increases, the CFD solution slightly underpredicts the lift coefficient both in the proximity of its maximum during the upstroke motion and near the minimum in the case of the downstroke motion. This is also related to a slight underprediction of the Theodorsen $c_{l_{max}}$ and $c_{l_{min}}$ with an error below 2%. Although these deviations are almost imperceptible, they might lead to small discrepancies in the aeroelastic coupled simulation for both the 1 and the 2 DoFs problems. A final interesting aspect to point out is that the size of the lift hysteresis loop - which is related to the phase lag with respect to the input motion [86] - is larger for $k = 0.11$ than $k = 0.17$. This has to do with the phase inversion point for the lift force predicted by the Theodorsen model for $k = 0.144$ [86], where both the lift phase and the amplitude of the loop are zero.

In the case of the quarter-chord moment coefficient, the match between the PowerFLOW® and the Theodorsen airload is still good but not as satisfactory as the one returned by the lift coefficient. The amplitude mismatch can be quantified as around 10% the amplitude of the Theodorsen moment coefficient for the three reduced frequencies and the two pitch amplitude values considered. When it

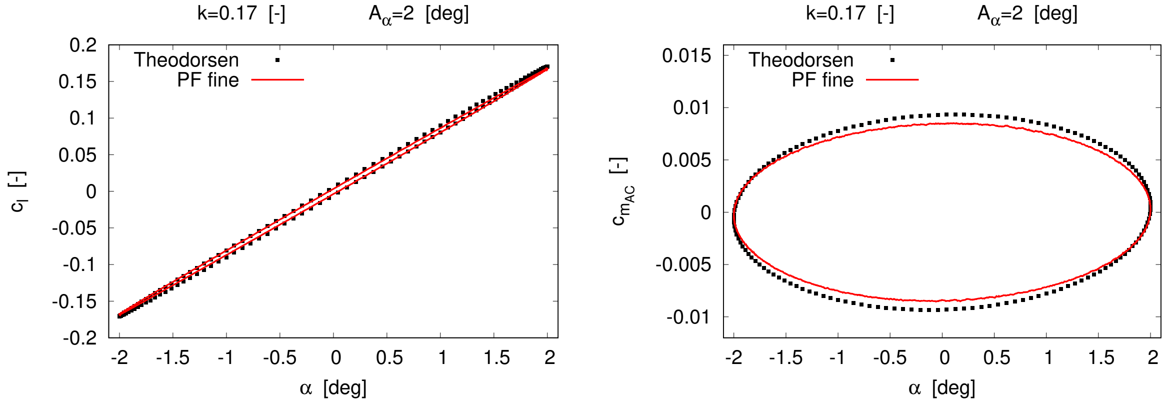


Figure 4.9: Lift and moment coefficients against the AoA for the #5 pitch-LRF PowerFLOW® simulation.

comes to the phase mismatch, this is not constant over the range of reduced frequencies investigated, with a maximum phase error for $k = 0.11$ of $\approx 11°$, a minimum of $\approx 5°$ for $k = 0.22$ and an intermediate $\approx 6°$ when $k = 0.17$. Furthermore, it is observed also that the quarter-chord Theodorsen moment is characterized by an almost constant phase-lag with respect to the input AoA oscillation. This leads to the conclusion that the CFD quarter-chord moment changes with the reduced frequency, differently from the Theodorsen one. A similar behaviour was observed also by Halfman [85] and, as suggested by Leishman [7], a possible explanation could be that the CFD prediction of the effective airfoil AC position is slightly shifted from the 1/4-chord prediction of the Glauert Theory on which the Theodorsen model is based. For this reason, it could be argued that the CFD prediction of the aerodynamic forces generated by a truly pitching airfoil might be more accurate than those provided by the Theodorsen model, as the more realistic description of the airfoil geometry can lead to a better prediction of the aerodynamic properties of the simulation body. In any case, the observed moment coefficient mismatch is expected to generate noticeable discrepancies also in the 2 DoFs aeroelastic coupled simulations, in particular in the pitch kinematic states for the first part of the simulation but also in the plunge states due to the energy transfer between the two coupled motions, especially for conditions in the neighbourhood of the flutter velocity. Finally, the effect of doubling A_α on both the Theodorsen and the CFD c_m-plots is just that of increasing their maximum and minimum values by a corresponding factor of two, without changing the shape of the hysteresis loop.

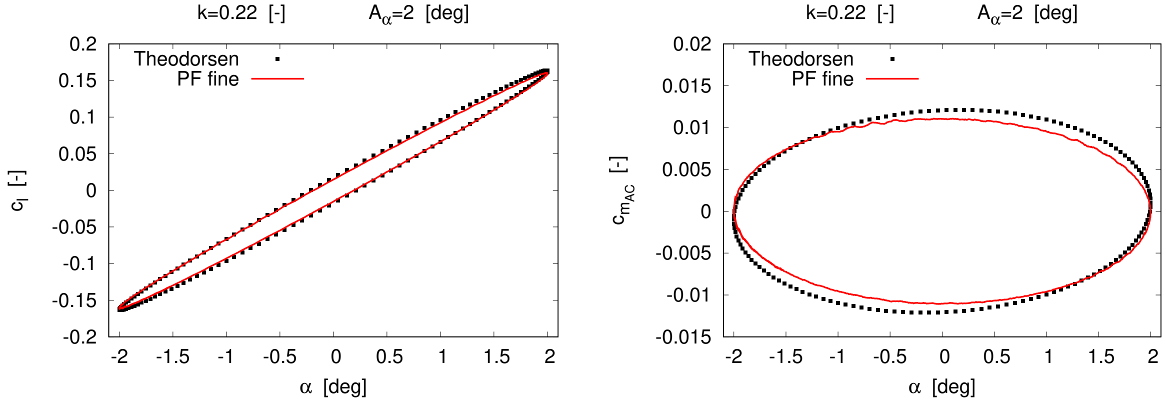


Figure 4.10: Lift and moment coefficients against the AoA for the #6 and last pitch-LRF PowerFLOW® simulation.

4.5. Plunge motion modelled by flow blowing/suction (normal Wall velocity boundary condition)

The second investigation carried out considers the lift and moment coefficients generated by a periodic plunge motion modelled by a normal wall velocity BC - which corresponds to adding flow blowing/suction to a free-stream flow kept parallel to the airfoil chord in this investigation - and compared against the reference results always returned by the Theodorsen model. Such a method, if directly applied to the simulation setup, returns excessive values of the lift coefficient (more than 30% error) and a significant under-prediction of the moment coefficient of more than 50%. For these reasons, as already explained in sec.4.1.1, the prescribed wall-velocity $\dot{h}(t)$ requires to be tuned by introducing an amplitude correction factor t_1, leading to the final expression shown in eq.4.4. In this analysis, values for the tuning coefficient are obtained by a manual tuning procedure, carried out to match at least the Theodorsen hysteresis loops of the lift coefficient, returning the values in tab.4.6 for the range of reduced frequencies - same as in the case of the pitch-LRF investigation - and amplitudes reported in tab.4.3. By looking at values for the amplitude correction coefficient in tab.4.6, this quantity changes very little (second decimal position) within the range of reduced frequencies considered and not at all for the amplitudes investigated, returning an average value of 0.723 for a fine mesh. Finally, it was also observed that t_1 changes by a factor 0.1 when employing a medium mesh moving from roughly 0.7 to 0.8 for the same range of k and A_h.

trimming coefficient	$k = 0.11$	$k = 0.17$	$k = 0.22$
$A_h = 0.5\%h/c$	$t_1 = 0.71$	$t_1 = 0.72$	$t_1 = 0.74$
$A_h = 5\%h/c$	$t_1 = 0.71$	$t_1 = 0.72$	$t_1 = 0.74$

Table 4.6: Values for the amplitude correction coefficient t_1 obtained by tuning the plunge-wall-velocity BC to match the lift signal returned by the Theodorsen model for the range of reduced frequencies and amplitudes considered.

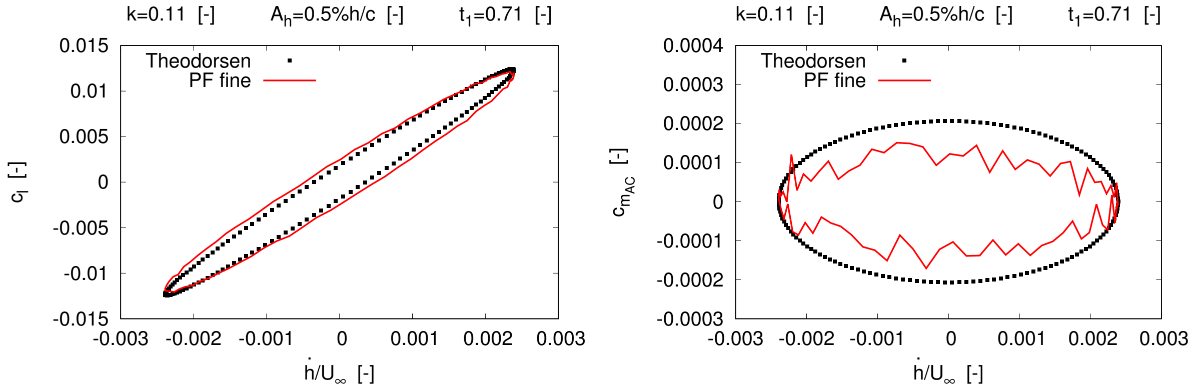


Figure 4.11: Lift and moment coefficients against the AoA for the #1 and first plunge-wall-velocity PowerFLOW® simulation.

Regarding the comparison with the reference solution, in addition to the different physical, mathematical and numerical models employed by the reference Theodorsen theory and the PowerFLOW® solution already mentioned in the previous section, in this investigation additional sources of mismatch need to be considered between the two solutions. The most important one is that the transpiration velocity can not be considered the same as a true vertical translation of the airfoil, although an appropriate blowing or suction of fluid should modify the streamlines of the flow in a similar fashion as the true motion close to the airfoil under certain conditions. The second one is related to the amplitude-tuning of the wall-velocity approach, which might be still imperfect in spite of the three decimal digits consider for t_1. Therefore, larger discrepancies are expected in this comparison compared to the previous one, in particular for large amplitudes of motion, as [52] suggests.

All the results obtained with this investigation are shown from fig.4.11 up to for fig.4.16 for both the lift coefficient and the quarter-chord moment coefficient. Globally, it could be argued that the wall-

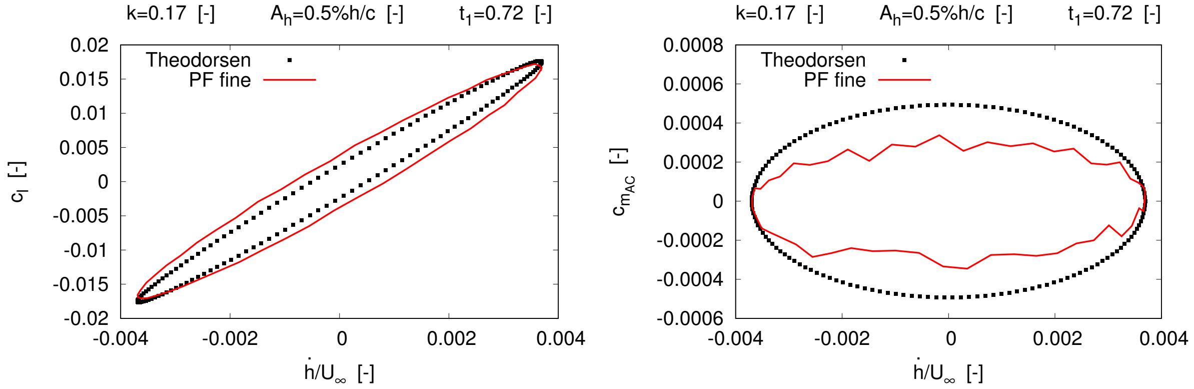


Figure 4.12: Lift and moment coefficients against the AoA for the #2 plunge-wall-velocity PowerFLOW® simulation.

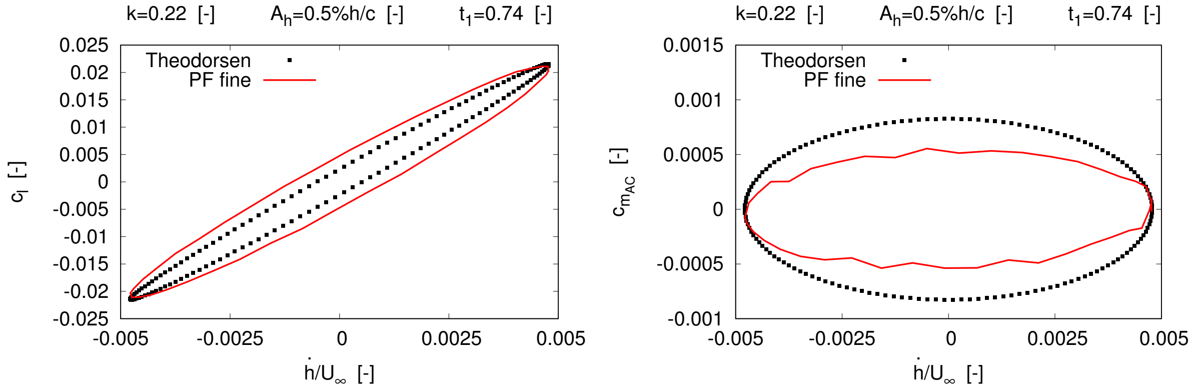


Figure 4.13: Lift and moment coefficients against the AoA for the #3 plunge-wall-velocity PowerFLOW® simulation.

velocity BC method, when tuned on the c_l, works quite well for the lift coefficient prediction, but not for the moment coefficient. The reason why the method is tuned on the c_l and not also on the c_m is because, during the several attempts carried out, only a slightly improvement of the c_m is obtained at the cost of a significantly large c_l worsening. Moreover, for the largest amplitude and for increasing k, a significant worsening in the loop-shape of the aerodynamic moment is observed, leading to a complete

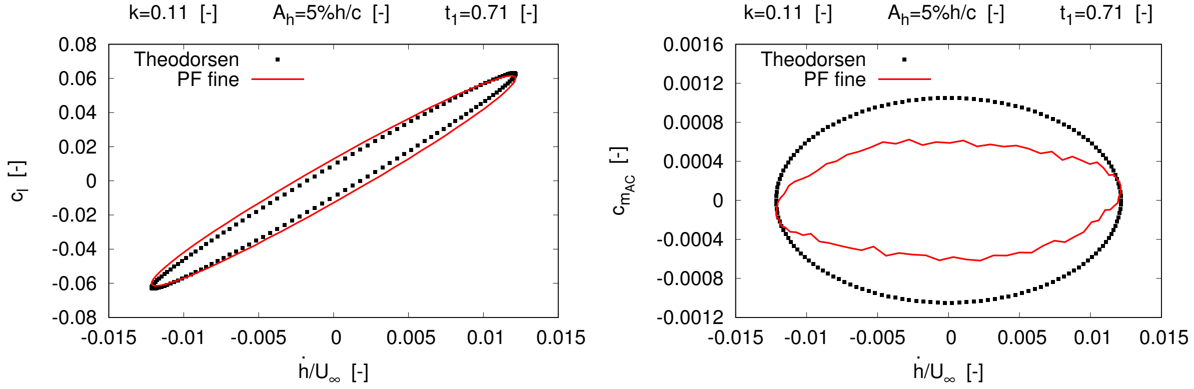


Figure 4.14: Lift and moment coefficients against the AoA for the #4 plunge-wall-velocity PowerFLOW® simulation.

loss of its elliptic character, as fig.4.15 and fig.4.16 partially show. For these reasons, it is decided to tune the model by considering only the lift coefficient at the cost of accepting up to a 50% under-prediction of c_m.

When the c_l curves are analyzed more in depth, in all the A_h-k pairs considered, the wall-velocity overestimates the values of the lift coefficient both in the upstroke and in the downstroke phases for a significant portion of the $\dot{h}/U_\infty$ range covered. This is primarily a phase-mismatch and it grows as both k and A_h increase, because not just the amplitude is relevant, but also the speed that it takes to span such motion, in particular the ratio between the plunge velocity and the freestream one. Moreover, the $c_{l_{max}}$ and $c_{l_{min}}$ values are slightly under-predicted and this has to deal primarily with the imperfect manual tuning of the model to match the Theodorsen lift or the probably insufficient number of digits employed for the t_1 parameter. Both phase and amplitude mismatches are expected to impact the results of the aeroelastic coupling, both in its 1 and 2 DoFs versions. The phase mismatch in these results could be minimized by introducing a phase-shifting coefficient in the wall velocity formulation of eq.4.4, but this has not been implemented because in no version of the coupling the plunge motion is prescribed, and thus it can not be phase-shifted.

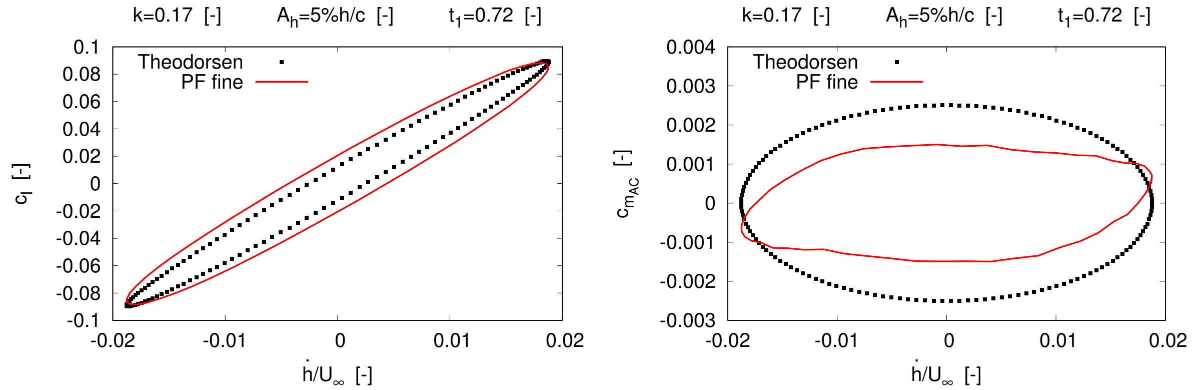


Figure 4.15: Lift and moment coefficients against the AoA for the #5 plunge-wall-velocity PowerFLOW® simulation.

If also the moment coefficient plots are considered, it can be noticed that, contrarily to the pitch motion, increasing k leads to an increase of both $c_{l_{max}}$ and $c_{m_{max}}$ for the same value of A_h. Whereas the width of the hysteresis loops doesn't change much for the for different k values and the same A_h. Another interesting observation is that when A_h is increased from 0.5% to 5% (1 order of magnitude), both $c_{l_{max}}$, $c_{m_{max}}$ and the ratio $\dot{h}/U_\infty|_{max}$ scale-up by a factor of 5. Next, because in the moment plots of fig.4.11 and fig.4.12 the motion amplitude and the ratio between the plunge speed and the freestream

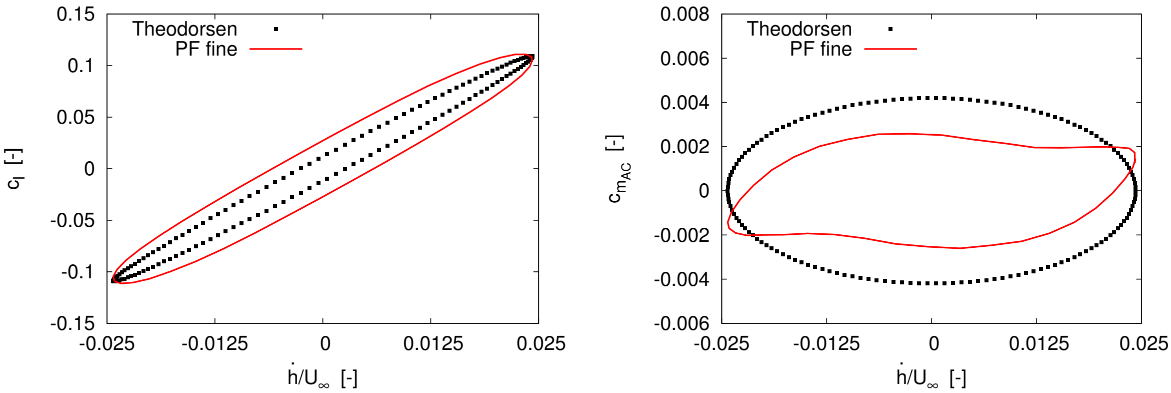


Figure 4.16: Lift and moment coefficients against the AoA for the #6 and last plunge-wall-velocity PowerFLOW® simulation.

one are very small, random fluctuations (associated to the unsteady nature of PowerFLOW® solution) are obtained, returning a rather spiky behaviour which makes it complex to evaluate the percentage of mismatch with the Theodorsen result. Another peculiar behaviour in the c_m can be observed in fig.4.16 and fig.4.15, where the elliptic shape of the reference is largely compromised by the appearance of two lobes near $\pm\dot{h}/U_\infty|_{max}$. A possible explanation could be related to the fact that the Theodorsen model deals with flat plates, while the CFD simulations employ a symmetric airfoil with a rounded LE (although its very small radius) and this might lead to a different flow behaviour in terms of pressure and skin-friction distributions. In conclusion, all the observed mismatches can be summarized by saying that the larger k and A_h get, the more critical it is the modelling of the airfoil translation in the CFD solver by applying a transpiration velocity to model such motion.

4.6. Pitch motion modelled by body forces

The last step of this verification activity is related to the comparison between the pitch motion modelled by the body force approach and the same obtained by the sliding mesh LRF technique used as reference solution. This comparison is aimed at assessing the limitations of the first approach in capturing the airload generated by the truly rotating geometry. This is important for the long-term goal of the project, as a spanwise-changing torsional deformation of a flexible blade cannot be directly simulated in PowerFLOW® with a sliding mesh, but it requires equivalent approaches, for instance the body-force one, as done in [6]. Therefore, the expected differences appearing in this investigation are only related to the airfoil motion modelling and imprecisions in the tuning of the body-force method.

trimming coefficient	$k = 0.11$	$k = 0.22$
$A_\alpha = 1°$	$\xi_1 = 39.3$ & $\xi_2 = -0.53$	$\xi_1 = 10.1$ & $\xi_2 = -0.56$
$A_\alpha = 2°$	$\xi_1 = 40.2$ & $\xi_2 = -0.53$	$\xi_1 = 10.3$ & $\xi_2 = -0.56$

Table 4.7: Values for the amplitude ξ_1 and phase ξ_2 correction coefficients required to tune body forces to match the lift signal returned by the pitch-LRF simulations for the range of reduced frequencies and amplitudes considered.

Regarding the tuning of the method, as extensively explained already in sec.2.2.2 and in sec.4.1, the body force method requires to be tuned both in amplitude and frequency to match at least the lift signal given by the sliding mesh simulations. Such trimming activity is manually carried-out on the range of reduced frequencies and amplitudes shown in tab.4.4 and it returns the ξ_1 and ξ_2 correction coefficients shown in tab.4.7 applied to the pitching velocity and acceleration expressions in eq.4.3. The computed values for the correction coefficients shown in tab.4.7 reveal that the factor ξ_1 increases very little when increasing the amplitude of motion from to 1 to 2 degrees, while it scales up by a factor of four when doubling the reduced frequency. Regarding the factor ξ_2, this one doesn't significantly change for changing k, and, at a first instance, it is approximated by the same values for changing A_α and fixed k. This means that between the pitch motion modelled by LRF and the same modelled by body forces there is a phase delay which is slightly more than half a period.

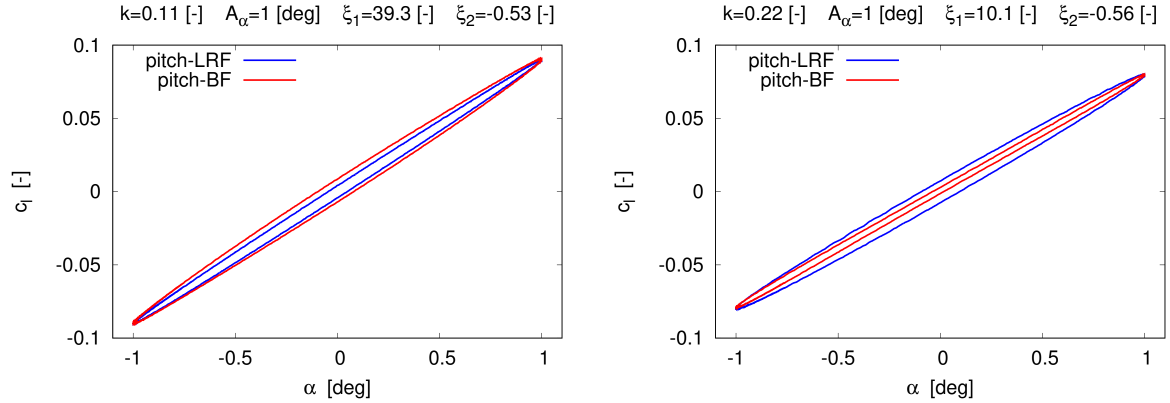


Figure 4.17: Lift and moment coefficients against the AoA for the #1 and #2 pitch-body-force PowerFLOW® simulations.

If the lift coefficient results show in fig.4.17 and fig.4.18 are considered, it is possible to notice a good amplitude match between the LRF (blue line) and the body force (red line) simulations over the range of reduced frequencies and amplitudes considered, with very small mismatches for the largest $A_\alpha = 2°$ value. Regarding the phase-match, this is small only in fig.4.17, where $A_\alpha = 1°$, while it increases for $A_\alpha = 2°$ in fig.4.18. Next, for the smallest value of reduced frequency ($k = 0.11$) the body-force lift overestimates the phase shift with respect to the input motion, while in the case of the largest value $k = 0.22$ the opposite occurs. Moreover, for $A_\alpha = 2°$ and $k = 0.11$ and even more for $A_\alpha = 2°$ and $k = 0.22$ in fig.4.18, the phase mismatch is not constant over the period of motion, due to the clearly irregular shape of the body-force curves. This is can be considered as an intrinsic property of the lift signal returned by the body-force method, which for high k and A_α features higher-order components of the harmonic input signal modifying the "smooth" shape of the LRF lift signal, as already observed by [6] and shown on the plots of fig.2.7. Such phenomenon can be considered as a limitation of the model in its current implementation and it is very unlikely that even an automatic optimization routine (for a more accurate phase-shifting of the body force signal) will help in fixing the shape of the hysteresis loop. Therefore further investigation are required to understand its origin and to find possible solutions. Finally, the plots related to the moment coefficient generated by this body-force method are not shown, as the c_m is not employed in both the 1 and the 2 DOF versions of the aeroelastic coupling. Furthermore, the mismatch is in this case much bigger than the one observed for the wall-velocity aerodynamic moment, with an amplitude overestimation of more than 5 times $c_{m_{max}}$ and a phase error of more than a quarter of period.

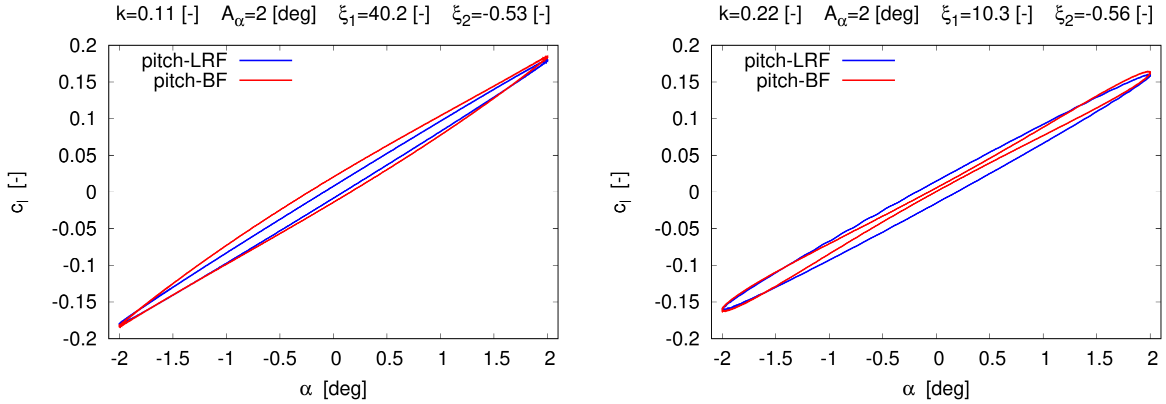


Figure 4.18: Lift and moment coefficients against the AoA for the #3 and #4 pitch-body-force PowerFLOW® simulations.

5
Coupling methodology

In the two previous chapters, attention is dedicated to the structure and fluid-flow domains as two separate entities, with the main goal of verifying the accuracy of the Simpack® and PowerFLOW® setups. More in detail, in ch.3 it is proved that the MBD setup can compute very accurate kinematic states under the effect of aerodynamic forces from analytical models. On the other hand, in ch.4 it is shown that the CFD setup can return correct lift forces but less accurate moments for harmonic pitching and plunging motions, although this depends on the technique employed to model the body motion in the simulation. In this chapter, the two domains are fully (two-way) coupled to achieve a proper FSI aeroelastic simulation. This means that at every timestep, the calculation of the kinematic states by the MBD solver is dependent on the airload computation by the aerodynamic solver and vice-versa. This leads to a loop between the two domains, as shown in fig.5.1, which requires the development of a coupling methodology that needs to be executed quickly and efficiently for a large number of iterations.

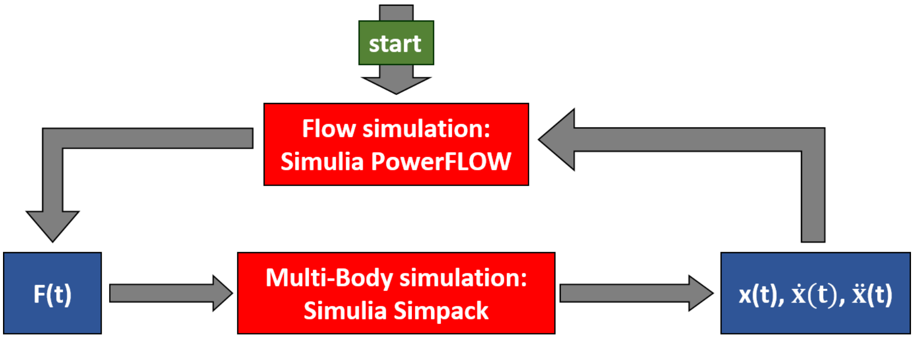

start
Flow simulation: Simulia PowerFLOW
F(t)
Multi-Body simulation: Simulia Simpack
x(t), ẋ(t), ẍ(t)

Figure 5.1: Diagram showing the (very) high-level functioning of the coupling loop between the LBM-based flow solver PowerFLOW® and the MBD simulation tool Simpack®.

This chapter is essentially dedicated to the description of this methodology starting from its definition in sec.5.1. Then, the properties of an initial coupling demonstrator between a reduced order aerodynamic routine and Simpack® are discussed in sec.5.2. The following sec.5.3 contains the results of the verification activity of such methodology on two test problems. Finally, the specific aspects related to the coupling between PowerFLOW® and Simpack are described in sec.5.4, in addition to what already mentioned for the coupling demonstrator. The results of the MBD-CFD coupling are instead discussed in ch.6.

5.1. Definition of the methodology

5.1.1. Solvers limitations/capabilities and project context

In the design process of a coupling strategy between PowerFLOW® and Simpack®, several constraints in terms of resources, time and expertise, as well as limitations in the tools have to be taken into account. The following paragraphs address each one of them.

Proprietary software

As both Simulia PowerFLOW® and Simpack® are commercial computer programs, their source code is not openly available. Besides, even if the thesis project is carried out at Dassault System which holds the rights for both tools, the author still has no access to the source code. Therefore, a coupling has to be developed based on existing features of the tools or by exploiting customization functionalities that allow to extend their capabilities. However, with respect to this aspect, both programs do not allow the same customization potential, with PowerFLOW® offering much less room for customization than Simpack®.

Simpack® customization

On the other hand, Simpack® can be widely customized, allowing a considerable extension of its capabilities by means of user-defined subroutines to create highly targeted modelling elements. Nevertheless, this approach requires a significant investment of time in understanding in depth the tool, programming and debugging. Besides, user-routines requires specific coding (Fortran) and computer skills that the author currently does not posses, and the time required to acquire them is beyond the one offered by a master thesis. Therefore, the coupling has to be reached considering what is already available in the computer programs, as it is the most reliable and robust solution, even though it might not be the most performing or efficient.

Co-simulation engine

The most common and recommended way to achieve a coupling between Simpack® and an external tool is by means of co-simulation engines, as this allows to minimize data transfer time and can be used to develop highly robust and general couplings [65]. However, the main obstacle preventing this approach is PowerFLOW®, as it does not allow - in the current version - to be coupled with external tools based on memory sharing and TCP/IP interface. This could be potentially solved by having access to the source code. But this is not allowed by the project context, and even if possible, the investment of time, resources and expertise is far beyond the scope of this thesis and the author's possibilities. This leads to the definition of a first important feature of the coupling, namely the coupling medium, which is file input/output. This method, although not the most efficient and performing, is simple and can help during the debugging process, as the user can easily check the correctness of the data being transferred at any time [88].

Look-up tables

As the coupling medium is determined, the aerodynamic forces and the kinematic states exchanged between the two solvers can be imported by means of look-up tables, and both PowerFLOW® and Simpack® do that in a reliable and robust way. As explained in sec.2.2.2, input tables can be used to provide data to the CFD solver in multiple formats, and the PowerFLOW® documentation [52] is quite rich of helpful examples. Besides, this method was already employed by the author's team members in previous projects, for instance [89], [6] and [19]. On the Simpack® side, table look-up by means of *.afs* input functions is employed already in this research project to carry out the Simpack® verification activity. As described in ch.3, this method works well to generate aerodynamic force elements in the MBD simulation based on time-excitations defined via input-tables. Therefore, the verification and the experience developed by the author with this technique make it reasonable to exploit it also for the coupling. Besides, also the Simpack® documentation [65] is a valuable reference in describing all the possible ways of using this functionality.

Simpack® scripting languages

Simpack® can be easily scripted to automate recurrent tasks related to simulation setup, solvers execution and post processing. The scripting language is based on a variant of JavaScript (EcmaScript) [71] and users can readily develop scripts, also with the help of the macro recording tool available on the GUI of the tool. These scripts can then be provided to the Simpack® solver or Post-Processor and can be run in background mode without the need for any user interaction. Besides, the process of running these scripts can be automated and included as a step of a coupling interface managing the data transfer between the two tools.

Explicit FSI schemes

It was already mentioned in sec.2.1.2 that the added mass effect due to fluid displacement becomes relevant in hydroelastic applications due to the comparable structure and fluid densities, prescribing implicit (tight) FSI schemes. On the other hand, in aeroelasticity this is not the case, as the fluid density is much smaller than the one of the structure. Therefore, explicit (weak) FSI schemes can still provide accurate results with a sufficiently small coupling timestep. Therefore, aeroelastic problems such as wing flutter, limit cycle oscillations, the "simple" blade deflection deformation of a helicopter and dynamic stall can be potential applications of this methodology. Furthermore, the explicit nature of the PowerFLOW® time-marching algorithm - due to the much smaller computational timestep than URANS or LES - suggests even more the employment of explicit coupling schemes with subcycling of the fluid phase (PowerFLOW® internal timesteps during a single coupling iteration).

Simpack® adaptive timestep integrator

Simpack® offers a wide range of time-integration methods to solve MBD systems by means of fixed or adaptive timestep schemes. Although fixed methods are used in ch.3 for consistency reasons with the numerical reference, for the sake of the coupling, adaptive schemes can be considered. The latter are more general, robust and reliable, especially for stiff systems [65]. This leads to subcycling also of the structure side of the problem, which might be unnecessary given the larger time scales of the structures compared to fluid-flows. However, as this choice will not lead to any increase of computational cost or complexity in the simulation setup, it seems to be a reasonable choice.

5.1.2. Essential features of the coupling

Based on the aspects pointed out in the previous subsection - namely limitations and possibilities in the solver, the constraints defining the research project, the time available to complete the master thesis and the knowledge/experience of the author with the tools/topic - the coupling between PowerFLOW® and Simpack® can be achieved by considering the following crucial aspects:

1. Coupling medium: ASCII file input/output or look-up-table method. The generation of such coupling files is the main task of the coupling interface, which also takes care of the communication with the two solvers and the exchange of the required data.

2. PowerFLOW® is both the starting point of the complete aeroelastic simulation and of the coupling loop. This means that the coupling consists in a single PowerFLOW® simulation spanning the whole aeroelastic calculation and one Simpack® simulation of one structural timesteps duration carried out per each coupling iteration.

3. An explicit, sequential, first-order accurate in time FSI coupling scheme is considered with subcycling for both PowerFLOW® (due to the explicit formulation in time of the underlying LBM) and Simpack® (motivated by the higher robustness and better performance of the adaptive-timestep numerical integrators).

4. Automated update of the Simpack® simulation setup, time integration execution and post-processing by means of dedicated Javascript routines.

5. The PowerFLOW® look-up-table method is automatically carried out by the simulator once the correct input files are provided. The aerodynamic force output needs to be processed by the coupling interface starting from the binary measurement files generated by the CFD simulation.

6. The Simpack® table reading method employs input *.afs* functions and always requires at least two force values - one for t_n and the other for t_{n+1} - to define an excitation-based force element spanning the whole duration of the MBD simulation.

The above aspects have a direct impact on the way the coupling interface has been developed, besides they also significantly affect the setup of both the PowerFLOW® and Simpack simulations. The explanation of such an impact is done in detail explained in two following sections for both the coupling between Simpack® and the Theodorsen routine and the coupling between PowerFLOW® and Simpack®.

5.1.3. FSI scheme

The FSI scheme employed to couple PowerFLOW® and Simpack® is the volume continuous version of the Conventional Sequential Staggered (CSS) scheme with subcycling of both the fluid and the multibody phases. This scheme is introduced in sec.2.1.2 in the context of a wider process of literature review on FSI coupling schemes and here its working mechanism is described more in detail. The CSS volume continuous scheme is shown in fig.5.2 and, for a generic coupling iteration n, it starts from the fluid solver

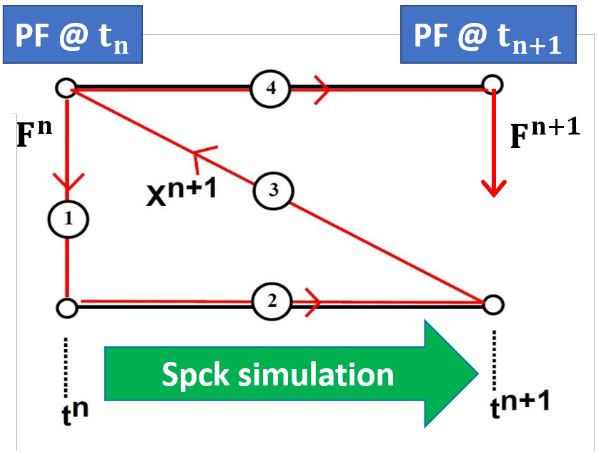


Figure 5.2: Volume continous CSS FSI scheme adopted to couple PowerFLOW® with Simpack®. Image taken from [8] and modified according to the needs.

(labeled as "PF" in the figure) which firstly provides the airload F^n at time t_n (step 1). Then Simpack® (labeled as "Spck" in the figure) runs a MBD simulation to compute the kinematic-states vector X^{n+1} at time t_{n+1} (step 2) by means of internal iterations and returns this to the fluid solver at time t_n (step 3). At this point, the aerodynamic tool advances up to t_{n+1} by performing a certain number of subcyces to return a new airload F^{n+1} (step 4). This approach assures the synchronization of the physical interface between the fluid and structural domains for the same coupling timestep, however the synchronization is not granted for the airload.

5.2. Simpack®-Theodorsen coupling demonstrator

In the process of developing the coupling between PowerFLOW® and Simpack®, an initial coupling between Simpack® and the convolution-integral formulation of the Theodorsen model is firstly implemented. This avoids some tricky aspects of the PowerFLOW® simulation and setup, such as those described in sec.5.4.1, which require special attention, might over-complicate the development of the methodology (in its early phases) by introducing further error sources and that can make the debugging process even harder. In this way, a self-implemented Theodorsen routine, which acts as surrogate of PowerFLOW®, helps in just focusing on the coupling Interface and its aspects/issues to achieve a working, robust and accurate method. Subsequently, the methodology is extended to PowerFLOW® by implementing additional features to the Interface to address the complexity of the aerodynamic solver on top of a verified framework. Therefore, this section covers the main aspects of the methodology for the Simpack®-Theodorsen coupling demonstrator.

5.2.1. Mechanism

As shown in fig.5.3, the coupling iteration process between Simpack® and the Theodorsen routine consists in the exchange of data by means of file input/output and the execution of commands between three main blocks. These blocks are the Theodorsen aerodynamic routine, the Simpack® MBD simulation and the Coupling Interface which links the two. The Theodorsen routine is a set of python routines acting as a surrogate of a CFD solver, at least from the external viewpoint, because it returns aerodynamic forces based on input kinematic states, as described in sec.5.2.4. Then, the Simpack® block features the MBD simulation and Post-processor, as discussed in sec.5.2.5. Finally, the Coupling Interface (see sec.5.2.3) is a collection of python function and classes specifically developed to process fluid and structure data and to generate the required input and output files for both simulation tools. The Interface is the tool that substantiates the coupling between the fluid and the structure domains of the aeroelastic simulation, and in this case, it is also the starting point of each coupling iteration. Finally, since the aerodynamic routine and structure solver communicate with each other by means of files, sec.5.2.2 is entirely dedicated to a detailed description of them.

The working principle of the coupling iteration mechanism can be described by employing the numbered solid or dashed arrows linking respectively files and software-component blocks in the diagram of

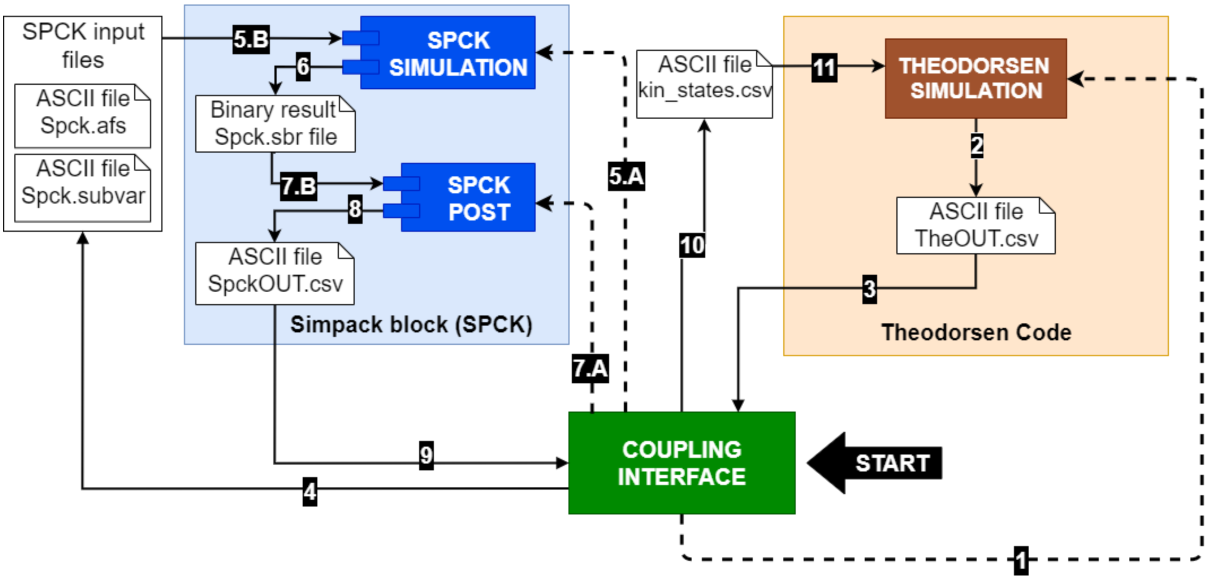


Figure 5.3: Diagram showing the working principle of the coupling between Simpack® and the Theodorsen routine in terms of software components and main files exchanged during each iteration.

fig.5.3. To begin with, the coupled aeroelastic simulation is started by a python script which controls the full process by looping though the prescribed number of coupling iterations and by calling the Coupling Interface for each iteration. Besides, this script specifies the number of coupling iterations and it takes care of tasks preliminary to the iterations, such as resetting those coupling files storing the time-histories of quantities of interest. Next, every coupling iteration features as step 1 the execution of the Theodorsen routine to compute the current airload from the kinematic states generated in the previous iteration. The aerodynamic output of the Theodorsen routine is "TheOUT.csv ASCII file (step 2) which is read by the Interface (step 3). Next the Interface generates the two required Simpack® input ASCII files, namely the "Spck.subvar" and the "Spck.afs" files (step 4). At this point, always the Interface launches a dedicated Simpack® script which sets-up and performs the MBD simulation (step 5.A) by reading the input "Spck.afs" and "Spck.subvar" files (step 5.B). The outcome of the Simpack® simulation is a "Spck.sbr" binary result file (step 6) which needs to be converted into an ASCII format to be processed. This is done by the Simpack® Post-Processor which is triggered by the Interface (step 7.A) and generates the ASCII "SpckOUT.csv" file (step 8). Finally, such "SpckOUT.csv" file is read and processed by the Interface (step 9) to generate the "kin_states.csv" input file (step 10) for the subsequent Theodorsen routine (step 11).

5.2.2. Files

As the chosen coupling medium is file input/output, a certain number of coupling files is employed to carry the data exchanged between Simpack® and the Theodorsen routine - these files are called "main coupling files" in tab.5.1 and can be visualized also in fig.5.3. On the other hand, other files are necessary to collect relevant simulation data required for the correct working of the overall process and to store the time-histories of the quantities of interest. These files are called "secondary coupling files" and are also shown in tab.5.1. In this subsection, focus is primarily on the function and type of data carried.

Theodorsen-Simpack® Coupling Files	
Main Files	Secondary Files
TheOUT.csv	Theodorsen_force_hist.csv
Spck.afs	iterator_list.txt
Spck.subvar	
Spck.sbr & SpckOUT.csv	
kin_states.csv	

Table 5.1: Main and secondary files employed for the coupling between the Theodorsen code and Simpack®.

Besides, also the main functions of the secondary files are shortly discussed. Regarding the structure and generation process of those coupling files shared also with the PowerFLOW®-Simpack® coupling (see tab.5.3), the reader can found an extensive description in the appendix sec.D.

Main coupling file: TheOUT.csv

This file contains the output of the Theodorsen aerodynamic routine and is generated during each coupling iteration. The main purpose of this file is to emulate the PowerFLOW® output measurement file by skipping the additional steps of the binary-to-ASCII file-conversion carried out by the PowerFLOW® CLI and the data processing operations required in the case of the CFD solver.

Main coupling file: Spck.afs

Regarding the Simpack® input files, two files are employed, with the "Spck.afs" one carrying the time history of the aerodynamic forces returned by the Theodorsen routine. Such file is very similar to the one used in the Simpack® verification activity and mentioned in sec.3.3. The difference is that now only two values for each aerodynamic force component are given: one at the beginning and one at the end of the Simpack® simulation spanning the single coupling timestep. The reason why Simpack® needs at least two force values when employing input functions to define excitation-based forces in the time domain is because the tool can only perform simulations of a given duration, say from an initial t_n to a final t_{n+1} time. Besides, another motivation is that two force values are needed by the *Time Integration solver* to perform interpolations during its internal iterations. Such two force values F_n at time $t_n = n\Delta_T$ (with Δ_T the coupling timestep) and F_{n+1} at time $t_{n+1} = t_n + \Delta_T = (n+1)\Delta_T$ related to each coupling iteration n can be obtained in the most simple case by assuming $F_{n+1} = F_n$, consistently with the explicit FSI scheme shown in fig.5.2 and discussed in sec.2.1.2.

Main coupling file: Spck.subvar

The second Simpack® input file is the "Spck.subvar" one and its purpose is to supply the MBD simulation with updated values for some simulation parameters. As every Simpack® simulation covers only one coupling iteration, the structural domain is not aware of the global status of the complete aeroelastic simulation. Therefore, such a fundamental piece of information needs to be provided, and the employment of *subvars* defined via the "Spck.subvar" file is the option here considered. As already explained in sec.3.3, *subvars* are objects generally used in Simpack® to parameterize the simulation setup. In this application, their main goal is to input the values of the kinematic states from the previous iteration and the global simulation time of the aeroelastic simulation. The kinematic states from the previous-iteration are employed as initial conditions and the global simulation time is used to model a prescribed motion spanning the whole aeroelastic simulation, when necessary. In this way, every Simpack® simulation is linked to the previous one emulating the behaviour of a co-simulation. Other quantities define *subvars* in this file, for instance the coupling timestep, the amplitude and frequency of the prescribed motion, when applicable. For these variables, their value does not change during the coupling and, alternatively, they could be internally defined in the "Spck.spck" simulation file, but they are included in the "Spck.subvar" file for code development and debugging reasons.

Main coupling files: Spck.sbr and SpckOUT.csv

The "Spck.sbr" binary result file is the main output file returned by Simpack® when a time integration is performed [65]. The settings of this file are prescribed in the *Solver Settings* tab of the Simpack® Pre GUI, where the precision (single) and the simulation data to write to the file (*joint states*) are chosen to minimize the file size/conversion time and to be consistent with the PowerFLOW® settings in ch.4, respectively. Given the fact that no information is available in the Simpack® User Guide [65] regarding the internal structure of such a "Spck.sbr" file, the employment of the Simpck Post-processor GUI for its conversion to an ASCII format is the most natural choice. This is done for every coupling iteration by loading both the post-processing file and script (see sec.5.2.5 for a definition of these aspects) into the Simpack® Post tool. In this way, the current kinematic states just returned by the Simpack® simulation can be selected from the plotted curves and finally exported to the "SpckOUT.csv" file by means of the *ASCII export* function[1]. A final aspect to mention, is that regardless of the adaptive-timestep LSODE solver employed, the binary result file exports the *joint states* every Δ_T, leading to a pair of values (one

[1] This functionality of Simpack® Post allows to select which data to export, how to structure them, their format and if a header is required, for more information the reader is referenced to [65].

for $t_n = 0$ and one for $t_{n+1} = \Delta_T$ in the time-frame of the Simpack® simulation) in the "SpckOUT.csv" file for the position, velocity and acceleration states of each DoF.

Main coupling files: kin_states.csv

This file serves as a data-container of the time-history of the kinematic states computed by the MBD simulation for the complete duration of the coupled simulation. This is firstly required to input the necessary kinematic states to the Theodorsen routine for the airload computation, according to the convolution integral formulation of the Theodorsen model. Next, the " kin_states.csv" file serves also as data container for the sake of collecting results for the subsequent post-processing.

Secondary coupling files

In tab.5.1, two secondary coupling files are shown, and their purpose is to either store the time-histories of quantities of interest - this is the case of the "Theodorsen_force_hist.csv" file - or to serve for the correct working mechanism of the coupling in the way it is conceived, and this is the case of the "iterator_list.txt" file. The "Theodorsen_force_hist.csv" file, contains the time-history of each component of the computed aerodynamic force by the Theodorsen routine for all the coupling iterations. The main reason for such file is to store data for the sake of verification with the reference solution, as discussed in sec.5.3. The other file, the "iterator_list.txt" one, is used to store the number of all the completed iterations by the aeroelastic coupled simulation. This file is read as one of the very first operations carried out by the Coupling Interface, which is the topic of the next section.

5.2.3. Interface

As already mentioned, the Coupling Interface consists of several python scripts, functions and classes organized in such a way that, at the top, there is a main function calling all the other developed sub-functions, each one related to a specific step or task of the coupling iteration mechanism. The main function of Interface developed for this demonstrator Simpack®-Theodorsen coupling requires a few input quantities to carry out its internal operations. These are the coupling parameters provided by the script controlling the full coupling simulation, the paths to the coupling files and the number of the current iteration from "iterator_list.txt". Once these preliminary operations are carried out, the Coupling Interface can perform all the tasks previously described in sec.5.2.1 and represented in the diagram of fig.5.3. In addition to these tasks, to carry out step 5.A and step 7.A, the two JavaScripts related to the Simpack® simulation-setup and time-integration, as well as post-processing, are executed. Besides, before the last step 11 in fig.5.3 and a new coupling iteration, the "iterator_list.txt" and the "Theodorsen_force_hist.csv" data-containers file are updated based on the current airload value and the number of the subsequent iteration. Finally, if the reader is interested in all the details of the Coupling Interface, the appendix ch.D contains an extensive description with respect to the PowerFLOW®-Simpack® coupling which is developed by extending the Interface related to this demonstrator.

5.2.4. Theodorsen routine as CFD surrogate

The Theodorsen routine, as already explained, can be considered as an emulator of a more complex CFD code, like PowerFLOW®, allowing to focus on the fundamental aspects of the coupling, namely the correct transfer of fluid and structure data, as well as the synchronised execution of the tools. At the same time, the Theodorsen model is sophisticated enough to provide aerodynamic forces in good agreement with those computed by a CFD solver under attached flow conditions, where incompressible and viscous flow phenomena can be neglected, as shown in ch.4. The Theodorsen code employed in this demonstrator coupling consists in the time-domain convolution-integral formulation of the model, as described in sec.3.1.3 and sec.B.2 for a 2 DoFs aeroelastic system. In addition, also a 1 DoF (free plunge) version of the previous problem is derived for a system with prescribed pitch motion, as shown in sec.5.3.1. Both the 2 DoFs and 1 DoF problems are implemented in the Theodorsen routine, and they are exploited as two separate test cases for the verification of the coupling demonstrator aeroelastic solutions against the same returned by the full analytical solutions of the same problems without coupling. In this section, the general working of the code is discussed, while the derivations of respective mathematical models for the two problems can be found in the previously mentioned sections of this report.

First of all, when considering the complete coupling mechanism shown in fig.5.3, this Theodorsen code can be considered as a "black-box" which returns an instantaneous aerodynamic force value $F(t_n)$ for a given time-history of kinematic states x, $\dot{x}$ and $\ddot{x}$ up to the coupling time step t_n. Regarding the internal working of this code, two sets of tasks are carried out by a few python routines depending on the coupling iteration number. In the case of the first coupling iteration, given the fact that only the position and velocity states are known for $t_0 = 0$ (initial conditions), it is necessary to compute also the acceleration states by solving the full aeroelastic systems of equations. In the case of the 1 DoF coupling, this requires also the loading of the pitch prescribed motion derivatives at t_0. After this, the Theodorsen code computes the lift airload F_0 for the 1 DoF problem and both the lift and aerodynamic moment for the 2 DoFs problem according to the respective formulations. When it comes to a generic coupling iteration n, the complete time-history of the Simpack® kinematic states $x(t_n)$, $\dot{x}(t_n)$ and $\ddot{x}(t_n)$ - where a generic x refers to only the plunge state (h) for the 1 DoF problem and to both the plunge and pitch (α) states for the 2 DoFs problem - is firstly imported from the "kin_states.csv" file due to the convolution integral formulation. Besides, for the 1 DoF problem, also the full time-histories of the pitch motion derivatives are required always because of the convolution integral. Next, the airload F_n at t_n is computed and returned to the Coupling Interface.

5.2.5. Simpack® simulation and post-processing

Although the Simpack® simulation setups for both the 1 and the 2 DoFs problems are rather simple and can be entirely automatized in all their aspects, in order to reduce the time required by each coupling iteration - this might lead to a significant reduction in the total time for the coupled aeroelastic simulation given the large number of iterations - only a few aspects are handled by the Simpack® simulation script. This means that most of the Simpack® simulation setup (those aspects that do not change during the simulation) is manually carried out only once for each problem investigated according to its formulation, as described in sec.5.3. The results of this manual setup is saved to the "Spck.spck" simulation file which is executed by the Coupling Interface at each iteration. Therefore, the script takes care only of those parameters changing during the iterations (initial conditions) and runs the time integration, as discussed in the appendix sec.D.4.3 under the "SpckSimManager.sjs" paragraph. Regarding the Simpack® post-processing, its purpose is to load the kinematic results contained in the "Spck.sbr" binary file, extract them and write such data to the ASCII "SpckOUT.csv" file. Also in this case, the same approach employed before in the setup of the MBD simulation is here applied. Therefore, a preliminary setup of the post-processing file ("Spck.spf") is manually carried out only once and for each problem simulated, where the layout of the time-plots with the kinematic states to export is prepared. After this, the generation process of the "SpckOUT.csv" file is carried out by the Simpack® post-processing script, as described in sec.5.2.2.

5.3. Verification of the coupling demonstrator

5.3.1. First test case: 1 DoF problem

The first test case consists in an airfoil undergoing a dimensional free plunge motion h_O - but from now on represented by the h symbol for simplicity - under the action of elastic, inertial and aerodynamic lift forces, as well as a prescribed harmonic pitch motion $\alpha(t)$. Similarly to the problem considered in ch.3, the elastic vertical force is modelled by lumped parameters (a linear translation spring) without including structural damping. The vertical lift force is provided by the Theodorsen routine every coupling time step and is applied to the quarter chord aerodynamic center, which is also the pitching axis location. The drawing of the aeroelastic airfoil system is shown in fig.5.4.

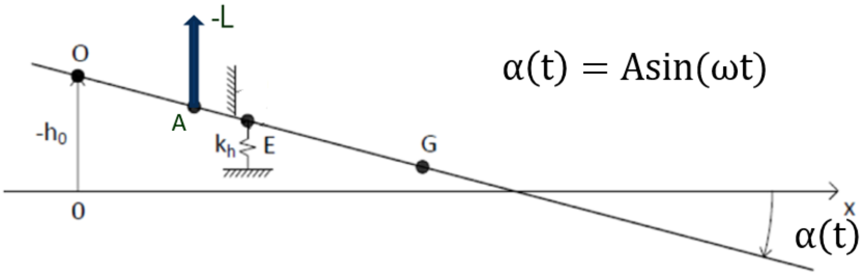


Figure 5.4: Drawing of the 1 DoF aeroelastic airfoil problem with prescribed pitch and free plunge motion employed for the Simpack®-Theodorsen coupling demonstrator.

Mathematical model

To derive the mathematical formulation for this problem, the same approach used to derive the EoMs in sec.3.1.2 for the 2 DoFs problem of the Simpack® verification activity is followed. Given the fact that the pitch motion is prescribed, the problem can be modelled by a single ODE for the plunge DoF, leading to the following EoM, where the lift airload can be split into a non-circulatory L_{nc} and a circulatory component L_c according to the Theodorsen model.

$$m\ddot{h} + k_h h = -mx_G\ddot{\alpha}(t) - k_h x_E \alpha(t) - L \qquad \text{with} \qquad L = L_{nc} + L_c \tag{5.1}$$

For L_{nc}, the following expression is employed, which is exactly the same as the one employed in sec.3.1.3, with the only difference that now only $\ddot{h}$ is an unknown quantity for this problem.

$$L_{nc} = \pi\rho b^2 \Big(\ddot{h} + \frac{b}{2}\ddot{\alpha}(t) + U_\infty \dot{\alpha}(t)\Big) \tag{5.2}$$

In the case of L_c, this can be further split according to the convolution-integral formulation derived in sec.3.2.1 and in the appendix sec.B.2 into two components L_{c_0} and CI. The first one L_{c_0} contains the Wagner function evaluated at a generic time instant $W(t)$, which multiplies the structural input evaluated instead at $t=0$. Regarding the second component CI, this is the true convolution integral between the delayed Wagner function $W(t-\tau)$ and the time derivative of the structural input function.

$$L_c = 2\pi\rho U_\infty b \Big(L_{c_0} + CI\Big) \qquad \text{with} \quad L_{c_0} = \Big[\dot{h}|_{t=0} + b\dot{\alpha}(t=0) + U_\infty \alpha(t=0)\Big] W(t)$$

$$\text{and with} \quad CI = \int_0^t W(t-\tau)\Big[\ddot{h} + b\ddot{\alpha}(\tau) + U_\infty \dot{\alpha}(\tau)\Big]\, d\tau \tag{5.3}$$

Given all the above equations, the following state-space-model of the aeroelastic system can be obtained based on the $y = \dot{h}$ assumption. In the derivation, all the h-terms in the non-circulatory lift are moved to the lhs, while all $\alpha(t)$-terms are known, and therefore kept on the rhs.

$$\begin{bmatrix} m-\pi\rho b^2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{y} \\ \dot{h} \end{bmatrix} + \begin{bmatrix} 0 & k_h \\ -1 & 0 \end{bmatrix} \begin{bmatrix} y \\ h \end{bmatrix} = \boldsymbol{f} \qquad \rightarrow \qquad \boldsymbol{\mathcal{M}}\dot{\boldsymbol{z}} + \boldsymbol{\mathcal{K}}\boldsymbol{z} = \boldsymbol{f}(t) \tag{5.4}$$

The above state-space-model of the problem leads to a combined mass-aerodynamic inertial matrix $\boldsymbol{\mathcal{M}}$ and a pure structural matrix $\boldsymbol{\mathcal{K}}$. Regarding the lhs term $\boldsymbol{f}(t)$, its complete expression in terms of the quantities used in the previous equations is reported below.

$$\boldsymbol{f}(t) = \Big[-L_{nc_\alpha} - p - L_c; 0\Big]^T \quad \text{with} \quad L_{nc}^{(\alpha)} = \pi\rho b^2 \Big(\frac{b}{2}\ddot{\alpha}(t) + U_\infty\dot{\alpha}(t)\Big) \quad \text{and with} \quad p = mx_G\ddot{\alpha}(t) + k_h x_E \alpha(t)$$

By applying the same approach employed in sec.3.1.3, the analytical solution $\dot{\boldsymbol{z}}$ is obtained by inverting matrix $\boldsymbol{\mathcal{M}}$.

$$\boldsymbol{\mathcal{M}}\dot{\boldsymbol{z}} + \boldsymbol{\mathcal{K}}\boldsymbol{z} = \boldsymbol{f}(t) \quad \rightarrow \quad \dot{\boldsymbol{z}} = -\boldsymbol{\mathcal{M}}^{-1}\boldsymbol{\mathcal{K}}\boldsymbol{z} + \boldsymbol{\mathcal{M}}^{-1}\boldsymbol{f}(t) = \boldsymbol{\mathcal{A}} + \boldsymbol{b}(t) \tag{5.5}$$

The numerical solution of this state-space problem, can be obtained by applying the forward Euler method to eq.5.5, similarly to what done in sec.3.2.1. This leads to the right expression below for the computation of the acceleration states. This expression is employed in the Theodorsen code at the very first iteration, as discussed in sec.5.2.4. On the other hand, the left expression below is used to get a benchmark solution for the reference kinematic states shown in the results part of this section.

$$\boldsymbol{z}_{n+1} = \Big(\boldsymbol{\mathcal{A}}\Delta t + \boldsymbol{I}\Big)\boldsymbol{z}_n + \boldsymbol{b}_n \Delta t \qquad \text{and} \qquad \dot{\boldsymbol{z}}_n = \boldsymbol{\mathcal{A}} + \boldsymbol{b}_n \tag{5.6}$$

Finally, regarding the numerical discretization of the quantity $\boldsymbol{b}_n$, the same approach employed for the Simpack® verification activity on the 2 DoFs airfoil is employed. Therefore, the reader is referenced to sec.3.2.1 and sec.B.3.

Simpack® setup

The setup of the Simpack® simulation is similar to the one described in sec.3.3, with major differences only in the *Joint* and *Time Integration Solver*. Regarding the *Joint*, to model the combined free and prescribed motions of the airfoil considered in this problem, the *6 DOF al-be-ga Joint type* is employed. This is a very general *Joint type* that allows to select which of the six DoF is characterized by a free motion and which instead by a prescribed/driven one. Along the driven directions, the user can specify an *Excitation*, besides it also possible not to provide any excitation and, in this case, Simpack® treats that motion as locked. In the reference frame of this problem[2], the only driven direction with a non-zero *Excitation* modelling the harmonic pitch motion is the rotation about the z-axis (ga rotation), while the y-direction is set as not-driven, because it is related to the only true DoF of the system. For the initial conditions in the plunge position and velocity, the corresponding *subvars* are applied to load the values for h and $\dot{h}$ from the previous coupling iteration. The *Excitation* modelling the harmonic pitch rotation is defined based on three *Expressions* for the $\alpha(t)$, $\dot{\alpha}(t)$, $\ddot{\alpha}(t)$ states defined below:

$$\alpha(t) = A\sin(\omega(t+T)); \qquad \dot{\alpha}(t) = A\omega\cos(\omega(t+T)) \qquad \ddot{\alpha}(t) = -A\omega^2\sin(\omega(t+T)) \tag{5.7}$$

where the term T is the time of the global aeroelastic simulation. The latter parameter is imported as a *subvar* provided by the "Spck.subvar" input file, while t is the time of the MBD simulation spanning the coupling iteration interval $[0 : \Delta_T]$. Finally, regarding the *Time Integration Solver*, as already introduced in sec.5.1.2, the adaptive time-step LSODE solver is selected with its main properties more in detail described in sec.2.3.2. As mentioned, in sec.5.1.2, the employment of an adaptive timestep integrator in Simpack® determines subcyling also for the structural phase of the aeroelastic simulation, with a structure subcycling factor in the order of 10 and 1 for the values of the coupling timestep considered.

Test conditions

The 1 DoF problem defined before is tested for the simulation parameters shown in tab.5.2. The coupled aeroelastic simulation considered has a duration of 1s, a time frame sufficient to reach an expected convergence of the aeroelastic solution (kinematic states and lift force) to a periodic solution, due to the prescribed harmonic plunge forcing. Both the reference and the coupling solution for this test problem are computed for 3 different values of the coupling timestep, namely $\Delta_T = 0.01s$, $\Delta_T = 0.001s$ and $\Delta_T = 0.0001s$ corresponding to 20, 200 and 2000 points per period of pitch motion respectively. Flow conditions compatible with the assumptions on which the Theodorsen model is based are employed, leading to a Reynolds number of 6.8 million and a Mach number of 0.294 (air temperature considered 15° C). Regarding the pitching motion, a small amplitude value of 1.5° is considered to ensure that the attached flow assumption used in Theodorsen model can be still considered as valid, and a pitch frequency of 5 Hz is employed to have enough oscillations (5 are expected) within the simulation duration.

Name	Symbol	Value	Unit	Subvar
airfoil semi-chord	b	0.5	[m]	$_b
airfoil depth	d	1.0	[m]	$_depth
point E x-coordinate with respect to AC	x_E	-0.05	[m]	$_xe
point G x-coordinate with respect to AC	x_G	0.25	[m]	$_xg
angular velocity of prescribed pitch	ω_α	$2\pi5$	[rad/s]	$_om_a
amplitude of prescribed pitch	A	1.5	[deg]	$_amp_a
flow density	ρ	1.225	[kg/m^3]	-
lift aerodynamic derivative	c_{l_α}	2π	[-]	-
flow velocity	U_∞	100.0	[m/s]	-
airfoil mass	m	20.0	[kg]	$_m
translation stiffness	k_h	1900.0	[N/m]	$_kh

Table 5.2: Geometric, aerodynamic and structure parameters employed for the Simpack®-Theodorsen coupling demonstrator problem involving a 1 DoF aeroelastic airfoil with free plunge and prescribed harmonic pitch motions.

[2] In Simpack®, the airfoil is still laying on the x-y plane, but the the positive y-direction is directed upwards and the z-axis is point outward the x-y plane, differently from the reference solution in fig.5.4.

Regarding the structural parameters, the values of the airfoil mass and of the translation spring stiffness return a frequency value for the plunge motion - assuming this to be fully decoupled from the pitch motion - of 1.55 Hz. In addition to the periodic pitch forcing, the system is perturbed by an initial plunge velocity perturbation of $\dot{h} = 0.01$ m/s.

Results

In this paragraph, the results of the coupling demonstrator for the first test problem are compared against the reference numerical solution of the analytical model without coupling. Both solutions are obtained from the same physical and mathematical formulation of the problem at stake. Besides, ch.3 showed that the tuning effect of the Simpack® MBD simulation parameters can be considered as negligible once an optimal setup is found. This means that the only sources of error - here the word error is more appropriate than mismatch due to the previous reasons - in this comparison could be related to the numerical discretization (fixed timestep forward Euler vs the LSODE adaptive timestep method), which are although expected to be very small, and the coupling timestep. The latter is very likely to be the main source of error, even though for decreasing values of the coupling timestep, it is should have a smaller effect from this parameter, leading to an expected perfect match between the two solutions in the limit of $\Delta_T \rightarrow 0$. Therefore, the main goal of this investigation is to assess the effect of this fundamental parameter of explicit FSI coupling algorithms on the aeroelastic variables for the three previously mentioned values of the coupling timestep. Being said this, the results of this test problem are shown in fig.5.5 for the plunge position and in fig.5.6 for the lift coefficient, while the plunge velocity and acceleration are respectively shown in fig.F.4 and fig.F.5 of the appendix. From these plots, it is apparent that the methodology is able to replicate the numerical reference solution of this aeroelastic problem with a very high level of accuracy, both in the kinematic states and in the lift signal. This is also achieved for the largest coupling timestep considered of 20 samples per period of pitch motion ($\Delta_T = 0.01$s), apart from small amplitude discrepancies in the initial transient of the response (especially in the case of the position state). However, these errors are substantially removed by reducing the

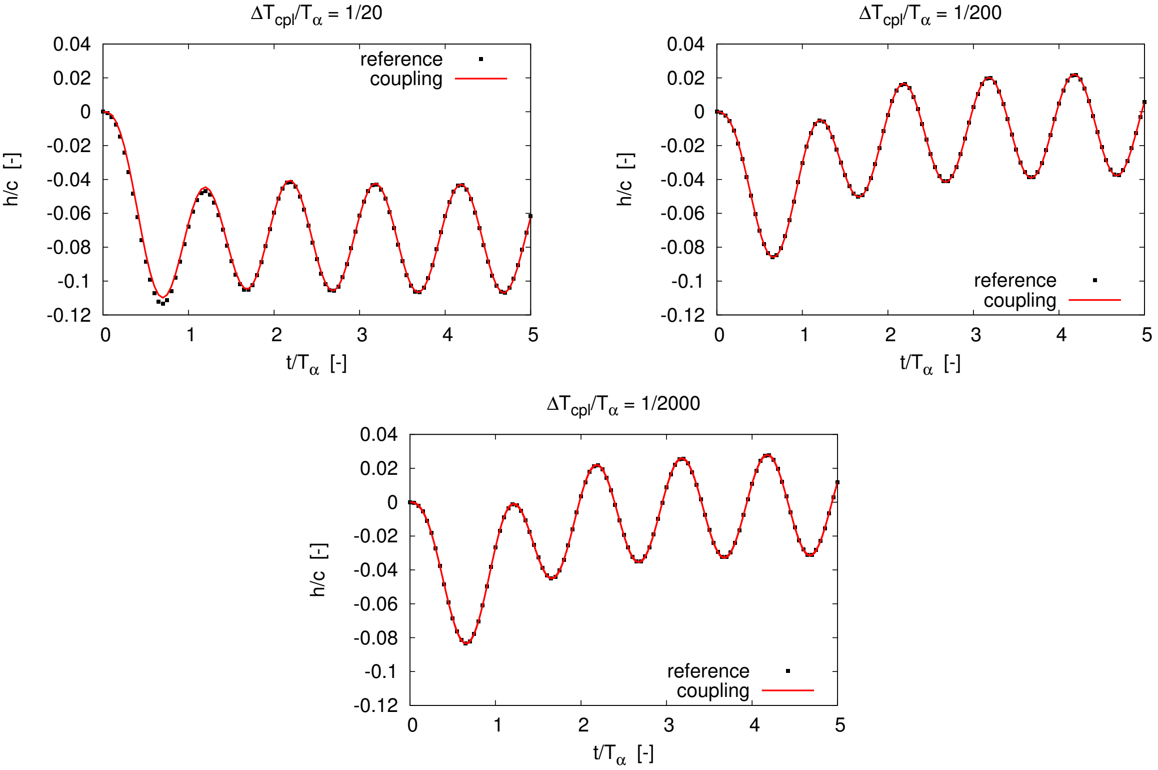


Figure 5.5: Time-histories of the chord-scaled ($c = 1$m) plunge positions h for the 1 DoF version of the Simpack®-Theodorsen coupling (red lines) compared with the reference aeroelastic numerical solution (black squares) under three values of the coupling time-step (top left $\Delta_T = 0.01s$, top right $\Delta_T = 0.001s$, bottom $\Delta_T = 0.0001s$) with $T_\alpha = 0.2$s.

coupling timestep already from 20 to 200 fractions of the pitch period, due to the underlying explicit FSI method employed. Moreover, it is interesting to observe that both the reference and the coupling solutions for the largest timestep are different from the ones calculated for the other timesteps, when considering the same variable of interest. Such phenomenon is rather apparent in the case of the plunge position state in fig.5.5. This is due to the fact that the underlying forward Euler method has not yet converged to a solution due to its low first-order global discetization error decrease rate. However, the goal of this investigation is to make sure that the developed coupling methodology is able to provide an accurate aeroelastic solution based on a given reference solution, and this is here achieved regardless of the convergence of the numerical solution. Regarding the appearance of the computed solution, it is possible to see that after the first two or three periods of pitch motion, all the kinematic states converge to a periodic solution fed by the harmonic pitch motion. The lift signal is the fastest to reach a periodic solution, most likely because the lift force in fig.5.6 is more influenced by the pitch motion than by the plunge one. On the other hand, the plunge position is the slowest to reach a periodic behaviour, due to an initial transient dominated by the elastic forces, where oscillations have also a larger time-scale than those of the periodic part of the solution.

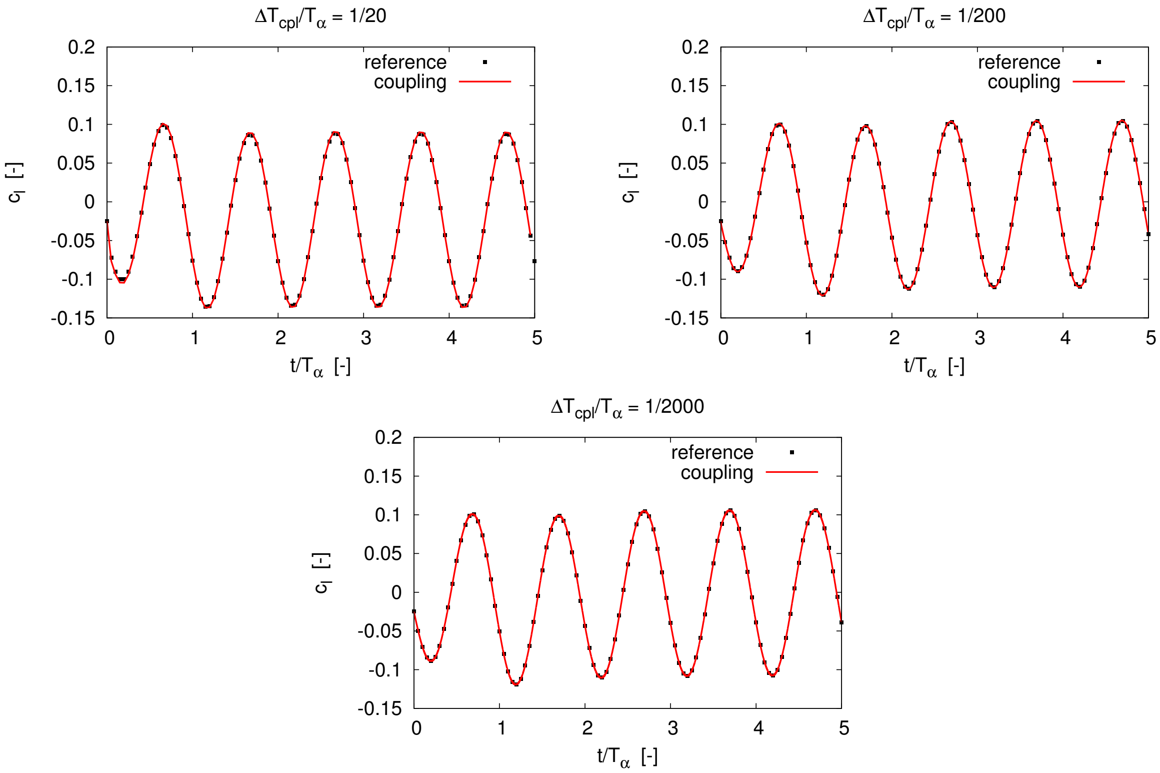


Figure 5.6: Time-histories of the lift coefficient c_l (scaling factors chord $c = 1$m, freestream velocity $U_\infty = 100$m/s and density $\rho = 1.225$kg/m^3) for the 1 DoF version of the Simpack®-Theodorsen coupling (red lines) compared with the reference aeroelastic numerical solution (black squares) under three values of the coupling time-step (top left $\Delta_T = 0.01s$, top right $\Delta_T = 0.001s$, bottom $\Delta_T = 0.0001s$) with $T_\alpha = 0.2$s.

Conclusion

With this investigation, it is possible to state that the coupling methodology developed works in a very accurate fashion for this problem. Furthermore, a coupling timestep between 20 and 200 samples per period of pitch motion - leading to 100 to 1000 coupling iterations for 1s aeroelastic simulations - can be a reasonable choice to have an accurate prediction of aeroelastic variables with a moderate number of iterations, and this is an essential indication for the 1 DoF coupling between PowerFLOW® and Simpack®. To mention a few aspects concerning performance, a significant portion of the time it takes for each coupling iteration is spent in the binary-to-ASCII conversion of the Simpack® results file. This bottleneck is solved for the coupling between PowerFLOW® and Simpack® by proposing a different approach, which improves both the performance and efficiency of the methodology.

5.3.2. Second test case: 2 DoFs problem

The coupling demonstrator is tested also on a second test problem consisting in a 2 DoFs airfoil with free pitch and plunge motions and where the aerdynamic forces are computed from the convolution integral formulation of the Theodorsen model. This is exactly the same problem employed in ch.3 in the case of the Theodorsen model in its convolution integral formulation for the verification activity of the Simpack® MBD setup. The difference is that, in this case, the kinematic states are obtained from a timestep-per-timestep two-way coupling loop with an aerodynamic routine, while in ch.3 the entire force time-history (computed externally beforehand) is supplied to Simpack® at the beginning of the simulation (just for the sake of assessing the structural part of the modeling). Therefore, the definition of the problem can be found in sec.3.1.1, while its analytical formulation is in sec.3.1.2, sec.3.1.3 and sec.B.2. Then, the discretization of the EoMs can be found in sec.3.2, while the expressions for the aerodynamic forces computed by the Theodorsen code for this problem are shown in the appendix sec.B.5.1 and when it comes to the Simpack® setup, this is the same as the one described in sec.3.3. Finally, concerning the testing conditions, also for this 2 DoFs problem the same airfoil-related quantities (chord, depth, *Markers* locations) and structural parameters (mass, translation spring stiffness) of the 1 DoF problem in tab.5.2 are used. What differs is the free-stream velocity, which is lowered to $U_\infty = 52\text{m/s}$, although the same density and aerodynamic derivative c_{l_α} are kept. Besides, this 2 DoFs problem employs a rotation spring with stiffness ($k_\alpha = 2600\text{Nm/rad}$) and an inertia moment about the CoG ($J_G = 1.3\text{kg}\cdot\text{m}^2$), while the same initial condition (plunge velocity perturbation) is employed to trigger the response of this system.

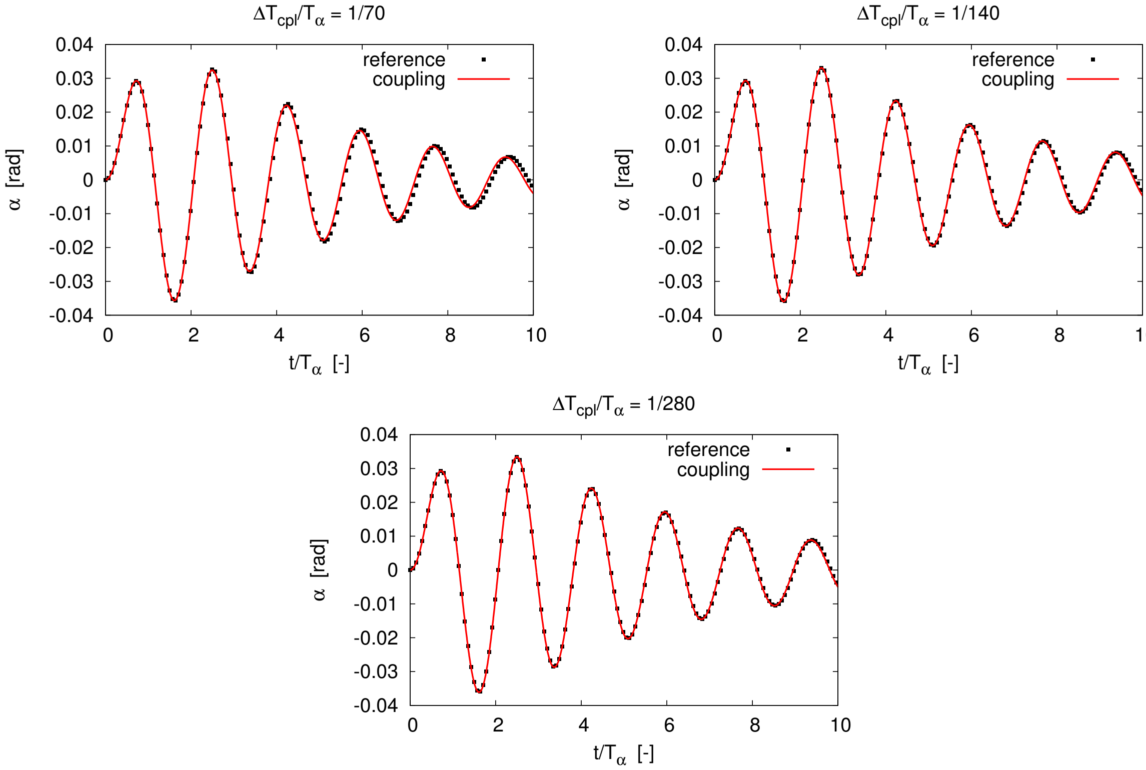


Figure 5.7: Time-histories of the pitch positions α for the 2 DoFs version of the Simpack®-Theodorsen coupling (red line) against the reference analytical model (black squares) under three values of the coupling time-step (top left $\Delta_T = 0.002s$, top right $\Delta_T = 0.001s$, bottom $\Delta_T = 0.0005s$) with $T_\alpha = 0.1405$s.

Also in this case, an almost perfect match between the two solutions should be obtained, once the influence of the coupling timestep can be considered as negligible. Therefore, when the coupling timestep is still large compared to a characteristic period, this quantity can be considered as the main source of error, given the same physical and mathematical modelling shared by the two solutions, while the influence of discretization and tuning of the Simpack® setup can be considered as minor or even negligible sources of error, as the results of sec.3.4 and sec.5.3.1 confirm. Being said this, the results of this verification are shown in fig.5.7 for the pitch position, in fig.5.8 for the plunge position, in fig.5.9 for the lift coefficient and in fig.5.10 for the moment one, while the pitch/plunge velocities and positions can be visualized in fig.F.6, fig.F.7, fig.F.8 and fig.F.9 of the appendix. All these figures show the previously mentioned quantities for three values of the coupling timestep, namely 70, 140 and 280 fractions of pitch period $T_\alpha = \sqrt{k_\alpha/J_G} = 0.1405$s. The main observation that can be made from such results is that the main effect of reducing the coupling timestep is a reduction of the phase error affecting the last portion of the response between the reference analytical solution and the coupling one. This is quite different with respect to the previous 1 DoF problem, where the effect is related to the amplitude error. For this 2 DoFs problem, already a timestep 70 times smaller than the pitch period is able to accurately capture in amplitude the values reached by the oscillations of the response. On the other hand, to also have an equally accurate match in phase, a four-times smaller timestep is necessary for this type of problem. This is quite close to what observed in the 1 DoF test problem, where a timestep of 20-200 fractions of T_α is enough. However, the additional complexity introduced by the interaction between the free plunge and pitch motions suggests a more conservative choice around 200-300 fractions to employ in the 2DoF coupling between PowerFLOW® and Simpack®.

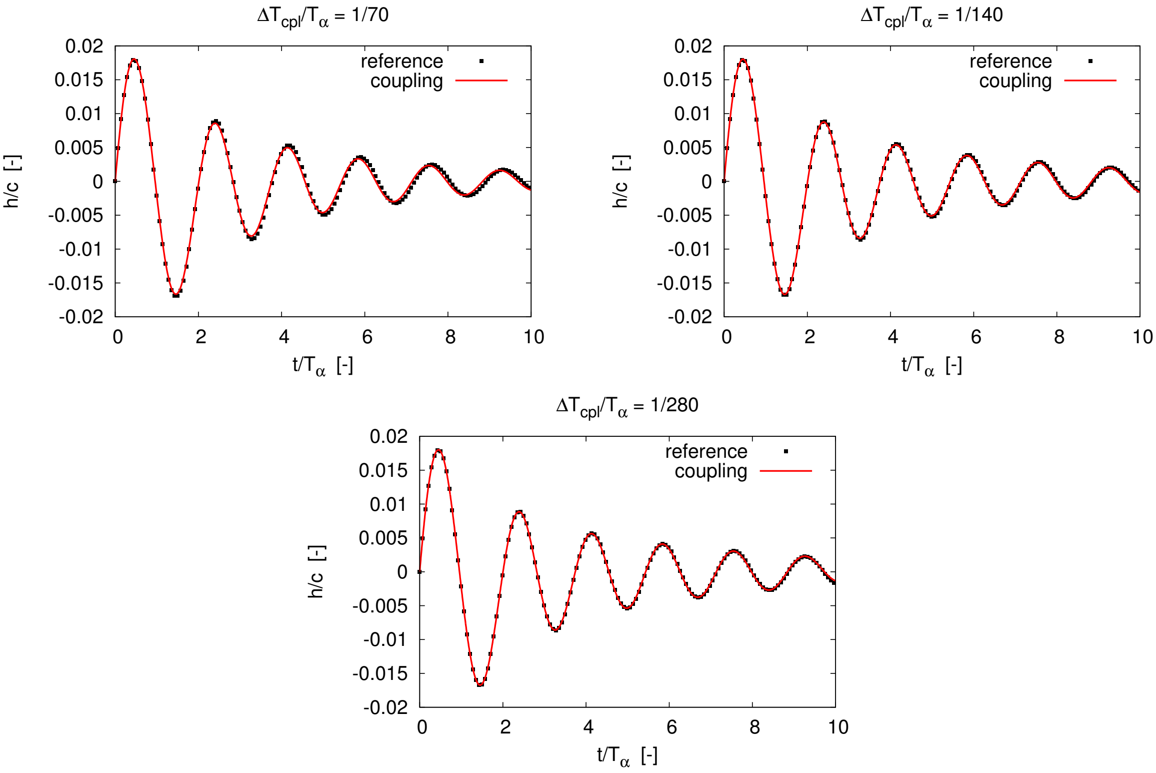


Figure 5.8: Time-histories of the chord-scaled (c = 1m) plunge positions h for the 2 DoFs version of the Simpack®-Theodorsen coupling (red line) against the reference analytical model (black squares) under three values of the coupling time-step (top left $\Delta_T = 0.002s$, top right $\Delta_T = 0.001s$, bottom $\Delta_T = 0.0005s$) with $T_\alpha = 0.1405$s.

When the physics of the aeroelastic response simulated is considered, this system features a similar behaviour to the one in sec.3.4 for the Theodorsen airload, as both the kinematic states and the aerodynamic loads oscillations are significantly damped within the 10 T_α oscillations periods considered, after an initial transient of a couple of periods. This type of response is rather different from the one obtained by the 1 DoF problem, where the system reaches a periodic solution due to the harmonic pitch forcing term. In this 2 DoFs system, the response observed can be considered as the response to a plunge impulse, which triggers the internal dynamics of the plunge motion more than the pitch one, at least at the beginning of the response. This is related also to the about 5 or 6 oscillations of the aeroelastic variables observed for the simulation duration considered, with a characteristic time per oscillation that is larger in the initial transient and closer to the characteristic plunge period $T_h = \sqrt{k_h/m} = 0.64$s.

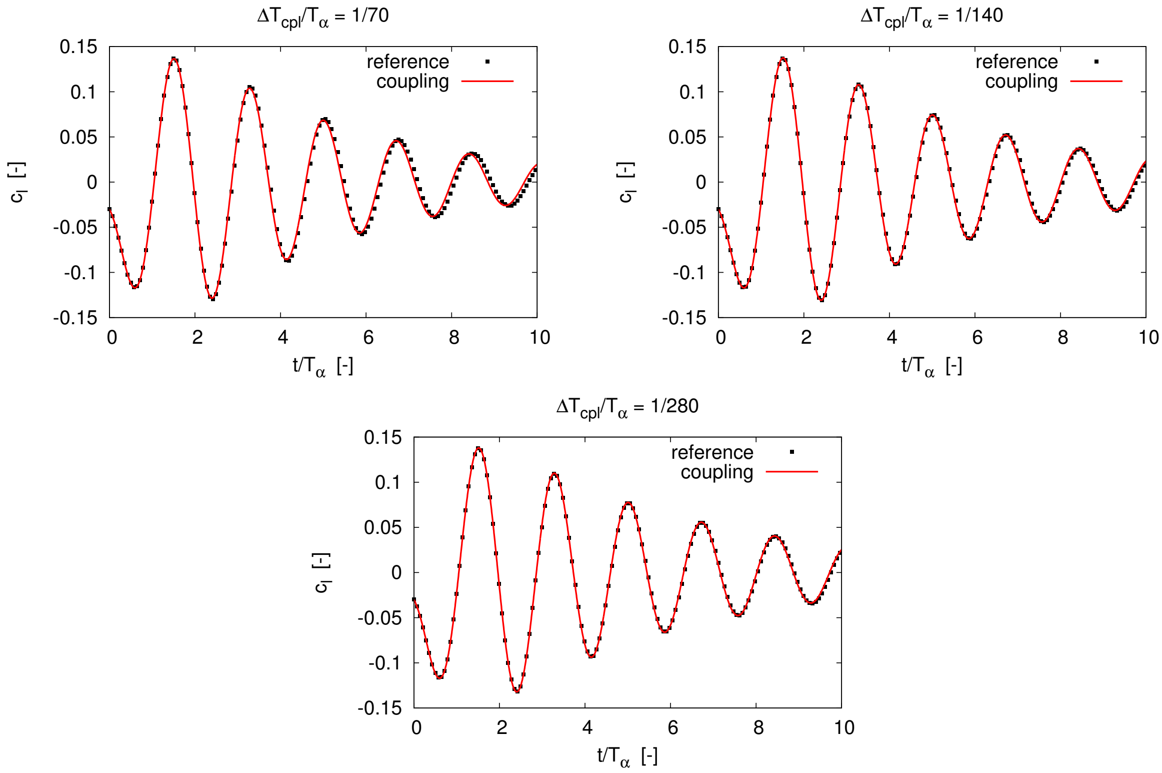


Figure 5.9: Time-histories of the lift coefficients c_l (scaling parameters $c = 1$m, $U_\infty = 52$m/s and $\rho = 1.225$kg$/m^3$) for the 2 DoFs version of the Simpack®-Theodorsen coupling (red line) against the reference analytical model (black squares) under three values of the coupling time-step (top left $\Delta_T = 0.002s$, top right $\Delta_T = 0.001s$, bottom $\Delta_T = 0.0005s$) with $T_\alpha = 0.1405$s.

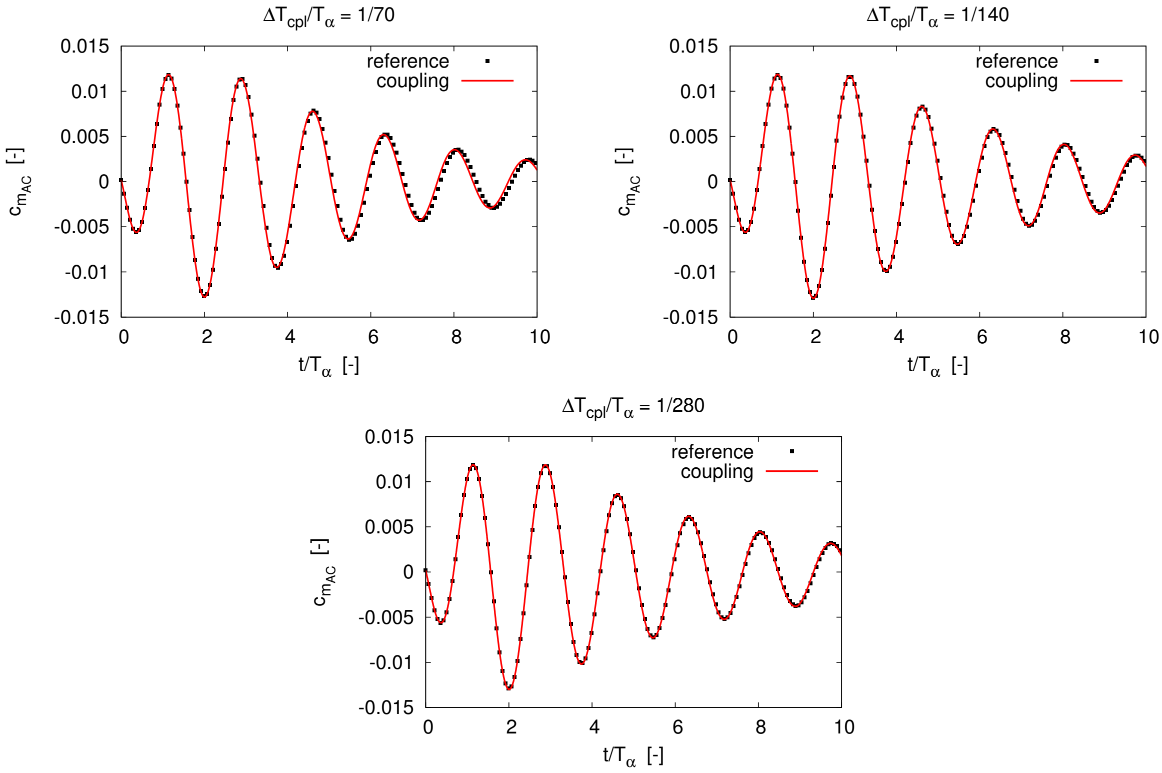


Figure 5.10: Time-histories of the quarter-chord aerodynamic moment coefficients $c_{m_{AC}}$ (scaling parameters $c = 1$m, $U_\infty = 52$m/s and $\rho = 1.225$kg$/m^3$) for the 2 DoFs version of the Simpack®-Theodorsen coupling (red line) against the reference analytical model (black squares) under three values of the coupling time-step (top left $\Delta_T = 0.002s$, top right $\Delta_T = 0.001s$, bottom $\Delta_T = 0.0005s$) with $T_\alpha = 0.1405$s.

5.4. PowerFLOW®-Simpack® coupling

This final section of the methodology chapter, focuses only on the main aspects of the coupling between PowerFLOW® and Simpack®, while the results of this method are shown in ch.6 for the same 1 DoF and the 2 DoFs aeroelastic problems employed in the verification activity of the Simpack®-Theodorsen coupling demonstrator.

5.4.1. Relevant coupling aspects and differences due to PowerFLOW®

There are several aspects differentiating the CFD tool with respect to the Theodorsen routine for the sake of the coupling and the most relevant ones are outlined in the following paragraphs.

Initial transient of the CFD simulation

Contrarily to the Theodorsen code, the PowerFLOW® simulation (as all CFD simulations) is affected by an unavoidable initial transient due to the adaptation of the fluid-flow to the boundary conditions - even when steady-state solutions are expected - where large spurious airload oscillations can occur. For this reason, the force value of the very first coupling iteration F_0 can not be immediately extracted and provided to Simpack®, as it might significantly affect the computation of the kinematic states, or even the stability of the coupled aeroelastic solution. Therefore, the standalone PowerFLOW® simulation needs to continue until the initial transient has ceased. Such an initial transient of the CFD simulation leads to the splitting of the Coupling Interface into a first mode related to the handling of the initial transient and one entirely devoted to the coupling with Simpack®. Besides, a method to establish the conclusion of the initial transient is introduced, which is based on the convergence of the root-mean-square (rms) of a scaled running-average of the force signal(s) to a user-defined tolerance, where the computation of the running average requires a window size, whose value is the flow-pass time c/U_∞. After extracting each force signal from the PowerFLOW® output file, a dedicated library - called "InitTransChecker" and described in the appendix sec.D.4.2 - computes the scaled rms of the running-averaged force signal as:

$$F^*_{rms} = \frac{1}{\bar{F}}\sqrt{\frac{\sum_{i=1}^{N}(F_i - \bar{F}\,)^2}{N-1}} \qquad \text{with} \qquad N = \text{ceil}\left(\frac{c/U_\infty}{\Delta_T}\right) \tag{5.8}$$

where the window size (N) is the ceiling (integer rounding towards plus infinity) of the ratio between the flow-pass time and the coupling timestep Δ_T, corresponding to a given number of coupling timesteps/iterations; then F_i is the i-th value of the force signal considered within the window-size and $\bar{F}$ is the average of the force signal within the moving window. The scaling parameter employed is the running-averaged force signal $\bar{F}$ itself, which makes this method applicable if a converged value of the force signal different from zero is expected[3]. Regarding the convergence criteria, the conclusion of the PowerFLOW® initial transient occurs when at least one of the force signals satisfies the condition $F^*_{rms} < \epsilon_F$, which can also be extended to a set of signals or even all of them. The value of the force tolerance ϵ_f and the choice of considering one or more force signals for the convergence creterion is a trade-off between computation time and accuracy, as well as depending also on the initial conditions set. This aspect is discussed in ch.6 based on the actual initial conditions employed, as well as their impact on the accuracy of F_0 and of the overall aeroelastic solution. Finally, the splitting of the coupling into an initial transient phase and a coupling phase, suggests the employment of two counters to distinguish the proper coupling iterations "cpl_it" from the total ones "tot_it" (initial transient plus proper coupling). The relation between the two is that the "tot_it" corresponding to the end of the PowerFLOW® initial transient corresponds to cpl_it=0 as very first coupling iteration. In this way, the same settings employed for the coupling demonstrator can be easily translated to the PowerFLOW®-Simpack® coupling, while the first value of the "tot_it" counter is tot_it=1 for reasons related to the numbering of the PowerFLOW® input file, as discussed in the next paragraphs.

PowerFLOW® frame measurement time and coupling timestep

In the test cases related to the coupling demonstrator, the coupling timestep Δ_T is prescribed as a fraction of the period of the pitch motion T_α. However, the PowerFLOW® internal timestep Δt_f - which depends on the mesh resolution and flow conditions, as discussed in sec.2.2.2 - is very likely not to be a perfect

[3] This is the case of the coupling, as the initial transient is aimed at computing the force signal due to an initial AoA or plunge velocity. When both initial conditions are set to zero and a zero F_0 is expected (this is the case of the 1 DoF coupling with pitch motion modelled by body forces sec.6.2) the convergence check is disabled.

fraction of Δ_T. Consequently, the actual timestep employed in the coupled aeroelastic simulation - and so the duration of each Simpack® simulation - is the frame time used by PowerFLOW® to sample the body surface airload to generated the binary CSNC measurement file shown in fig.5.12. This approach benefits the coupling in two ways. The first one is that no time-mismatch between the two solvers is introduced, as Simpack® and PowerFLOW® are well synchronized, in this way, spurious phase errors are not introduced into the FSI process. The second one is related to the setup of the measurement file in PowerCASE®, where the frame time is determined as the integer multiple of Δt_f returning the closest estimate of the desired Δ_T as a fraction of T_α. Another important aspect related to the PowerFLOW® measurement file is that, by prescribing an interval for the time-averaging of the computed solution equal to the sampling frame time, the force value sent to Simpack® is a better estimate than the force value computed at the last fluid subcyle within the coupling timestep, as discussed in sec.2.1.2. Furthermore, as suggested by [4], this choice helps in preserving the stability of the first order explicit CSS FSI scheme discussed in sec.5.1.3 when fluid subcycling is employed.

Conversion of the CFD binary measurement file by means of the PF-CLI

In a PowerFLOW® CFD simulation, the user needs to define measurement windows to save to disk the fluid-flow quantities of interest. To extract the airfoil surface airload, the same binary ".csnc" *Composite measurement file* employed in the PowerFLOW® validation activity of ch.4 is employed. Such binary file is generated by the PowerFLOW® simulator at the beginning of the simulation and updated every frame (coupling timestep). Due to the lack of information regarding the internal data-structure of this file in both [52], [53] and [87], its conversion into an ASCII format has to rely on functionalities provided by the PowerFLOW® software. The most suitable for this purpose is the function "exaritool forces.ri" from the PowerFLOW® CLI with additional attributes to get the required data in the desired format, as described in the "Model" paragraph of the appendix sec.D.2. To whole process is handled within the Coupling Interface by a specific class called by the "TriggerManager" (the appendix sec.D.3). The functions of the PowerFLOW® CLI are also required to start the coupled aeroelastic FSI process by launching the PowerFLOW® simulation[4]

PowerFLOW® input-table reading

As already described in sec.2.2.2, PowerFLOW® can read input tables during the simulation in two different ways. The one employed in this coupling is based on the option *Read After Measurement Frame Written*, as it allows to launch a bash command to trigger the Coupling Interface before reading the subsequent table. Besides, the import of the subsequent table is carried out only after the update of the measurement file, which is necessary to feed the Interface with a new force value. Therefore, this option is perfect to handle both input and output data, as required by a two-way coupling loop. To correctly perform such repeated table reading, the name of the PowerFLOW® input file must include the current reading iteration-number carried out by the simulator. Besides, such number coincides with the "tot_it" quantity previously defined and, since the very first value for tot_it must always be 1 [52], the corresponding table "PFin_1.txt" is directly loaded during the simulation setup. In this way, when the simulation starts, its data are immediately applied to the simulation. Once the simulation is started, PowerFLOW® spans the interval [0:Δ_T], and at $t = \Delta_T$ the Interface is triggered to generate a new table PFin_2.txt. After loading the new table, PowerFLOW® computes the airload during the second iteration [Δ_T:2Δ_T] and at $t = 2\Delta_T$ the Coupling Interface is triggered again repeating the process, as fig.5.11 shows. The bash command that is executed by the PowerFLOW® simulator on the Linux terminal

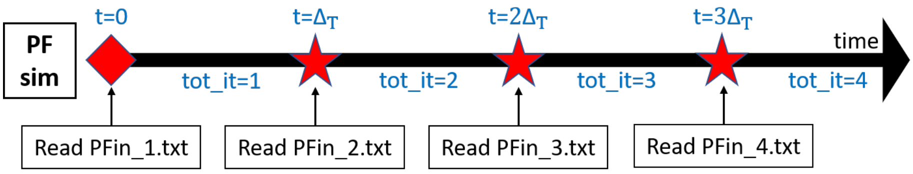


Figure 5.11: The process of "PFin_<tot_it>.txt" table reading done by the PowerFLOW® simulation.

[4] This is done by the "exaqsub" function when directly passed to the Linux terminal, followed by the attribute "-nprocs" to prescribe the number of processors on which the simulator can parallelize the calculations.

launches the Coupling Interface by initializing a clean bash environment with only basic commands and the $HOME environmental variable to execute the Simpack® solver in background mode. In this way, all the possible PowerFLOW® environmental variables that might create conflicts with Simpack® and cause a failure of the coupling loop are ignored.

Reconstruction of the body motion in the CFD solver

As already extensively described, the modelling of the airfoil pitch motion is carried out by both a truly rotating geometry (LRF-approach) or by the employment of body forces, while the plunge motion can only be modelled by a transpiration velocity. The LRF-approach requires an input pitch angular velocity, the wall velocity a plunge velocity, while the body forces need both pitch angular velocity and acceleration. If in ch.4 the previous quantities are prescribed by analytical expressions evaluated every internal fluid timestep Δt_f by the PowerFLOW® solver, in the case of the coupling, they are provided by Simpack® only every Δ_T, leading to the problem of the reconstruction of the airfoil motion during the internal subcycles of the aerodynamic solver. This issue is not present in the case of the coupling demonstrator, but it is present in the case of PowerFLOW® due to its time-explicit formulation and should be taken into account. As the pitch angular velocity and acceleration, as well as the plunge velocity (depending on the motion approach), keep a constant value during the PowerFLOW® subcyles spanning Δ_T, the pitch and plunge positions are reconstructed in PowerFLOW® by means of a linear interpolation. If this choice is in agreement with [4] in terms of even distribution of the body motion during the subcycles of the fluid solvers, the lack of continuity of the velocity and acceleration states at every coupling instants could be a limitation for aeroacoustic simulations, due to the presence of artificial perturbations in the surface pressure distributions, as discussed in sec.2.1.2.

5.4.2. Coupling mechanism

As shown in fig.5.12, the iteration process for the coupling between PowerFLOW® and Simpack® is very similar to the one discussed in sec.5.2.1 for the coupling demonstrator between Simpack® and the Theodorsen routine. Therefore, attention is given in this section only to the differences related to the high-level working of the coupling mechanism.

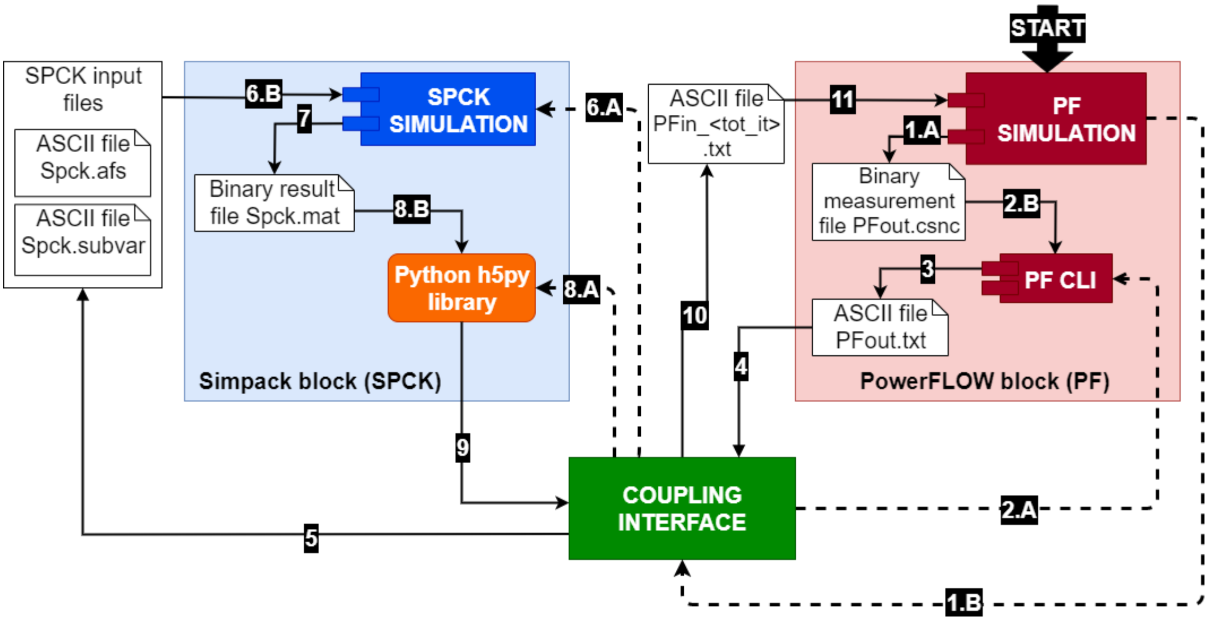


Figure 5.12: Diagram showing the working mechanism of the coupling between PowerFLOW® and Simpack® in terms of software components and main files exchanged.

The first obvious difference is in the aerodynamic block, as it contains the CFD simulation and its CLI component. In between the two software blocks, the binary result "PFout.csnc" file is generated (step 1.A) by the CFD solver. Next, the simulator triggers the Coupling Interface (step 1.B) which then executes the PowerFLOW® CLI to convert the binary "PFout.csnc" file into the ASCII "PFout.txt" file by means of the "exaritool" function (steps 2.B and 3). Next, such file is read by the Interface (step 4), and the other steps until the 7th are exactly the same as those carried out by the Simpack®-Theodorsen coupling described before. At step 7, the difference is in the type of output file generated by the Simpack® *Time Integration Solver*, which is the "Spck.mat" Matlab file. This file is then processed by the Interface exploiting the "h5py" python library (step 8.A) to extract the required kinematic states data (step 9). These aspects are described more in detail in sec.5.4.3 in addition to the motivations why this part of the coupling mechanism has been modified with respect to the corresponding one in the coupling demonstrator. The last two steps are related to the generation of the PowerFLOW® input table - namely the "PFin_"tot_it.txt" ASCII file - (step 10) and the reading of such a file by the CFD simulation (step 11), which can then continue to the following coupling iteration. A final aspect to highlight is that differently from the coupling demonstrator, where the overall handling of the aeroelastic simulation and triggering of each coupling iteration is implemented in a dedicated python script, in the case of the PowerFLOW®-Simpack® coupling, both tasks are carried out by the PowerFLOW® simulation. In other words, the CFD solver is the main actor of the coupling by both starting the coupled aeroelastic simulation and triggering the Coupling Interface every Δ_T.

Regarding a more in depth description of the main components previously mentioned, the general aspects of the PowerFLOW® simulation setup can be found in sec.4.1, while the details related to the 1 DoF and the 2 DoFs versions of the coupling are in sec.6.1, sec.6.2 and sec.6.3. Regarding the Simpack® MBD setup, its general aspects are covered in sec.3.3, while the more detailed ones concerning the test problems are in sec.5.3.1 and sec.5.3.2. Next, when it comes to the Coupling Interface, its main aspects are discussed in ch.5.4.4, while an extensive documentation of the tool can be found in the appendix sec.D. Finally, the coupling files shown in fig.5.12 are introduced in the following section.

5.4.3. Coupling Files

Due to the additional complexity introduced by the CFD solver, more secondary/auxiliary coupling files are employed for the correct working of the coupling between PowerFLOW® and Simpack®. Regarding the main coupling files primarily dedicated to the exchange of data between the MBD and the CFD solver, "PFin_<tot_it>.txt" replaces "kin_states.csv" in the way the latter was employed in the demonstrator coupling, and "PFout.txt" is in place of "TheOUT.csv", for the sake of input and output files of the CFD solver respectively. As pointed out in the previous section, the "Spck.sbr" output binary file is replaced by the "Spck.mat" file but it contains the same type of data. All in all, the coupling files used for the coupling between PowerFLOW® and Simpack® are shown in tab.5.3 and, in the following paragraphs, attention is focused on the purpose and data carried by those files specifically related to this coupling. This is the case of "PFout.csnc", "PFout.txt", "PFin_<tot_it>.txt" and "Spck.mat" for the main coupling files. For the secondary files, "cpl_setup.txt", "init_trans_data.txt" and "cpl_force.csv" are shortly described. In the case of all the other files, some are exactly the same as the ones employed in the demonstrator coupling ("kin_states.csv" and "Spck.subvar"), others are just generated differently or contain slightly different data. For instance, the "Spck.afs" file is filled with the airload returned by PowerFLOW®, while the "tot _iter_list.txt" file contains the number of total iterations ("tot_it") carried out. Also in this case, the details of the structure and generation process of such coupling files can be found in the appendix ch.D.

PowerFLOW®-Simpack® Coupling Files	
Main Files	Secondary Files
PFout.csnc & PFout.txt	cpl_setup.txt
Spck.afs	tot_iter_list.txt
Spck.subvar	init_trans_data.txt
Spck.mat	cpl_force.csv
PFin_<tot_it>.txt	kin_states.csv

Table 5.3: Main and secondary files employed for the coupling between PowerFLOW® and Simpack®.

Main coupling files: PFout.csnc and Pfout.txt

As already mentioned, the binary "PFout.csnc" file is generated at every coupling timestep and the airload values it carries are obtained from a time-averaging spanning the measurement frame time. Its ASCII-conversion is performed by the "exaritool force.ri" function of the PowerFLOW® CLI with additional attributes to decompose the airload into the two in-plain force components (lift and drag) and an out-of-plain aerodynamic moment, including also the corresponding simulation time to which the airload is related. Besides, each force/moment component is reported in the desired units (N/m and N) due to the 2D *Dimensionality* option of the CFD simulation.

Main coupling file: PFin_<tot_it>.txt

The "PFin_<tot_it>.txt" file contains the values of the kinematic states at the current iteration returned by the Spck.mat file and necessary to reconstruct the airfoil pitch and plunge motions in the CFD simulation. To prescribe a new value for the wall-velocity BC modelling the plunge motion in the case of both the 1 DoF and the 2 DoFs versions of coupling, the current value of the plunge velocity is needed, while, in the case of the 2 DoFs version, also the current value of the pitch angular velocity is required to let the airfoil rotate according to the LRF approach. Therefore, the "PFin_<tot_it>.txt" PowerFLOW® input table always carries the Simpack® plunge velocity at t_n and in the 2 DOF version also the pitch angular velocity always at t_n. In addition, such an input file to the CFD simulation contains also additional parameters, such as the time at which the PowerFLOW® initial transient is over, and a flag indicating whether the coupled aeroelastic simulation is in its initial transient or coupling phase. These quantities are related to the fact that, when the aerodynamic simulation is in its initial transient phase, only the prescribed initial conditions for the pitch and plunge motions are active and kept constant, as long as the initial airload value has reached convergence. Moreover, in the case of the 1 DoF coupling, once the convergence of the initial airload is reached, the harmonic functions contained in the analytical expressions for the prescribed pitch motion need to be time-shifted, as the actual pitch motion starts after the end of the initial transient and not when the CFD simulation is stared, as shown in sec.6.1 and sec.6.2.

Main coupling file Spck.mat

In the coupling demonstrator, the "Spck.sbr" binary result file is employed to export the kinematic states computed by the Simpack® *Time Integration Solver*. The data in this binary file can be easily extracted and written to an ASCII file by the Simpack® Post-processor and the whole process can be also readily automated. The only issue of this approach is that it is quite time consuming - generation of the "Spck.sbr" file, its compression, loading the "Spck.spf" post-processing file and generation of the "SpckOUT.csv" file - and requires the Linux "-X server" application for graphics display to be activated. This last aspect is what caused the coupled simulation to crash several times, due to the large number of executions of the "-X server", especially when this is employed with the "ssh" command to execute the coupling simulation on an a remote server. For this reason, a solution is found in the replacement of the "Spck.sbr" file with the "Spck.mat" file, which is a special type of binary file developed by Mathworks for the popular Matlab program. The great advantage offered by a *.mat* file is that it can be easily processed by open source Application Programming Interfaces (APIs). As the whole Coupling Interface is written in python language, the python library "h5py" [90] is employed to extract data from *.mat* files in the v7.3 format, like those generated by Simpack® 2019. Once the kinematic states are extracted, they can be directly handled by the Coupling Interface to update both the "kin_states.csv" file or to generate a new "PFin_<tot_it>.txt" file without going through the "SpckOUT.csv" file. This saves time, and does not require the Linux "-X server" to be activated, improving the stability and robustness of the coupling methodology in addition to its overall performance.

Secondary coupling files

Similarly to the coupling demonstrator case, files like "cpl_force.csv" and "kin_states.csv" serve as data-containers for the time-histories of the main aeroelastic quantities of interest (aerodynamic forces and kinematic states after the PowerFLOW® initial transient). With this regard, the "cpl_force.csv" file is the analog of the "Theodorsen_force_hist.csv" file in the demonstrator coupling. Therefore, these files are essential for both post-processing reasons but also for a quick comparison with the reference solution while the simulation is running. On the other hand, "tot_iter_list.txt", "init_trans_data.csv" and "coupling_setup.txt" are required for the working mechanism of this coupling in the way it is conceived. Briefly, the "init_trans_data.csv" file stores important data related to the end of the CFD initial transient, namely the tot_it and time of convergence as well as the value of the converged force components and their error. The "cpl_setup.txt" is loaded only once at the beginning of the CFD simulation and provides both PowerCASE® and Simpack® (via the "Spck.subvar" file) with parameters like the kinematic states initial conditions, amplitude and frequency of the prescribed pitch motion for the 1 DoF coupling and the coupling timestep.

5.4.4. Coupling Interface

The Coupling Interface developed for the coupling between PowerFLOW® and Simpack® consists of a main python script called "InterfaceManager" calling several other python libraries and the Javascript "SpckSimManager.sjs" code for the automatic setup and execution of the Simpack® simulation. The python libraries are responsible for the steps of the coupling mechanism defined in sec.5.4.2 and, among them, two main ones manage all the tasks required by the initial transient and the coupling phases of the aeroelastic simulation. These are the "TransientIterationManager" and "CouplingIterationManager", with the first specifically related to this CFD-MBD coupling, while the second one is rather similar to the Coupling Interface discussed for the Theodorsen-Simpack® demonstrator coupling. The three previously mentioned main libraries of the Interface are here described, while all the other lower-level libraries and the Simpack® simulation script can be found in the appendix ch.D.

Interface Manager script

The main component of the Interface is the "InterfaceManager" python script which calls all the other lower-level libraries and handles the overall coupling iteration mechanism within a single iteration. The main structure of this script is shown in fig.5.13 and for every coupling iteration, the "InterfaceManager" script is launched directly by the PowerFLOW® simulator via the "trigger_interface.sh" bash script. Initially, the script receives the absolute paths of all the required coupling files, the user defined parameters required for the coupling (force tolerance for the initial transient check and the flow-pass value in terms of iterations number) and the "tot_it" parameter from the "tot_iter_list.txt" file. After these preliminary steps, the script proceeds by resetting the "cpl_force.csv" and "kin_states.csv" files, if tot_it=1, to have empty data-containers at the begging of the coupling. On the other hand, for tot_it≠1, the InterfaceManager always converts the binary "PFout.csnc" file into the ASCII "PFout.txt" file. As third step, the script contains an "if-statement" which allows to activate either the initial-transient or the coupling mode of the Interface. The switching condition is based on the outcome of the search for the "init_trans_data.csv" file, which is generated by the Interface only when the PowerFLOW® initial transient is over. If this file is not found, then the CFD simulation is still in its initial transient and the Interface launches the "TransientIterationManager" library. On the other hand, if the "init_trans_data.csv" file is found, then the initial transient is over and the "CouplingIterationManager" can be executed to couple PowerFLOW® with Simpack®. Eventually, once the previous libraries have finished their internal operations, then also the InterfaceManager has completed its operations and the process can go back to the PowerFLOW®

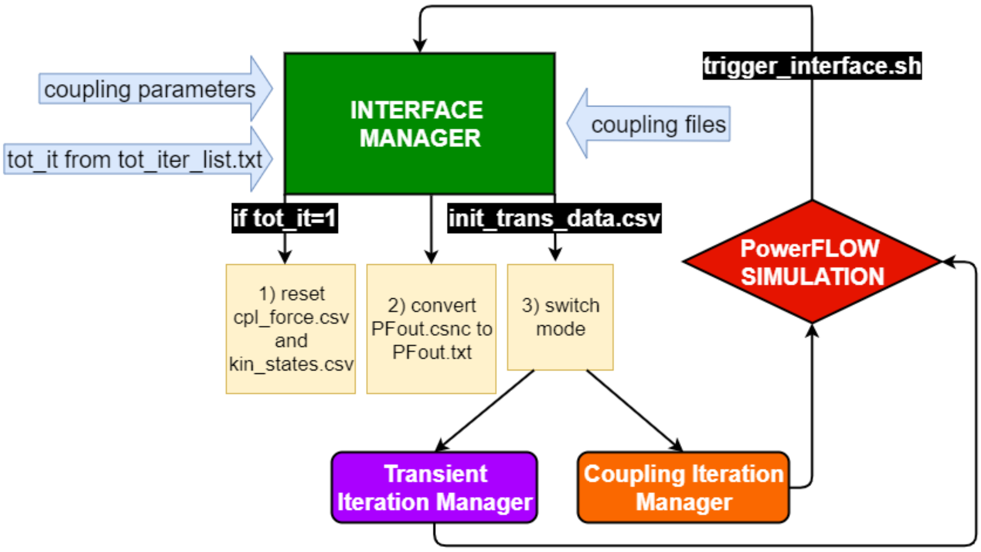

coupling parameters
tot_it from tot_iter_list.txt
INTERFACE MANAGER
coupling files
trigger_interface.sh
if tot_it=1
init_trans_data.csv
PowerFLOW SIMULATION
1) reset cpl_force.csv and kin_states.csv
2) convert PFout.csnc to PFout.txt
3) switch mode
Transient Iteration Manager
Coupling Iteration Manager

Figure 5.13: Diagram representing the internal working of the "InterfaceManager" script and the way it is linked to the two modes of the Coupling Interface and the PowerFLOW® simulation.

simulation, which now advances in time.

Initial Transient Iteration Manager library

The initial transient mode of the PowerFLOW®-Simpack® Coupling Interface is activated when the CFD simulation is in its initial transient phase. Its purpose is primarily to check whether such initial flow transient is over or not and to perform different operations based on the outcome of this check, as shown in fig.5.14. If the initial transient is not over - which can be due to both a value of "tot_it" below the window size or a missed convergence for the force signals considered in the convergence check - then a new "PFin_<tot_it>.txt" file for the following iteration is generated as a copy of the current one and the "tot_iter_list.csv" file is updated. On the other hand, if the convergence condition (see sec.5.4.1) is satisfied for at least one force signal, then different tasks are carried out. Firstly, the "init_trans_data.txt" file is generated with the data mentioned in sec.5.4.3, because these are required by the "CouplingIterationManager" for its internal operations. Secondly, the "cpl_force.csv" file is updated with the very first values of the converged force signals (averages within the last moving window) corresponding cpl_it=0. An thirdly, the "CouplingIterationManager" library for cpl_it=0 is triggered to directly launch the coupling with Simpack® at the iteration where the CFD initial transient is considered to be over.

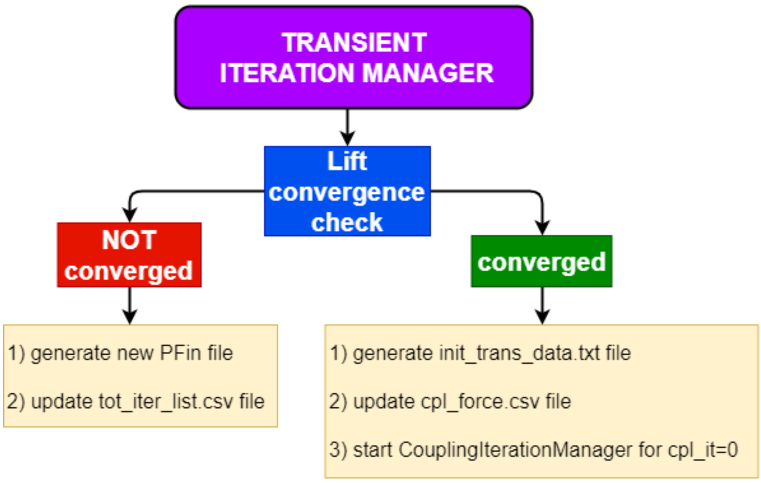

TRANSIENT ITERATION MANAGER
Lift convergence check
NOT converged
converged
1) generate new PFin file
2) update tot_iter_list.csv file
1) generate init_trans_data.txt file
2) update cpl_force.csv file
3) start CouplingIterationManager for cpl_it=0

Figure 5.14: Diagram showing the logic and the main tasks carried out by the "TransientIterationManger" library during the initial transient phase of the CFD-MBD coupled simulation.

Coupling Iteration Manager library

The second mode of the Interface is the one implementing all the necessary operations to couple Simpack® with PowerFLOW®, once the initial transient of the CFD simulation is over. This library can be either launched directly from the "InterfaceManager" script, if the "init_trans_data.txt" file does exist in the Simulation folder, or it is triggered by the "TransientIterationManager" at the "tot_it" when convergence is reached. In the latter case, the "cpl_force.csv" file is not updated, as this operation is already carried out by the TransientIterationManger library. On the other hand, when cpl_it$\neq$0, the "cpl_force.csv" is updated with the last value of the force signals from "PFout.txt" file as the first task carried out by the "CouplingIterationManager" library. The subsequent operations up to the generation of the "Spck.mat" file are the same as those described in sec.5.2.3 for the demonstrator coupling. After the "Spck.mat" file is obtained, the library proceeds by extracting the results of the MBD simulation from the "Spck.mat" file itself to both update the "kin_states.csv" data-container and to generate a new "PFin_<tot_it>.txt"file. Finally, the "tot_iter_list.txt" file is updated.

6

Results CFD-MBD PowerFLOW®-Simpack coupling

After the verification activities related to the setups of the Simpack MBD simulation in ch.3 and the PowerFLOW® CFD simulation in ch.4, as well as an extensive description and verification of the coupling methodology in ch.5, in this chapter, the results of the coupling between PowerFLOW® and Simpack are shown and verified. The solution of the 1 DoF version in the case of pitch motion modelled by the LRF-sliding mesh method is compared with a reference numerical solution based on the original Theodorsen model in sec.6.1. Next, the 1 DoF version with pitch motion modelled by body forces is compared with the previous coupling with pitch-LRF in sec.6.2. Finally, the results of the 2 DoFs version of the coupling with pitch motion modelled by the LRF approach can be found in sec.6.3, where the validation is again carried out by means of the numerical solution of the corresponding analytical model for the aeroelastic problem investigated.

6.1. 1DoF coupling with pitch motion modelled by sliding mesh (LRF)

6.1.1. Testing conditions

The first investigation considers an airfoil undergoing free plunge and forced harmonic pitch motion, which is the same problem described in sec.5.3.1 for the coupling demonstrator. The coupling between PowerFLOW® and Simpack® for this specific problem is tested on the four different conditions specified in tab.6.1 in terms of the motion parameters for the forced pitching motion and the translation stiffness coefficient. Regarding the coupling timestep value employed, based on the results of the sensitivity study carried out for the same aeroelastic problem by means of the coupling demonstrator in sec.5.3.1, a target coupling step of around 200 (precisely 192) fractions of period of the prescribed pitch motion is the criterion here employed to minimize the effect of the underlying explicit FSI algorithm on the computation of the aeroelastic solution. Therefore, given the two values of pitch reduced frequency employed, two values of the target coupling timestep are applied: $\Delta_T = 0.002$s when $\mathrm{k}_a = 0.1082$ ($T_\alpha = 0.384$) in test cases #1 and #3, as well as $\Delta_T = 0.001$s when $\mathrm{k}_a = 0.2164$ ($T_\alpha = 0.192$) in test cases #2 and #4. The word "target" for the chosen coupling timestep, is related to the fact that the actual coupling timestep employed in the computations is the PowerFLOW® frame time of the measurement file "PFout.txt", which is an integer multiple of the internal CFD solver's timestep. As mentioned in

#test	A_α [deg]	α_0 [deg]	k_a [-]	f_α [Hz]	T_α [s]	k_h [N/m]	k_p [-]	t_1 [-]
#1	1°	$-1°$	0.1082	2.6	0.384	5900	0.1138	0.71
#2	1°	$-1°$	0.2164	5.2	0.192	23000	0.2246	0.74
#3	2°	$-2°$	0.1082	2.6	0.384	5900	0.1138	0.71
#4	2°	$-2°$	0.2164	5.2	0.192	23000	0.2246	0.74

Table 6.1: Testing conditions for the four cases considered in this verification activity of the PowerFLOW®-Simpack® coupling with 1DoF and pitch-LRF. The parameters A_α, α_0, k_a and f_α are respectively the pitch amplitude, initial condition for the AoA (obtained by a rotation of the "freestream_csys" in the CFD setup), reduced frequency and frequency of the pitching motion. Next, k_h is the value of the translation spring stiffness coefficient, k_p is the plunge reduced frequency and t_1 is the coefficient employed to tune the wall velocity based on the results of sec.4.5.

sec.5.4.1, this is done to have perfectly synchronised PowerFLOW® and Simpack® simulations, although the two quantities differ by 0.02% when $\Delta_T = 0.002$s and 0.09% when $\Delta_T = 0.001$s. In any case, it important to use the effective time at which PowerFLOW® returns the airload to avoid phase mismatches that can potentially have an impact on the solution for large times. Finally, the only (non-zero) initial condition employed is a negative AoA (α_0 parameter shown in tab.6.1) necessary to obtain a lift force oscillating around a zero mean value, given the expression for the angular velocity of the rotating mesh (LRF approach) discussed in the next subsection.

6.1.2. Main aspects of the CFD and MBD setups

Regarding the general setup of the CFD simulation, in addition to the quantities in tab.6.1, the flow conditions and airfoil geometric properties employed for this investigation are the same as those in tab.4.1 employed to the assess the CFD setup. These flow conditions lead to a flow pass time of 0.0132s covered in about 14 or 7 coupling iterations. Besides, the size of the SimVol, the chord length and the fine resolution employed (512 voxels per chord) lead to a value for the internal PowerFLOW® timestep $\Delta t_f = 1.5 \cdot 10^{-6}$s. This means that, given the two coupling timesteps considered ($\Delta_T = 0.002$s and $\Delta_T = 0.001$s), the subcyling factors of the fluid phase are equal to $n_{sc,f} = \Delta_T/\Delta t_f = 1340$ and 670 respectively.

Focusing on the expressions for the angular velocity employed to move the sliding mesh LRF during the CFD simulation, the same expression shown in eq.4.1 is employed to get a co-sinusoidal pitch oscillation. This returns the desired initial value of the AoA α_0 shown in tab.6.1 for each test case and minimizes spurious oscillations in the lift signal due to the start-up of the rotating mesh, as explained in sec.4.1.4. The only difference is that the $\dot{\alpha}(t)$ expression needs to be time-shifted because the LRF starts to move not at $t = 0$, but at t_{cvg}, namely the time at which the initial CFD transient is over, which is related to the convergence of the aerodynamic force to the prescribed tolerance, as already mentioned in sec.5.4.3. This leads to the following expression for the LRF angular velocity, with t_{cvg} as a variable imported from the "PFin_<tot_it>.txt" input table.

$$\dot{\alpha}(t) = -A_\alpha \sin\big(\omega_\alpha(t - t_{cvg})\big) \tag{6.1}$$

Concerning the wall-velocity BC in the CFD solver, always as mentioned in sec.5.4.3, when the initial transient is still active, its value is set to $t_1\dot{h}_0$, with $\dot{h}_0 = 0$ as the initial condition for the plunge velocity specified in the "cpl_setup.txt" input table. Besides, as already discussed in sec.4.5, t_1 is the tuning coefficient of the method computed for the same conditions as those in this 1 DoF version of the coupling. On the other hand, when the aeroelastic coupling is active, the wall-velocity is set to the constant value $t_1\dot{h}_n$ applied to the CFD simulation for the whole duration of the coupling iteration from t_n to t_{n+1}. The value $\dot{h}_n$ is extracted from the Simpack® simulation at the previous iteration and provided always by the "PFin_<tot_it>.txt" input table.

To check the convergence of the only lift signal considered for this version of the coupling, which is necessary to determine the completion of the PowerFLOW® initial transient, it is necessary to prescribe the values of a few parameters required to by the functions in the "InitTransChecker" library (see the appendix sec.D.4.2) implementing such a check. These parameters are a lift tolerance of 1% the average lift signal in the current sliding window and a sliding window equal to 15 iterations, as explained in sec.5.4.1. This means that for the first 15 total iterations of the process, the convergence check is not performed and it takes around 50 additional iterations (0.05s of simulated time) for test cases #1 and #3 and 70 (0.14s of simulated time) for test cases #2 and #4 to reach the convergence and start the true coupling with Simpack®. The choice of a tolerance equal to 1% the average lift value is motivated by the fact that it is reasonable to expect a weakly varying force signal returned by the intrinsically unsteady PowerFLOW® simulation for the initial α_0 conditions considered, due to the attached and incompressible flow conditions.

Finally, the setup of the MBD simulation for this problem is very similar to the one employed in sec.5.3.1 for the coupling demonstrator. The only differences are in the value of the flat-plate chord ($c = 0.46$m) and in the formulas modelling the pitch motion. The latter need to be set according to the resulting AoA motion in the CFD simulation, given the fact that this quantity has an effect on the aeroelastic behaviour of the system considered. Therefore, the pitching motion *Expressions* for the position, velocity and acceleration components of the *Excitation* to apply to the *User Defined Joint type* are:

$$\alpha(t) = A_\alpha \cos\big(\omega_\alpha(t_s + t_n)\big); \quad \dot{\alpha}(t) = -A_\alpha\omega_\alpha \sin\big(\omega_\alpha(t_s + t_n)\big); \quad \ddot{\alpha}(t) = -A_\alpha\omega_\alpha^2 \cos\big(\omega_\alpha(t_s + t_n)\big)$$

where t_s is the time of the n-th Simpack® simulation spanning the interval $[0{:}\Delta_T]$ and $t_n = n\Delta_T$ is the global simulation time at the iteration n imported into the MBD simulation by means of the "Spck.subvar" input file.

6.1.3. Aeroelastic analytical reference solution

The reference aeroelastic solution employed in the validation of the four test cases of this investigation is obtained by using a formulation similar to the one described in sec.5.3.1. The main difference is in the expression of the circulatory lift term, which in this case corresponds to the one provided by the original hybrid time-frequency domain Theodorsen model. This leads to the following EoM:

$$\left(m + \pi\rho b^2\right)\ddot{h} + 2\pi\rho U_\infty b\, C(\mathrm{k}_p)\dot{h} + k_h h =$$

$$= -\underbrace{\left[m x_G \ddot{\alpha}(t) + k_h x_E \alpha(t)\right]}_{p-term} - \underbrace{\left[\pi\rho b^2 \frac{b}{2}\ddot{\alpha}(t) + \pi\rho b^2 U_\infty \dot{\alpha}(t)\right]}_{L_{nc}^{(\alpha)}-term} - \underbrace{\left[2\pi\rho U_\infty b\, C(\mathrm{k}_a) b\dot{\alpha} + 2\pi\rho U_\infty b\, C(\mathrm{k}_a) U_\infty \alpha\right]}_{L_c^{(\alpha)}-term}$$

where the lhs term contains all the unknown structural and aerodynamic terms dependent on the plunge kinematic states, whereas the rhs is filled with all known quantities, due to the prescribed pitch motion $\alpha(t)$. On the rhs term, terms are grouped into three major terms in order to relate this formulation to the one in sec.5.3.1, namely the structural $p - term$, the pitch-dependent non-circulatory lift $L_{nc}^{(\alpha)} - term$ and the pitch-dependent circulatory lift $L_c^{(\alpha)} - term$. Furthermore, by applying the superposition principle allowed by the linearity of the Theodorsen theory, all the $C(k)$-dependent terms are evaluated by employing the plunge reduced frequency k_p when these are related to a plunge state, while, in the case of pitch-related terms, the pitch reduced frequency k_a is used. The values for k_p and k_a reported in tab.6.1 are chosen as very close[1] to the target values of reduced frequency ($k = 0.11$ and $k = 0.22$) employed in the CFD setup verification activity to tune the wall-velocity approach.

6.1.4. Results

Before comparing the results of the coupling with those of the numerical reference solution discretizing the analytical model previously described, it is once again important to define the main possible sources of mismatch and, based on them, what kind of comparison should be expected. As the aeroelastic variables returned by this investigation result from a FSI coupling between a CFD solver and a MBD tool, the identification of all the possible sources of mismatch is related to the combination of those due to the prediction of the aerodynamic forces, those due to the computation of the kinematic states and those related to the FSI algorithm employed to couple the fluid and structural domains.

- As discussed in ch.4, the different assumptions in the modelling of the flow physics made by the theories behind the LBM and the Theodorsen model lead to rather distinct mathematical formulations of the flow dynamic behaviour, which are also discretized with dedicated methods. These differences are substantial, but they are not expected to play a major role, given the low Ma and high Re number flow conditions, as well as the small amplitude of motions considered (inviscid flow assumption is acceptable), which allow to use the simplified Theodorsen model as a reliable reference for validating the CFD results. On the other hand, the fact that the airfoil plunging motion is reproduced in the CFD solver by flow blowing/suction and not by a truly translating geometry is a significant source of mismatch, especially for large and fast motions combined with the uncertainties introduced by a manual tuning of the method. Therefore, a relevant contribution to the global mismatch in the aeroelastic variables comes from the CFD solver, and in particular from the method employed to handle the airfoil plunging motion.

- In the case of the MBD solver, as pointed out in sec.3.4, the physical and mathematical formulations of the problem are similar between the reference solution and Simpack®, with potential mismatches only in the details and tuning of the MBD simulation setup and in the implementation of the time integration algorithm of the solver. It can be stated, however, that the previous aspects have a

[1] However sufficiently separated to avoid the structure resonance of the airfoil, with the latter caused by a frequency of the forcing pitch motion too close to the characteristic one of the system. This explains also the oscillatory nature of the results obtained.

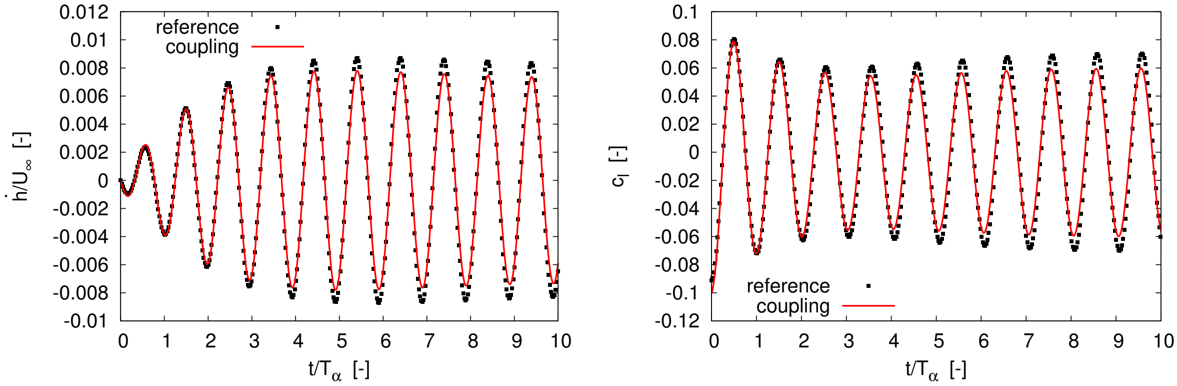


Figure 6.1: Time-histories of the plunge velocity $\dot{h}$ (right) scaled by $U_\infty = 34.7223$m/s and lift coefficient c_l (left) ($\rho = 1.1766$kg/m^3, $c = 0.46$m same U_∞) for the first test case #1 of the 1 DoF PowerFLOW®-Simpack® coupling with LRF (red line) and analytical reference (black squares).

negligible effect, given the perfect match obtained. Therefore, the MBD solver has a very little, if not negligible, influence on the mismatch of the variables here considered.

- Regarding the FSI algorithm, the coupling timestep sensitivity analysis carried out in sec.5.3.1 on the same aeroelastic problem here considered, shows that, as long as the coupling timestep is sufficiently small (around 200 fraction of the characteristic pitching motion period), the influence on the results of the first-order accurate explicit scheme can be considered as very small.

Consequently, it could be argued that a mismatch between the coupling solution and the reference numerical one is expected for this investigation, moreover this can be attributed primarily to mismatches in the prediction of the aerodynamic forces made by the CFD solver.

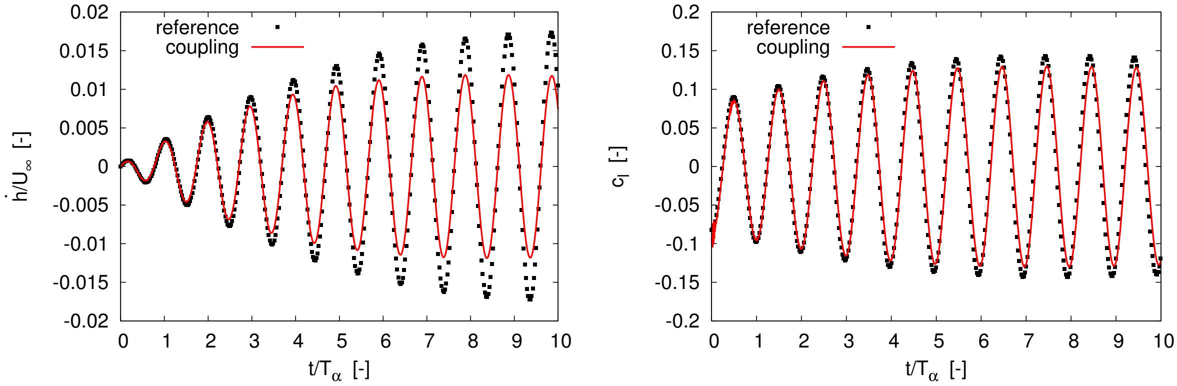


Figure 6.2: Time-histories of the plunge velocity $\dot{h}$ (right) scaled by $U_\infty = 34.7223$m/s and lift coefficient c_l (left) ($\rho = 1.1766$kg/m^3, $c = 0.46$m same U_∞) for the second test case #2 of the 1 DoF PowerFLOW®-Simpack® coupling with LRF (red line) and analytical reference (black squares).

The results of this version of the PowerFLOW®-Simpack® coupling (red lines) compared against the reference numerical solution based on the original Theodorsen model (black dots) can be visualized in fig.6.1 for test case #1, fig.6.2 for test case #2, fig.6.3 for test case #3 and fig.6.4 for test case #4, where the aeroelastic variables considered are the plunge velocity (as this is the quantity required by the CFD solver for the transpiration velocity) and the lift signal; the other kinematic plunge states can be found in the appendix fig.F.10, fig.F.11, fig.F.12 and fig.F.13. Globally, the previously mentioned quantities, for the four test cases considered, are predicted with a good level of accuracy when it comes to the phase error, especially the plunge velocity, during the whole time-range considered. However, an increasing phase mismatch can mainly be noticed in the lift coefficient, which becomes more clear in fig.6.3 and fig.6.4 close to the end on the response. This is consistent with what observed in sec.4.5 for the lift coefficient, where the wall-velocity BC method always returns a phase mismatch compared with the Theodorsen solution under the same input harmonic plunge oscillation and for the same values of reduced frequency considered in the coupling. Moreover, also in this case the phase mismatch is larger for the largest value of reduced frequency and can be considered as a limitation of the approach.

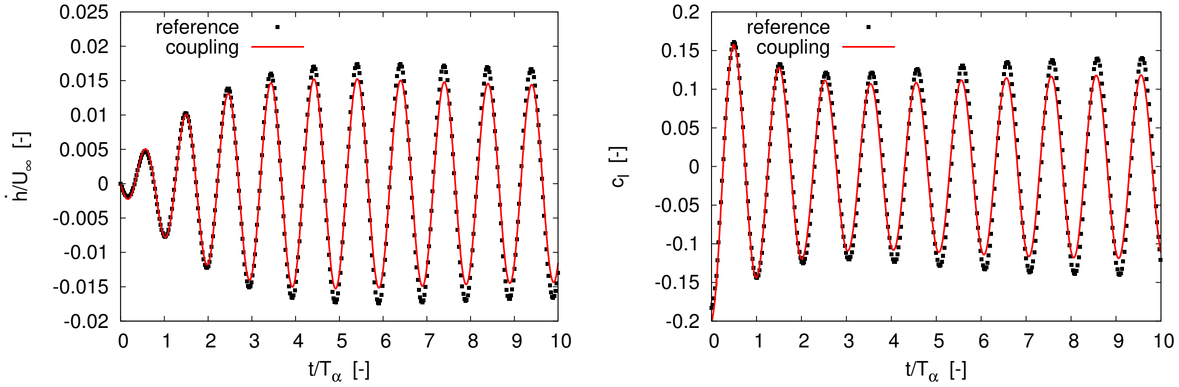


Figure 6.3: Time-histories of the plunge velocity $\dot{h}$ (right) scaled by $U_\infty = 34.7223$m/s and lift coefficient c_l (left) ($\rho = 1.1766$kg/m^3, $c = 0.46$m same U_∞) for the third test case #3 of the 1 DoF PowerFLOW®-Simpack® coupling with LRF (red line) and analytical reference (black squares).

When it comes to the amplitude mismatch in the coupling results, this is quite apparent on both the lift and plunge kinematic states and for all the values of pitch amplitude and reduced frequencies employed. It can be noticed that for $t < 2T_\alpha$ all the aeroelastic variables are well predicted, but after this time, an amplitude underprediction starts to become more and more visible. Furthermore, a closer look reveals that when the amplitude of the response is growing, also the amplitude mismatch increases, whereas, when the response reaches a statistically converged condition, then the mismatch stays constant, with the exception of the lift signal in fig.6.1 and fig.6.3, where the amplitude mismatch still grows for a response decreasing in amplitude. In general, it could be argued that the low frequency cases match better the reference than the high frequency ones, that the effect of amplitude increase does not seem to noticeably affect the results, and that the plunge states have a larger mismatch (up to 40% the maximum values reached by the reference solution in the test cases #2 and #4) than the lift coefficient, where the maximum discrepancy of the latter does not exceed 15% in all the test cases.

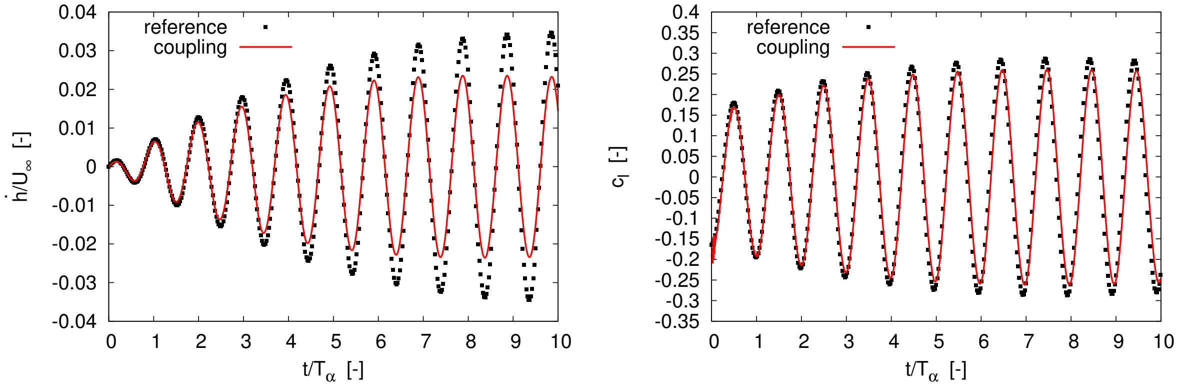


Figure 6.4: Time-histories of the plunge velocity $\dot{h}$ (right) scaled by $U_\infty = 34.7223$m/s and lift coefficient c_l (left) ($\rho = 1.1766$kg/m^3, $c = 0.46$m same U_∞) for the fourth #4 and last test case of the 1 DoF PowerFLOW®-Simpack® coupling with LRF (red line) and analytical reference (black squares).

The amplitude underprediction of the lift signal, might be explained by considering the results of the aerodynamic setup verification activity for the transpiration velocity method in sec.4.5 and the LRF sliding mesh approach in sec.4.4 over the whole range of A_α and reduced frequency values considered in the test cases of this coupling. In both investigations, an underprediction of the maximum and minimum lift values is observed in the lift coefficient returned by the CFD solution with respect to the Theodorsen model, with a much smaller underprediction for the LRF sliding mesh approach (due to a non optimal CFD setup) than the one due to the transpiration velocity (due to the method itself and its tuning). Nevertheless, the amplitude mismatch observed in the coupling lift is significantly larger than the one in the lift signals due to prescribed pitching and plunging motions, and this might be caused by the free plunge motion resulting from the fluid-structure interaction, which is significantly underpredicted in the coupling.

Regarding the relevant amplitude mismatch in the plunge kinematic states, it could be argued that its origin might be due to the aeroelastic reference. The choice to employ the original Theodorsen model has been done as both the time-domain approximations by means of the convolution integral and lag-states methods have led to a much larger mismatch. Even though the employment of the the hybrid time-frequency Theodorsen formulation to compute the time evolution of aeroelastic quantities might be questionable and formally inappropriate[2], in this investigation it has been the only one getting sufficiently close to the coupling results, and for this reason selected as reference solution. The much larger mismatch obtained with the time-domain approximate versions of the Theodorsen model is not shown here for the sake of brevity, but it might be due to errors in the implementation or in the analytical derivation of the 1 DoF airfoil problem with prescribed pitch motion, and it has not been possible to spot even after a long time spent in debugging the code and checking the mathematical derivation. For this reason, it is difficult to judge the degree of accuracy of this 1 DOF coupling with respect to a reference numerical solution based on analytical models, as the reference solution itself might not be completely trusted. At this point, the results of the 1 DoF version modelled by body forces could help in formulating a final assessment.

6.2. 1DoF coupling with pitch motion modelled by body forces

The second version of the coupling consists in the 1 DOF problem with pitch motion modelled by body forces according to the setup and methodology described in sec.2.2.2 and sec.4.1.4. To verify the results of this coupling variant, the results from the previous investigation (1 DoF coupling with pitch motion modelled by sliding-mesh) are here employed as reference solution. This means that the same parameters employed in the four previous test cases are also here exploited, in addition to the coefficients related to the amplitude modulation and phase shift of the body force approach.

6.2.1. Differences with respect to the LRF coupling version

The main and only difference is in the CFD setup is related to the employment of body forces to model the airfoil pitching motion. With this approach, the initial condition for the AoA, differently from the α_0 values shown in tab.6.1, is set to zero for all the four test cases, because the body force approach does not introduce any physical movement of the airfoil[3]. Besides, the virtual AoA oscillation, that the body force method returns, already provides the same zero-mean lift signal returned by the LRF approach. This can be obtained by employing the same pitch angular velocity expression used for the LRF approach (eq.6.1), in addition to the tuning coefficients ξ_1 and ξ_2 computed in sec.4.6 to match the LRF lift from a prescribed pitching rotation and reported in tab.4.7. Therefore, the expressions for the amplitude modulated and phase shifted pitch velocity and acceleration are analogous to those in eq.4.3, with a minor difference to take into account the time-shift due to the initial transient:

$$\dot{\tilde{\phi}}(t) = -\xi_1 A_\alpha \omega_\alpha \sin\left[\omega_\alpha\left(\tilde{t} - \xi_2 T_\alpha\right)\right]; \qquad \ddot{\tilde{\phi}}(t) = -\xi_1 A_\alpha \omega_\alpha^2 \cos\left[\omega_\alpha\left(\tilde{t} - \xi_2 T_\alpha\right)\right] \tag{6.2}$$

[2] As shown in sec.6.1.3, Theodorsen function $C(\mathrm{k}_p)$ in the $2\pi\rho U_\infty b C(\mathrm{k}_p)\dot{h}$ term might have been evaluated for a value of the plunge reduced frequency which is not exactly the one occurring in the plunge response, due to the interaction with the pitching motion which has a slightly different value (k_a). Moreover, the employment of the hybrid time-frequency formulation in a time-domain EoM might lead to an incorrect evaluation of the wake-memory effect, as the one of an airfoil periodically moving for an infinite time and not starting from an initial condition.

[3] In the body force approach, if a non-zero rotation of the "freestream_csys" is applied, then the virtual airfoil pitching will be around a non-zero AoA.

where $\tilde{t} = t - t_{cvg}$ is the shifted time employed to start the pitch motion at the time instant when the initial transient is over. However, since no initial condition is set for the AoA set during the CFD initial transient (no rotation of the "freestream_csys"), the convergence check is disabled and the coupling is started after a number of iterations (15) corresponding to the same window size used for the LRF version of the 1 DoF coupling.

6.2.2. Results

In this investigation, the results of the previous 1 DoF coupling with pitch-LRF approach (reference solution here) are compared with those of the same 1 DoF coupling with pitch-body-force approach, therefore, the only source of mismatch between the two solutions is related to the body force approach, with the already observed limitations in terms of amplitude and reduced frequency of motion (see sec.4.6), as well as uncertainties due to the manual tuning procedure. Being said this, the results of such a PowerFLOW®-Simpack® coupling in its 1 DoF version with pitch motion modelled by the body-force method are shown for the plunge velocity and lift coefficient time-histories in fig.6.5 for test case #1, fig.6.6 for test case #2, in fig.6.7 for test case #3 and finally in fig.6.8 for test case #4. The same comparison for the plunge position and acceleration is shown in the appendix fig.fig.F.14, fig.F.15, fig.F.16 and fig.F.17 for the same test cases. In all these figures, the blue lines are the reference coupling solutions with LRF and the red lines are related to the body-force couplings.

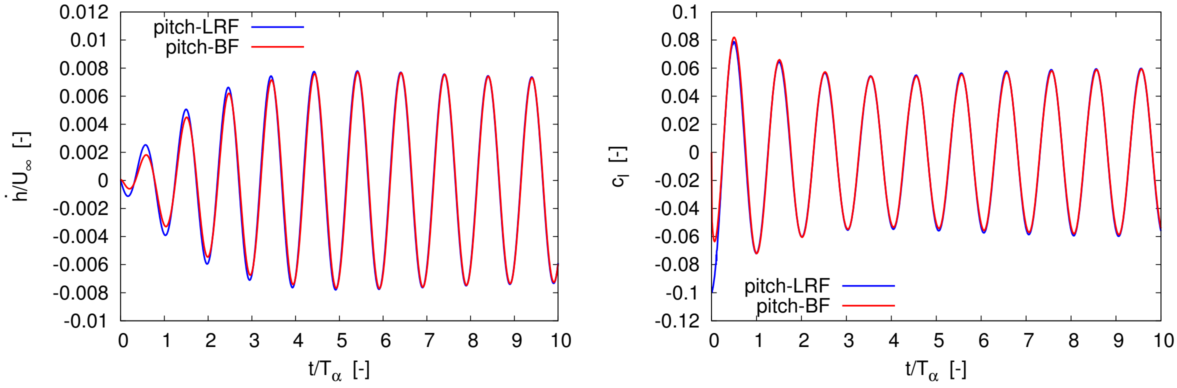


Figure 6.5: Time-histories of the plunge velocity $\dot{h}$ (right) scaled by $U_\infty = 34.7223$m/s and lift coefficient c_l (left) ($\rho = 1.1766$kg/m^3, $c = 0.46$m same U_∞) for the first #1 test case of the 1 DoF PowerFLOW®-Simpack® coupling with pitch body-forces (red line) and the coupling with pitch-LRF (blue line).

What is quite apparent from all the results obtained for this investigation, is that there is a much smaller discrepancy between the two coupling versions than between the LRF-coupling and the reference numerical solution based on the hybrid time-frequency domain Theodorsen model. Besides, a closer look reveals that the lift response of the body-force coupling is reproduced with a good level of accuracy with respect to the LRF coupling approach in all the four test cases. Such a lift prediction of the body force method becomes very accurate for the test cases with the lowest pitch amplitude, while a small over-prediction of the oscillation peaks for the test cases with $A_\alpha = 2°$ is observed, in addition to a different shape of the oscillation wave with a less rounded crest in the case of test case #4. These observations can be related to what noticed in sec.4.6 in the case of prescribed motion, where a small overestimation of $c_{l_{max}}$ due to tuning inaccuracies and changes in the shape of the hysteresis loops are observed for the $A_\alpha = 2°$-$k = 0.11$ and $A_\alpha = 2°$-$k = 0.22$ testing conditions. Furthermore, in the very first time instants of the response, the mismatch between the LRF-coupling and the body-force coupling lift signals is related to the different way the two methods model the initial value of the AoA. On the one hand, the LRF-coupling provides a $c_{l_0} \neq 0$ value corresponding to a non-zero initial condition ($\alpha_0 = -A_\alpha$) for the AoA modelled by a rotation of the "freestream_csys" during the CFD initial transient. On the other hand, the body-force-coupling returns a $c_{l_0} \approx 0$ value, as there is no initial AoA condition ($\alpha_0 = 0$). However, given the same expressions for the pitch velocity and acceleration, when the the body forces are switch-on, the two lift curves get very quickly (in much less than a T_α time) almost one on top of the other and behave almost identically.

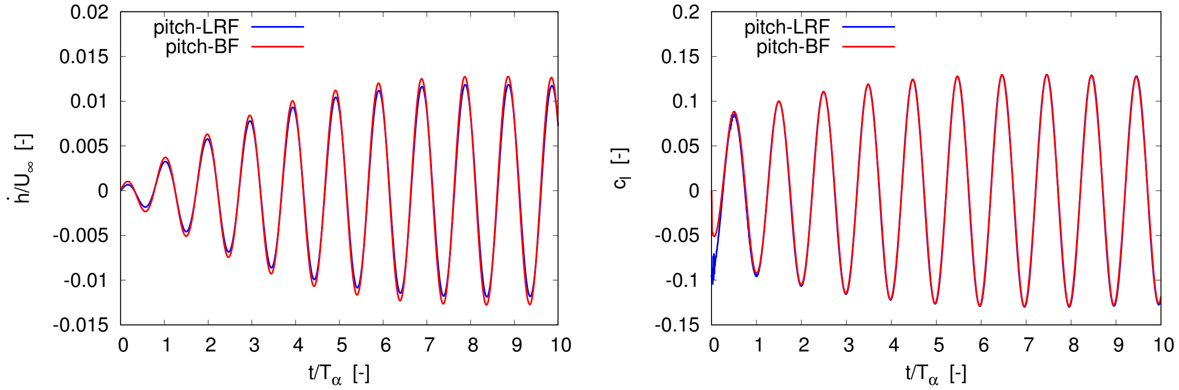


Figure 6.6: Time-histories of the plunge velocity $\dot{h}$ (right) scaled by $U_\infty = 34.7223$m/s and lift coefficient c_l (left) ($\rho = 1.1766$kg/m^3, $c = 0.46$m same U_∞) for the second #2 test case of the 1 DoF PowerFLOW®-Simpack® coupling with pitch body-forces (red line) and the coupling with pitch-LRF (blue line).

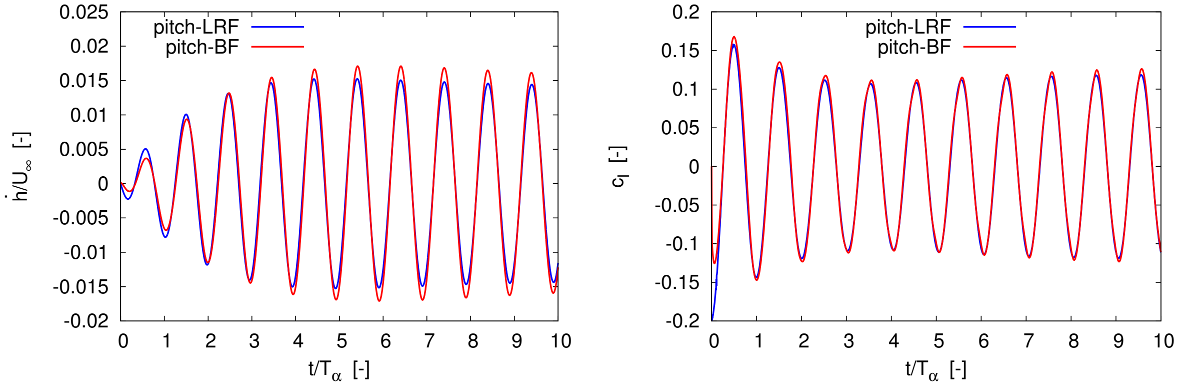


Figure 6.7: Time-histories of the plunge velocity $\dot{h}$ (right) scaled by $U_\infty = 34.7223$m/s and lift coefficient c_l (left) ($\rho = 1.1766$kg/m^3, $c = 0.46$m same U_∞) for the third #3 test case of the 1 DoF PowerFLOW®-Simpack® coupling with pitch body-forces (red line) and the coupling with pitch-LRF (blue line).

Regarding the kinematic plunge states, the different lift initial value leads to a different behaviour of the plunge kinematic states in the first part of the solution ($t < 4T_\alpha$) and in all the four test cases. In the test cases related to the smallest reduced frequency, the larger c_{l_0} of the LRF-coupling, leads to larger values for the plunge velocity for this version of the 1 DoF coupling, while in test cases #2 and #4, the body-force version returns larger plunge velocity values and a smaller amplitude mismatch with the LRF solution. Besides, in test case #1, after such an amplitude mismatch in the first 5 periods of oscillation, the coupling solutions are on top of each other, when a statistically-converged condition in the plunge velocity is reached. This does not occur for the other three test cases, with the amplitude mismatch of tests #3 and #4 most likely due to the lift amplitude overestimation, while the plunge velocity amplitude mismatch of test #2 is difficult to justify given the very good amplitude-match between the corresponding lift signals.

All in all, it is possible to conclude that the coupling with the body force method for the pitching motion is able to give similar predictions of the aerodynamic and kinematic plunge states returned by the coupling with the LRF sliding mesh approach. This is observed on a 1 DoF aeroelastic airfoil problem where only the lift signal triggers the aeroelastic response in the plunge motion. If the results of this body force version of the coupling are related to those of the LRF version, as the two results are very close, it might be argued that the PowerFLOW®-Simpack® coupling on a 1 DoF aeroelastic airfoil problem can lead to more accurate and realistic results than those returned by the reference analytical model employed for its validation. This is because it is not completely possible to trust the results

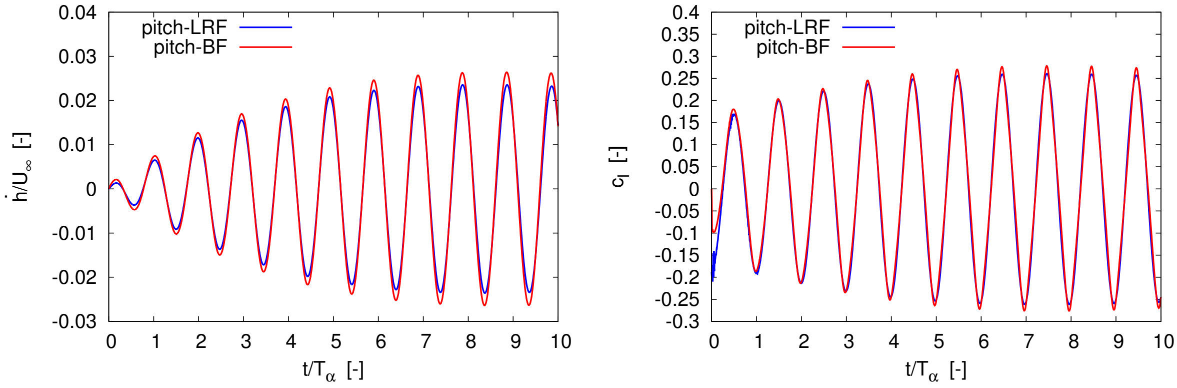


Figure 6.8: Time-histories of the plunge velocity $\dot{h}$ (right) scaled by $U_\infty = 34.7223$m/s and lift coefficient c_l (left) ($\rho = 1.1766$kg/m^3, $c = 0.46$m same U_∞) for the fourth #4 and last test case of the 1 DoF PowerFLOW®-Simpack® coupling with pitch body-forces (red line) and the coupling with pitch-LRF (blue line).

of the analytical model, given the theoretically inappropriate use of the hybrid time-frequency domain formulation of the Theodorsen model in a time-domain EoM, or the potential presence of errors in the implementation or derivation of the reference solution which it has not been possible to spot.

6.3. 2 DoFs coupling

6.3.1. General aspects, testing conditions and setup

Problem definition

As already mentioned, the third and final version of the PowerFLOW®-Simpack® coupling consists in a 2 DoFs aeroelastic problem with lumped structural parameters involving a pitching and plunging airfoil. The free response of this body is triggered by an initial impulsive perturbation of the plunge velocity. Once again, this is the same problem considered in ch.3 to verify the Simpack® setup and in sec.5.3.2 to test the coupling demonstrator.

Goal of the investigation

The main goal of this investigation is to assess the accuracy of the coupling methodology in predicting the time-histories of the kinematic states and aerodynamic forces returned by the aeroelastic system for conditions above, below and in the neighbourhood of its flutter boundary. This leads to four tests conditions corresponding to free-stream velocity values equal to fractions of the flutter velocity U_{flt}. For the below and above-flutter test cases #1 and #2, the inflow velocities $0.9U_{flt}$ and $1.1U_{flt}$ are respectively considered. Regarding the neighbourhood of the flutter boundary, test condition #3 is set to $U_\infty = 0.99U_{flt}$, while test case #4 is related to $U_\infty = 1.01U_{flt}$. The goal of the last two simulations is, therefore, to verify weather or not the flutter velocity of the system can be predicted with a 1% accuracy level.

Reference solution

The reference value of the flutter velocity of the system is obtained by employing the V-g method, whose derivation for the 2 DoFs airfoil problem considered is shown in the appendix ch.E. The V-g method is employed in this investigation as it allows to compute the flutter boundary of an analytical system by directly employing the Theodorsen model in its original hybrid time-frequency domain formulation. Alternatively, a procedure similar to the one employed in sec.3.1.4 - which is based on the calculation of the eigenvalues for the aerodynamic lag-states approximation method of the Theodorsen model - can also be employed. The two methods are expected to give similar results, however the V-g one is more accurate, as it considers all the infinite number of aerodynamic lag-states related to the wake induction effect. Concerning the reference solution for the time-histories of the aeroelastic variables - shown in fig.6.9, fig.6.11, fig.6.13 and fig.6.14 by means of black dots for the sake of comparison with coupling solution (red line) - is based on the Theodorsen model under its time-domain approximation by means of the lag-states method derived in the appendix sec.B.4. Such approximated analytical unsteady aerodynamic model is coupled with the EoMs for the structural side of the system, as shown in sec.3.1.3. Finally, this analytical model is solved numerically by means of the Crank-Nicolson method for a timestep value (0.0001s) leading to a converged discrete solution, as discussed in sec.3.2.2.

Testing conditions

Regarding the geometric and structural conditions for the system considered in the four test cases mentioned before, their values can be found in tab.6.2. These dimensional values are obtained by employing the dimensionless quantities in tab.6.3 and by exploiting the dimensionalization relations shown in the appendix ch.C, where the latter are obtained directly from the definitions of the dimensionless quantities in tab.6.3 reported in eq.3.9. However, differently from the MBD setup verification activity in ch.3, in this investigation different values for the the airfoil chord ($c = 0.46$m), pitch angular reduced frequency ($\omega_\alpha = 35$rad/s) and free-stream fluid-flow density ($\rho = 1.1766$kg/m^3) are employed. These values are also consistent with the CFD setup verification activity in ch.4 and the flow conditions employed in the 1 DoF versions of the coupling. Besides, the employment of a large value of ω_α allows to keep the global duration of the aeroelastic response in the order of 1s, significantly reducing the simulation time. After defining the testing conditions for this investigation, a dimensionless value of $\dot{h} = 0.01$ for the initial plunge-velocity perturbation is applied to the reference solution. Its corresponding dimensional value $\dot{h}_O = 0.01b\omega_\alpha = -0.0805$m/s[4] is applied to both the Simpack® and PowerFLOW® setups. Regarding the flutter velocity, under the parameters above defined for the problem investigated, the V-g method returns a dimensionless value $\hat{U} = 5.1695$, which in dimensional terms translates to $U_\infty = 5.1695b\omega_\alpha = 41.6145$m/s.

Parameter	Symbol	Value	Unit
Airfoil semi-chord	b	0.23	[m]
Aerodynamic center	x_A	0.0	[m]
Mass center	x_G	0.115	[m]
Elastic center	x_E	0.023	[m]
Mass	m	27.38	[kg]
Moment of inertia	J_E	0.36	[kg·m]
Translation stiffness	k_h	21462.41	[N/m^2]
Rotation stiffness	k_ω	443.50	[N/m/rad]

Table 6.2: Values of the dimensional geometric and structural parameters employed for the 2 DoFs PowerFLOW®-Simpack® coupling.

Symbol	Value
r_α	0.5
a	70.0
Ω	0.8
ξ_A	0.0
ξ_G	0.5
ξ_E	0.1

Table 6.3: Values of the dimensionless geometric and structural quantities employed to derive the dimensional problem parameters in tab.6.2.

Coupling timestep

When it comes to the coupling methodology, the first aspect to take into account is the coupling timestep Δ_T of the underlying explicit FSI algorithm employed. Concerning this 2 DoFs version, a timestep analysis is carried out for the first test case at $U_\infty = 0.9U_{flt}$ by considering three values for Δ_T, namely $\Delta_T = 0.002$s, $\Delta_T = 0.001$s and $\Delta_T = 0.0005$s, corresponding to 90, 180 and 360 fractions of pitch period $T_\alpha = 2\pi/\omega_\alpha = 0.1795$s. Initially, also an even smaller value $\Delta_T = 0.0001$s has been considered for a short simulation duration (40% the one of the other three previous simulations), but the simulation has not been completed due to the negligible improvement of results (the mismatches visible and discussed in sec.6.3.2 and sec.6.3.3 have not improved with this even smaller coupling timestep) and the much larger computational cost. The outcome of this investigation - the time-history plots of the aeroelastic variables for changing coupling timestep are not shown for the sake of brevity - is that the smallest timestep value is able to return the best prediction of the time responses of the aeroelastic variables, in particular by reducing the amplitude mismatch at the oscillation peaks. Therefore, in all the test cases, the coupling solutions are obtained with $\Delta_T = 0.0005$s ($T_\alpha/\Delta_T = 380$).

CFD initial transient handling

Another aspect is the handing of the initial transient of the PowerFLOW® simulation. For the 2 DoFs coupling, the force-signal convergence check is applied also to the aerodynamic moment. The convergence

[4] The minus sign is due to the different orientation of the y-axis between the reference frames of the CFD/MBD (positive upwards) computational domain and of the analytical solution (positive downwards).

criterion employs the same window size of 15 coupling iterations and scaled tolerance of 1% the average value of each force signal in its last window, similarly as for the 1 DoF pitch-LRF coupling version. Besides, the convergence of at least one force signal between the lift and the aerodynamic moment is considered and not the convergence of both signals, with a significantly larger error for the moment (from 0.4 to 1.5 times its average value). This has been done as it is much easier to get a better convergence quality of the lift signal with a reasonable amount of iterations (around 120 for test cases #1, #3 and #4 while "only" 83 for test #2). Furthermore, in all the test cases, the converged moment signal is much smaller (two orders of magnitude) than the converged lift one, thus the intrinsic oscillations of the PowerFLOW® solution are much more amplified, and it might not even be possible to satisfy such a small tolerance over the whole simulation duration. Alternatively, a less strict tolerance - say 10% or even 50% - can be considered, but this requires more iterations and might also lead to a less accurate lift signal than the one here obtained. Finally, to require an error for the converged moment below 1% might not even be necessary, since reasonably accurate predictions of the plunge states and c_m can still be reached even with poorly converged initial moment values, as fig.6.9, fig.6.11, fig.6.13 and fig.6.14 show.

PowerFLOW® setup

In the case of the fluid block of the coupling, the PowerFLOW® simulation features the same resolution (fine 512 voxels per chord) and internal timestep ($\Delta t_f = 1.494 \cdot 10^{-6}$s) as in the 1 DoF variants. One difference is in the value of the fluid subcycling factor ($n_{sc,f} = 335$) for the target coupling timestep. As already discussed in sec.5.4.3, the modelling of the pitch motion by means of the LRF approach requires one (constant) value $\dot{\alpha}_n$ from the MBD simulation for the angular velocity of the sliding mesh to be applied for the whole duration of a generic coupling iteration n, leading to a linear variation of the airfoil pitch angle (or AoA for this problem). Similarly, the modelling of the plunge motion relies on the wall-velocity BC by means of the $\dot{h}_n$ plunge velocity value from Simpack® corrected by the tuning coefficient t_1, obtaining also here a linear change for the plunge position. The value of the tuning coefficient is set to $t_1 = 0.72$ for all the four test cases given the range of plunge reduced frequency k$_p$ shown in tab.6.4 and according to the results of sec.4.5. Besides, as discussed in sec.5.4.3, both $\dot{\alpha}_n$ and $\dot{h}_n$ are extracted from the Simpack® simulation via the "PFin_<tot_it>.txt" input table and, to account for the initial transient mode of the simulation, the initial conditions $\dot{\alpha}_n = 0$rad/s and $\dot{h}_n = \dot{h}_O$ are prescribed until the lift signal reaches convergence, as described in the previous paragraph. Coming back to the testing conditions, each one of them differs just in terms of flow-related quantities, as the structural and geometric parameters (see tab.6.2) define a unique system with a specific flutter boundary behaviour. Therefore, to capture the different system response for conditions above, below or in the neighbourhood of the flutter velocity, the only parameter that is changed in the PowerFLOW® setup is the Mach number, leading to the variations of Re number and flow velocity shown in tab.6.4. These aerodynamic conditions are still close to the $Re = 10^6$ and $Ma = 0.1$ values employed in ch.4 for the verification of the CFD setup and within the range of applicability of the Theodorsen theory, as long as the flow stays attached for the NACA 0003 airfoil.

	#1	#2	#3	#4
Ma	0.1079	0.1318	0.1187	0.1210
Re	$1.0981 \cdot 10^6$	$1.3422 \cdot 10^6$	$1.2080 \cdot 10^6$	$1.2324 \cdot 10^6$
U_∞ [m/s]	37.4529	45.7757	41.1981	42.0304
Flow-pass time [s]	0.0123	0.0100	0.0112	0.0109
k$_a$	0.2149	0.1759	0.1954	0.1915
k$_p$	0.1719	0.1407	0.1563	0.1532

Table 6.4: Values of the dimensional aerodynamic parameters employed in the four test cases considered in the 2 DoFs version of the PowerFLOW®-Simpack® coupling, with k$_a = \omega_\alpha b/U_\infty$ and k$_p = \omega_h b/U_\infty$ ($\omega_h = \Omega\omega_\alpha$).

Simpack® Setup

The last aspects to discuss is the setup of the Simpack® MBD simulation, which is essentially the same as the one discussed in sec.3.3 for the Simpack® verification activity in the case of the Theodorsen aerodynamic model and in sec.5.3.2 for the 2 DoFs test case of the coupling demonstrator. Regarding the values of the geometric and structural parameters, these are the same as those shown in tab.6.2, the only difference is in the units which take into account the 3D geometry of the simulation by employing a 1m span-wise length for the flat plate *Primitive* representing the airfoil. The employment of the LSODE adaptive time-step integrator leads to a structural subcycling factor $n_{sc,s} = \Delta_T/\Delta t_s$ in the order of 1 with an internal structural time-step $\Delta t_s >> \Delta t_f$. The internal Δt_s is automatically determined by the solver given coupling time-step duration, problem parameters and numerical aspects, and can change during the simulation depending on the properties of the solution.

6.3.2. Above and below-flutter conditions

Before discussing the results of this 2 DoFs coupling, the same considerations made in sec.6.1.4 regarding the main possible sources of mismatch between the reference and the coupling solutions (fluid and MBD models/setups, as well as FSI algorithm) can also be applied in this investigation. In addition, for this type of problem, the reference solution is based on the lag-states method of the Theodorsen model and, for this reason, it is expected to provide a better representation of wake-memory effect than the one given by the hybrid time-frequency original Theodorsen formulation applied to a time-domain EoM, as done in the 1 DoF coupling. This means that this reference solution can be considered as more trustworthy and, therefore, the expected mismatches with the coupling solution can be essentially attributed to the methods employed in the CDF solver to compute the aerodynamic forces due to the airfoil pitching and plunging motion, in particular limitations of these methods and tuning inaccuracies.

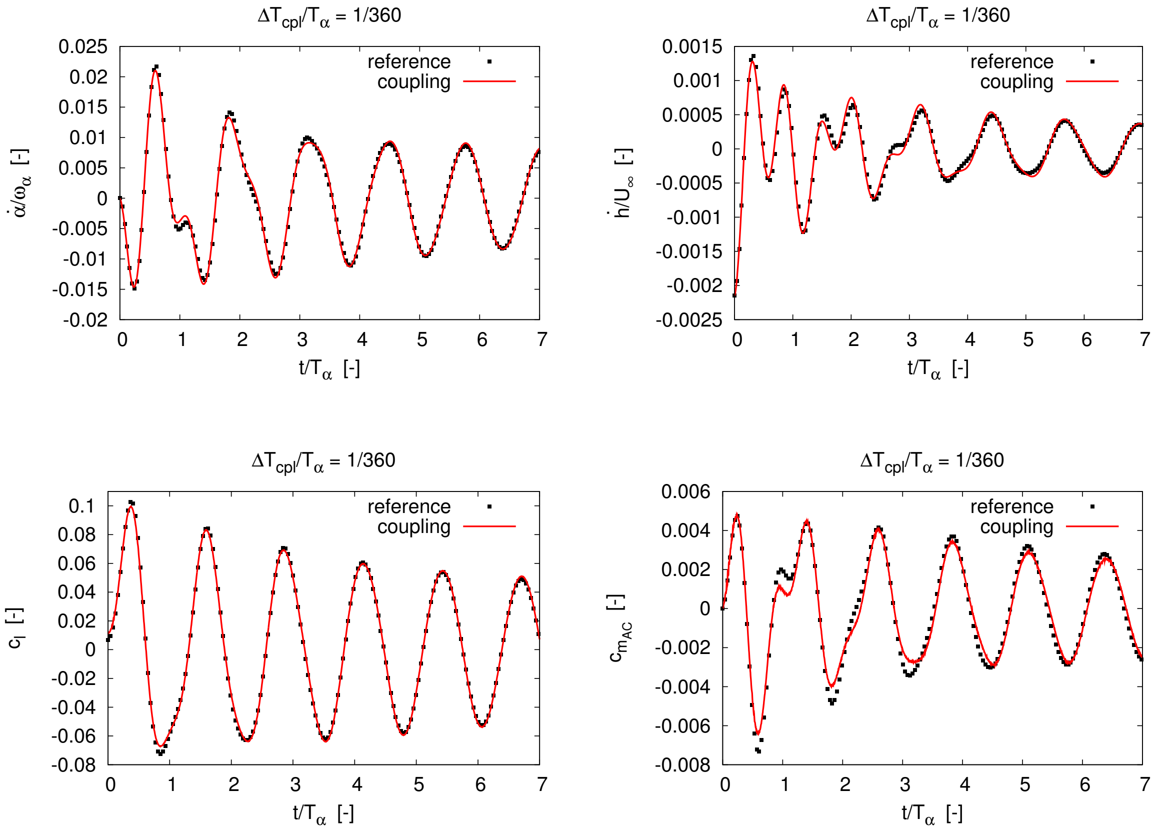


Figure 6.9: Dimensionless time-histories of the kinematic and aerodynamic quantities obtained by the 2 DoFs PowerFLOW®-Simpack® coupling with LRF (red line) against the reference analytical solution (black squares) for an inflow condition $U_\infty = 0.9U_{flt}$ and coupling timestep 0.0005s.

The results (time-histories of the pitch/plunge velocity as well as lift and aerodynamic moment) of test #1 related to a condition below the flutter velocity of the system (namely $U_\infty = 90\%U_{flt}$) are shown in fig.6.9, while the same for a condition above flutter ($U_\infty = 110\%U_{flt}$) can be found in fig.6.11. The time-histories of the other kinematic states (plunge/pitch position and acceleration) can be found in the appendix fig.F.18 and fig.F.19 for the two respective testing conditions. In both test cases, the duration of the two aeroelastic fully-coupled simulations is set to 7 periods of pitch motion, which corresponds to $7T_\alpha = 1.2566$s. The results for test #1 are obtained considering values of pitch k_a and plunge k_p reduced frequencies (shown in tab.6.4) well matching also with the aerodynamic results of the PowerFLOW® setup verification activity shown in ch.4 for the LRF and the wall-velocity investigations. On the other hand, the results of test #2 cover smaller values of k_a and k_p, which are not simulated in ch.4. However, the very small variation of t_1 with the pitch k_p and the similar behaviour of the pitch-LRF airload for changing A_α and k_α show that this is not a relevant issue, as it does not affect the kinematic results significantly when displacements are small.

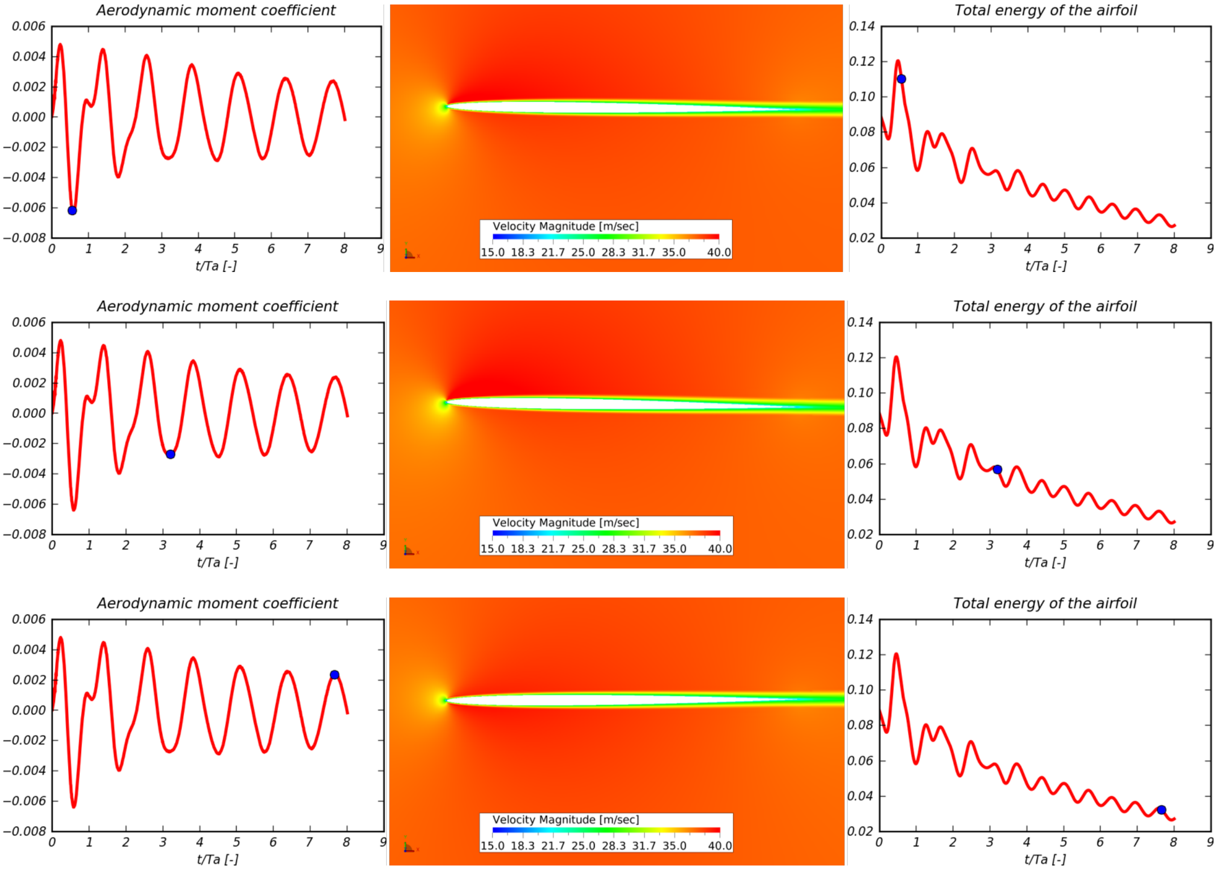


Figure 6.10: Time-histories of the aerodynamic moment (left pictures) and total energy of the structure (right pictures), as well as flow-field snapshots for four time instants of the $U_\infty = 0.9U_{flt}$ test case.

By considering the aeroelastic results for test #1 in fig.6.9, it is possible to state that the coupling solution shows an overall good agreement with the reference numerical solution, with only minor deviations in areas (especially in the plunge states and c_m), where the characteristic time-scale of the response gets significantly smaller than the pitch period $T_\alpha = 0.1795$s. This could be explained by the fact that it is difficult to reproduce plunge oscillations faster than a flow-pass time with the wall-velocity method, since the wall velocity applied over the whole body surface is characterized by an intrinsic memory effect that lasts on the airfoil for at least a flow-pass time. This might be an intrinsic limitation of the PowerFLOW® setup in the description of the airfoil plunging motion, which might restrict the applicability of the coupling methodology to rapidly changing oscillations. Furthermore, the occurrence of this phenomenon also in the pitch states (although less frequent) might be due to both the structural coupling between inertia and elastic forces, due to the presence of distinct mass and elastic centers, and the energy transfer between the pitch and plunge motions due the fluid-structure interaction in the vicinity of the flutter boundary for this 2 DoFs problem. Finally, the fact that a condition below and close to the flutter velocity is simulated explains why the plunge states, after an initial transient, reach a periodic damped motion well captured by the coupling methodology both in amplitude and phase. Therefore, it is possible to argue that the coupling methodology can predict with a good level of accuracy the time response of a 2 DoFs aeroelastic airfoil system close and below a flutter condition.

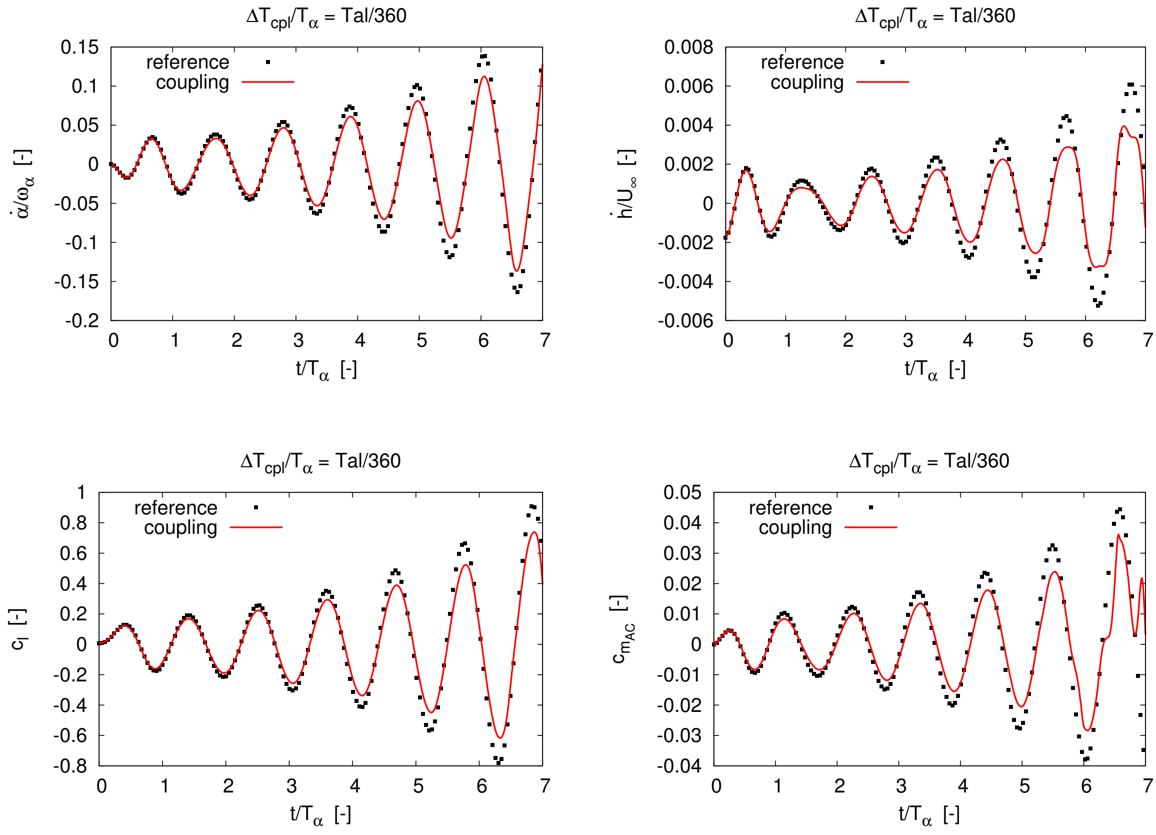


Figure 6.11: Dimensionless time-histories of the kinematic and aerodynamic quantities obtained by the 2 DoFs PowerFLOW®-Simpack® coupling with LRF (red line) against the reference analytical solution (black squares) for an inflow condition $U_\infty = 1.1U_{flt}$ and coupling timestep 0.0005s.

Always in the case of test #1, a few snapshots of the flow field (obtained with the Simulia PowerVIZ® post-processing tool) close to the airfoil are shown in fig.6.10 in addition to the aerodynamic moment coefficient and the total energy of the structure, with the latter defined as the sum of the potential and kinetic energy of the 2 DoFs system shown in eq.3.4 and eq.3.3 respectively, returned by the FSI coupled simulation. The time instants at which the flow field visualizations are referred are marked on the total-energy and c_m plots by a blue cursor and are related to a condition during the transient part of the response (top figures), a condition at the beginning of the smooth converging oscillations (middle figures) and one at the end of the response (bottom figures). All the three flow field visualizations snapshots look very similar, with a clear attached flow, a very thin boundary layer and a smooth wake, because of the small amplitude conditions related to such a below flutter condition. Therefore, the system is expected to return to its initial state after an initial transient and damped oscillations, as the total energy plots clearly show.

Regarding the results of test #2 in fig.6.11, a good match between the reference and the coupling solution can be found as long as the magnitude of displacements is small, say $\dot{\alpha} < 0.05$rad and $\dot{h}/U_\infty < 0.002$. When the pitch motion reaches AoA above 0.05rad (3°) and the chord-scaled plunge-motion amplitudes is beyond 0.005 at $t \approx 3T_\alpha$, discrepancies start to appear and these get larger and larger as the amplitude of motions increases. This does not necessarily mean that the coupling methodology gets inaccurate for large displacements, as the assumptions of both the Theodorsen model (attached flows and negligible viscous effects) and the structural system (linearization of the EoMs for small displacements) are no longer compatible under these conditions, leading to a loss of validity for the reference solution. Therefore, the coupling solution can be considered as more accurate in predicting the behaviour of the system for large displacements, given the more likely better prediction of the flow dynamics returned

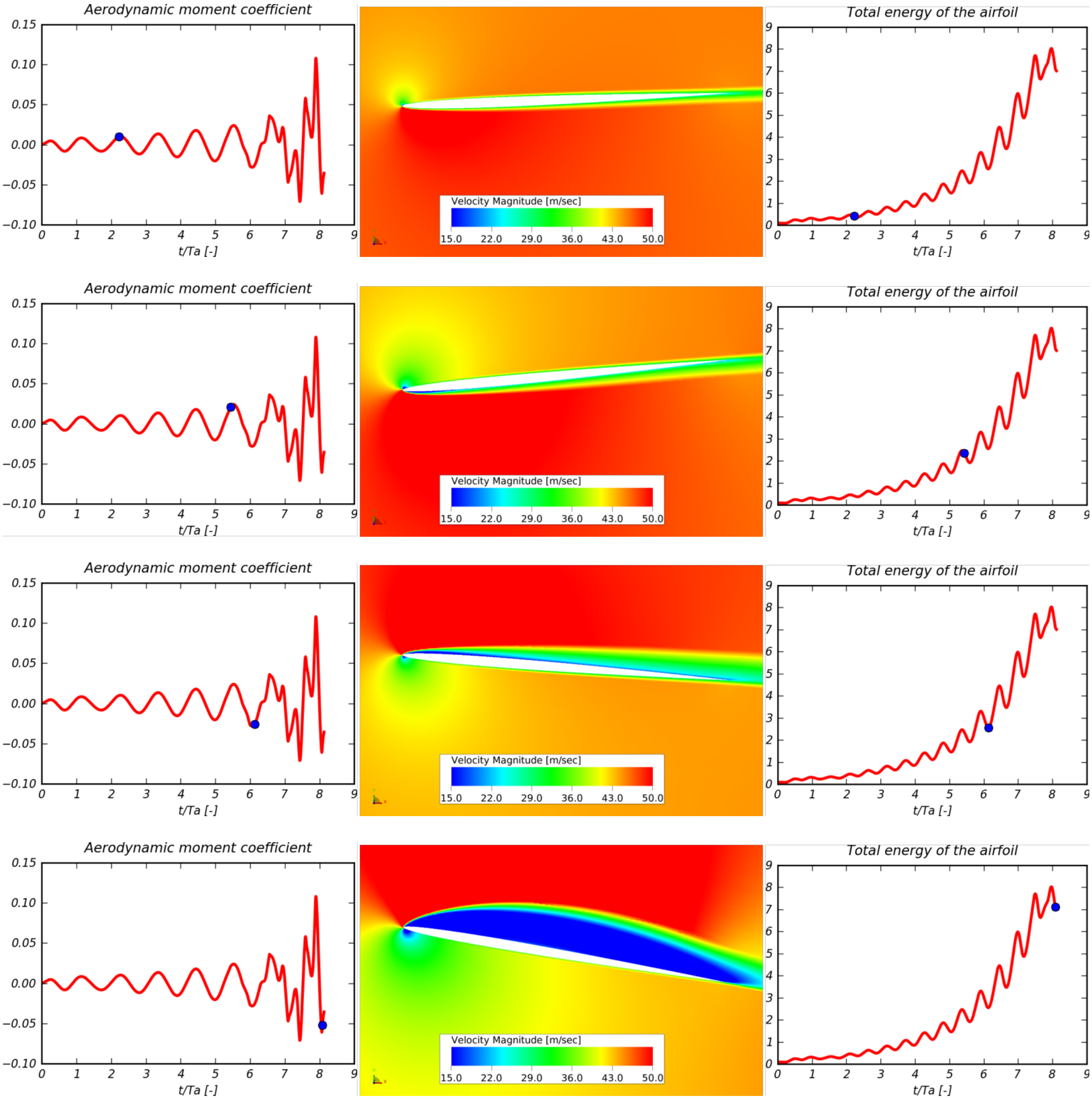


Figure 6.12: Time-histories of the aerodynamic moment (left pictures) and total energy of the structure (right pictures), as well as flow-field snapshots for four time instants of the $U_\infty = 1.1U_{flt}$ test case.

by PowerFLOW® compared to the Theodorsen model when the flow behaviour starts to get affected by viscous phenomena.

Given the previous consideration, the flow-field visualizations of the CFD simulation in fig.6.12 can be considered to explain the physical phenomena that the Theodorsen model can not account for, and that play a crucial role in the aeroelastic behaviour of the system for large amplitudes of motion. For example, the top figures show a flow dynamic behaviour similar to the one already observed in fig.6.10 for the below flutter condition with a smooth, attached flow field. However, when the total energy of the structure increases given the above flutter condition considered, in the second series of pictures from the top, LE-stall can be noticed in the blue area in the lower surface of the airfoil near its LE. This can be expected given the very thin LE of the NACA 0003 profile employed, which leads to stalled conditions starting from the airfoil LE at a relatively small value of the AOA (around 5.5°) and quickly propagating to the whole airfoil contour in about half of the pitch period T_α. The occurrence of LE-stall at around $t = 5.5T_\alpha$ can be also noticed in the plunge velocity and c_m plots of fig.6.11, with a loss of the smooth character of the oscillation, and less in the pitch velocity and lift ones. This is due to the fact that the onset of LE-stall is a localized phenomenon generating only a local change of the pressure and skin friction distributions. This plays a significant role in the moment coefficient, due to the role played by the distance from the pitching axis in the computation of this quantity, as well as in the plunge velocity, given the interaction of the separated region with the flow blowing/suction employed to account for the airfoil plunging motion. For $t > 5.5T_\alpha$, the values of the airfoil pitching and plunging displacements gets large enough to overcome the static stall value for this airfoil ($\alpha \approx 5°$), and dynamic-stall conditions are very likely to be reached, as the last set of plots in fig.6.12 might suggests. Furthermore, very large and abrupt oscillations in the moment coefficient are obtained and the flow-field around the airfoil upper surface is massively stalled for a value of the AoA $\alpha \approx 10°$.

All in all, it can be stated that the coupling methodology is capable of predicting the unstable behaviour of a 2 DoFs aeroelastic airfoil beyond its flutter boundary. For a more in depth analysis of the behaviour of the methodology under stalled conditions and large displacements, different benchmark solutions should be employed, necessarily experimental results or other CFD-MBD aeroelastic couplings considering airfoil dynamic stall.

6.3.3. Flutter velocity prediction

The second part of the investigation for the 2 DoFs PowerFLOW®-Simpack® coupling is related to the prediction of the flutter boundary with an accuracy of 1% by comparing the results of the two aeroelastic fully-coupled simulations (test #3 at $U_\infty = 99\%U_{flt}$ and test #4 at $U_\infty = 101\%U_{flt}$) against those returned by the corresponding reference numerical solutions. Such a comparison is shown in fig.6.13 for test case #3 and in fig.6.14 for test case #4. Besides, both simulations are run for a longer duration of $12T_\alpha$ for the purpose of a better evaluation of the converging/diverging character of the solution given the closeness to the flutter boundary. In both test cases, it possible to affirm that the

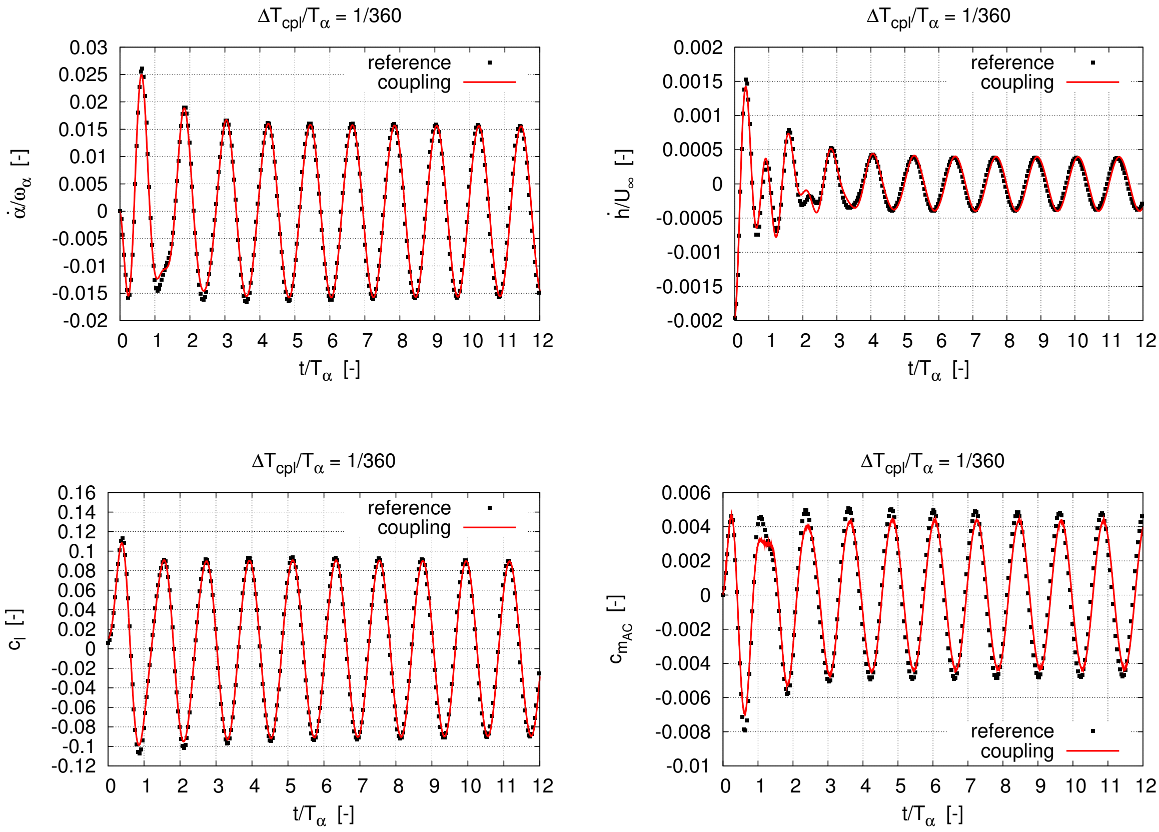


Figure 6.13: Dimensionless time-histories of the kinematic and aerodynamic quantities obtained by the 2 DoFs PowerFLOW®-Simpack® coupling with LRF (red line) against the reference analytical solution (black squares) for an inflow condition 99% the flutter velocity and coupling timestep 0.0005s.

coupling methodology is able to capture the converging character of the response for a conditions slightly below the flutter boundary, as well as the diverging character for a condition slightly above. Also for this investigation, the results reported in this section are related to the plunge/pitch velocity as well as aerodynamic moment and lift, while the pitch/plunge position and acceleration can be found in the appendix fig.F.20 and fig.F.21.

If the diverging character of the solution above flutter is clearly visible in the time-histories of all the aeroelastic variables considered, the same cannot be stated for the converging behaviour of the below flutter test case, where oscillations are rather weakly damped in their statistically-converged phase of the response after the initial transient. This might be due to the too short simulation duration considered, although a larger simulation duration would have been more expensive and could still potentially not highlight a clear convergent trend. For this reason, the time-histories of the total energy of the structure for these two cases, shown in fig.6.15, can help due to the combination of first and zero-th derivatives of both kinematic states. This is actually the case for the below flutter condition, where a clear reduction of the structural total energy can be visualized over the 12 periods of pitch oscillation considered. A closer look to these plots reveals that this quantity is well predicted after the initial transient (where plunge oscillations with a characteristic time below the flow-pass time might not be well captured by the wall-velocity method) in the case of the below flutter condition. While in the above flutter condition, it seems that the coupling is not able to reach the energy prediction of the reference solution. This is a similar behaviour as the one observed also in sec.6.1.4 where the oscillations of the kinematic states returned by the coupling are smaller than those of the reference solution.

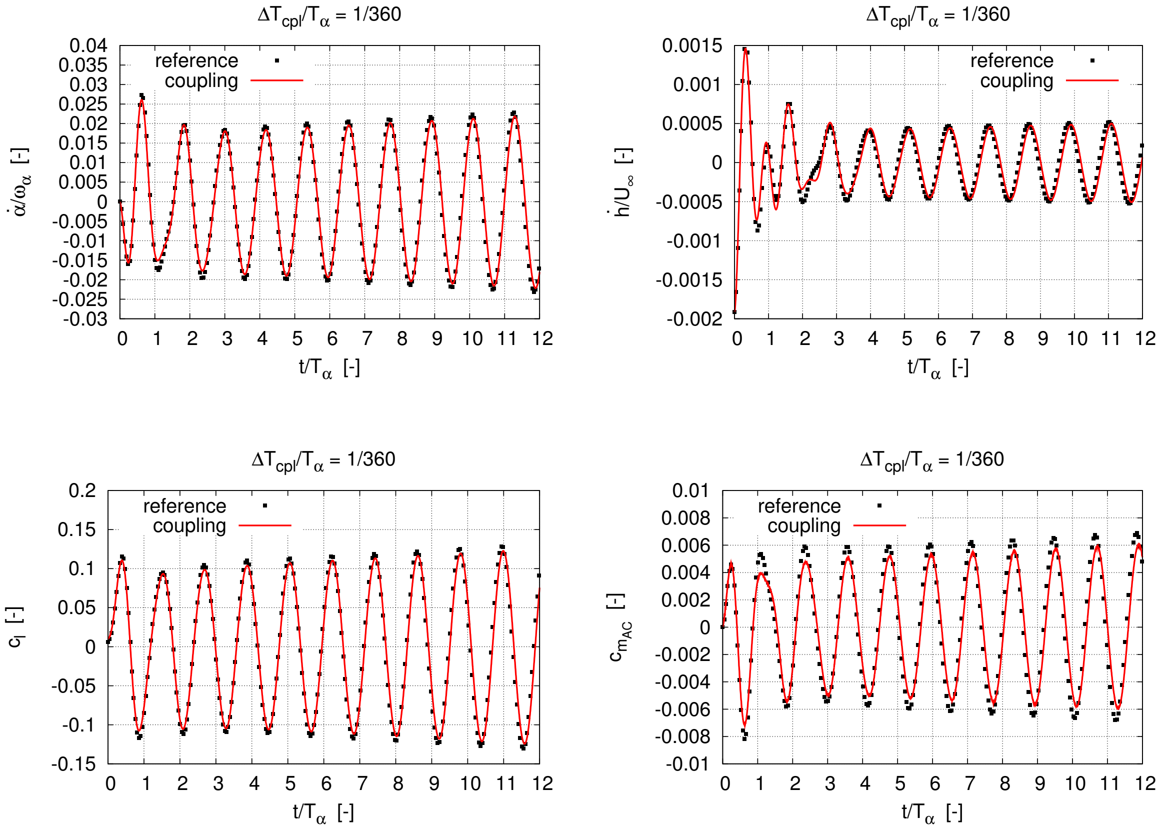


Figure 6.14: Dimensionless time-histories of the kinematic and aerodynamic quantities obtained by the 2 DoFs PowerFLOW®-Simpack® coupling with LRF (red line) against the reference analytical solution (black squares) for an inflow condition 101% the flutter velocity and coupling timestep 0.0005s.

Coming back to fig.6.13 and fig.6.14, both test cases show the same mismatches between the coupling solution and the reference one in areas of the response characterized by characteristic time-scales less than the flow-pass time, as observed also for test #1. This is primarily the case of the plunge kinematic states in their initial transient, and it is related to the poor match visible also in the initial transient of the total structural energy, as well as in the pitching moment during the first two T_α periods of the response. Another observation is that the aerodynamic moment coefficient is affected by a much larger amplitude deficit in the statistically-converged part of the motion after the initial transient than all the other quantities, besides also a small phase delay can be noticed in the last periods of the response. Similar observations can be made also in the c_m plot of test #1 of fig.6.9 and they might be explained by considering the c_m plots both in sec.4.4 for the LRF-pitch motion and in sec.4.5 for the plunge-wall-velocity. In the c_m plots of the aerodynamic validation chapter, a similar phase and amplitude mismatch between the Theodorsen solution and the PowerFLOW® airload is apparent during both the upstroke and down-stroke phases of the prescribed pitch/plunge motions, and this occurs for a range of values of α, h/c and $\dot{h}/U_\infty$ compatible with the last test cases of the 2 DoFs CFD-MBD coupling.

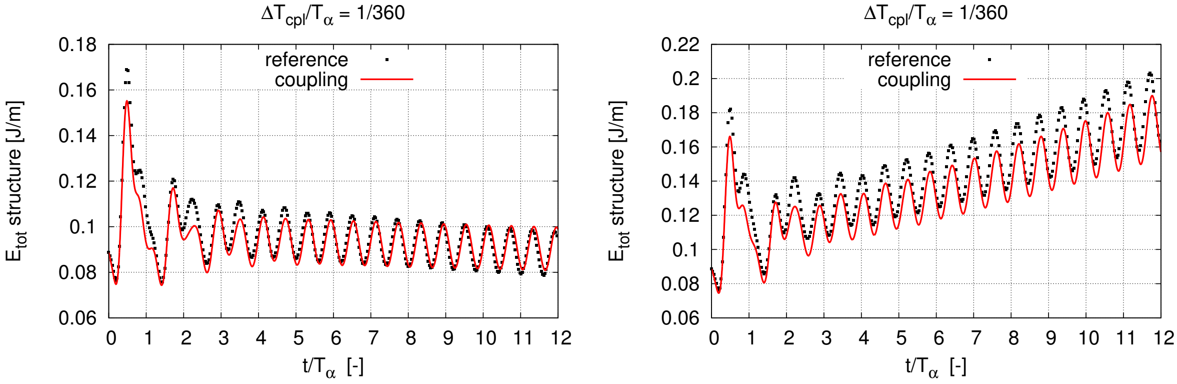


Figure 6.15: Time-histories of the total structural energy of the system for conditions slightly below (-1% on the left) and above (+1% on the right) the flutter velocity prediction returned by the V-g method in the case of the 2 DoFs PowerFLOW®-Simpack® coupling with LRF (red line) against the reference analytical solution (black squares) with a coupling timestep of 0.0005s.

Concluding, it could be argued that the coupling methodology developed is able to predict the flutter boundary (U_{flt}) of a 2 DoFs semirigid airfoil system with lumped parameters under a tolerance of $1\%U_{flt}$. This is cannot be clearly noticed for the below-flutter condition due to the very low convergent behaviour of its kinematic states and airload components, but it is apparent from the total structural energy. In any case, this investigation shows that the coupling between PowerFLOW® and Simpack® with pitching motion modelled by a truly rotating geometry has been successfully applied for the prediction of a fundamental property of aeroelastic systems

7
Conclusions and Recommendations

In this thesis, a FSI methodology to couple the LBM-based CFD solver PowerFLOW® and the MBD simulation tool Simpack® is discussed. The goal of such a CFD-MBD coupling is to predict the aeroelastic behaviour of airfoil systems featuring lumped structural parameters and undergoing free plunge and prescribed pitch displacements (1 DoF coupling), as well as free plunge and pitch motions (2 DoFs coupling). Furthermore, for the latter problem, predictions of its flutter-onset behaviour are also attempted. These objectives are pursued by employing a sequential approach, where the simulation-setup of each separate domain, the coupling algorithm and the final CFD-MBD coupling are carefully verified by comparisons with analytical models based on time-domain EoMs for the structural domain, unsteady aerodynamic theories for the fluid domain and a combination of both in the case of the reference aeroelastic solution to assess the coupling results.

7.1. Conclusions

From the four milestones of this sequential approach introduced in sec.1.2 and corresponding to the four core chapters of this thesis report, the main conclusions of this MSc research project can be summarized as it follows:

1. The outcome of ch.3, namely the verification of the Simpack® setup, is that the MBD simulation setup developed is able to compute with a high level of accuracy kinematic states from analytical aeroelastic models for a 2 DoFs airfoil problem. Besides, this activity returns a method to apply externally calculated time-histories of aerodynamic forces to the MBD simulation, as well as knowledge and experience with the tool. This verification is carried out by prescribing aerodynamic forces from analytical models (steady Glauert, quasi-steady theory and time-domain approximated Theodorsen model) in the Simpack® simulation for the computation of kinematic states to compare with those given by complete analytical aeroelastic systems featuring the same aerodynamic models.

2. Regarding ch.4, which deals with the verification of the PowerFLOW® CFD setup, the following outcomes are worth reporting:

   (a) The pitch-LRF approach returns very accurate computations of the Theodorsen lift signal for moderately up to highly unsteady flows and pitch amplitudes below 2°, while regarding the aerodynamic moment, small phase and amplitude mismatches are observed for the same motion conditions.

   (b) For the transpiration-velocity plunge motion, after an amplitude tuning of the method, a reasonably good match in the c_l is obtained, although a phase mismatch increasing with the reduced frequency, but a less accurate c_m is obtained with an error up to 50% for the same range of reduced frequencies and plunge amplitudes below 5% of the chord.

   (c) In the pitch body-force method, by firstly amplitude tuning and phase shifting the model, a decent match of the c_l is obtained, although this degrades rapidly for increasing reduced frequency and amplitude with a non-constant phase difference compared with the LRF-signal.

   Globally, it could be argued that the PowerFLOW® lift matches well the Theodorsen model for a truly pitching airfoil, while for the pitch-body-force and plunge-wall-velocity motions this occurs only for the smallest amplitudes and reduced frequencies and after tuning the models. The moment comparison is instead less favourable for both true and virtual motion cases. In these CFD simulations, the airload is computed for attached, incompressible and moderate Reynolds-number flow conditions on a NACA 0003 airfoil.

3. In ch.5, an initial coupling methodology between Simpack® and a low-order aerodynamic routine (convolution-integral formulation of the Theodorsen model) acting as surrogate of PowerFLOW® returns a very good prediction of the analytical aeroelastic variables of both the 1 and 2 DoFs aeroelastic problems. This is obtained if a sufficiently small coupling timestep (around 200-400 fractions of a characteristic period of motion) is employed due to the underlying explicit FSI algorithm with subcycling. In the coupling, the exchange of fluid and structural variables is carried out by means of file input/output and look/up tables, in addition to the choice of running one Simpack® simulation per each coupling iteration. Such a successfully verified coupling methodology is then extended to achieve the coupling between PowerFLOW® and Simpack® by implementing additional features to deal with the complexity of the CFD solver, such as initial transient handling, binary result file generation and conversion, input table reading and body motion reconstruction.

4. The final verification of the PowerFLOW®-Simpack® coupling, in ch.6, on the 1 DoF problem for different amplitude and reduced frequency of motion conditions and on the 2 DoFs one for test cases at ±10% and ±1% the flutter velocity prediction of the V-g method leads to the following outcomes:

   (a) For the 1 DoF problem with pitch-LRF, a quite good prediction of the lift force is obtained with limited amplitude mismatches, while the reference kinematic states are not as accurately predicted in amplitude, especially for increasing amplitude and reduced frequency. This could be due to both inaccuracies in the CFD prediction of the lift force (especially in the contribution due to the wall velocity BC) or in the implementation/derivation of the reference solution based on the hybrid time-frequency domain Theodorsen model.

   (b) The 1 DoF problem with pitch simulated by body forces, provides aeroelastic results much more similar to the LRF coupling than the reference solution, with small amplitude deviations in the statistically converged part of the responses for the largest reduced frequency, as observed also in the aerodynamic setup verification activity, and in the initial transient due to the different way the LRF and body forces approaches simulate the same pitch velocity.

   (c) Regarding the 2 DoFs coupling implementation, in the case of the 90%, 99% and 101% flutter velocity conditions, a good match for the lift and pitch states is obtained, and, more importantly, the converging and diverging behaviour in time of the total structural energy is correctly captured for conditions 1% above and below the flutter boundary. However, in the plunge states and aerodynamic moment, some deviations in areas of the response characterized by oscillations with time-scales below the flow-pass time are observed, most likely due to limitations in the transpiration velocity approach. In the test case at 110% the flutter boundary, the coupling solution is in good agreement with the reference one, based on the lag-states approximation of the Theodorsen model, for small displacements. When displacements get larger, flow visualization snapshots reveal a flow-behaviour affected by LE and dynamic stall, thus the analytical solution cannot be used anymore as a reference.

On the whole, the coupling between PowerFLOW® and Simpack® can return accurate predictions of aeroelastic quantities for attached, incompressible and moderate Reynolds number applications when moderate values of amplitude and reduced frequency of motion, as well as oscillations with a characteristic time larger than the flow-pass time, are considered. Besides, this requires also the tuning of the models (wall velcity BC and body forces) employed in the CFD setup to emulate the effects true plunging and pitching motions.

7.2. Recommendations for further work

Based on the previously discussed results and conclusions that be drawn for each milestone of the thesis research, the following recommendations and suggestions for the continuation of the work can be outlined:

- Regarding the tuning of the body forces and wall velocity models, it is observed that a tuning procedure taking into account both the lift and aerodynamic moment can hardly replicate both the signals returned by the Theodorsen model with a similar level of accuracy, primarily due to a phase lag. Therefore, an equally good prediction of both signals is likely to be achieved only by employing more sophisticated models than those employed in this thesis. This deals with the analytical formulation of both the wall velocity and body forces and a different shape of the volume where body forces are applied.

- The less accurate prediction of the c_m, compared to the c_l in the case of the pitch-LRF simulations, might be due to deficiencies in the aerodynamic setup. The latter could be still improved, for example, a better boundary layer modelling (by using a laminar or a fully turbulent wall model on different portions of the airfoil contour) can improve the skin friction contribution to the moment coefficient.

- Concerning the aerodynamic verification activity of the PowerFLOW® setup, it would be interesting to consider also other motion conditions, for example smaller values of reduced frequency or larger values of amplitude to further understand the behaviour of the wall velocity BC and body force methods, as typical rotorcraft (HART-II benchmark [16] and [15]) and wind-turbine blade applications ([10]) cover larger ranges of both quantities than those considered.

- Once a wider range of motion conditions is explored for the body-force and wall-velocity BC methods, it could be useful for a 3D application of the coupling methodology to extract reduced order models for the tuning parameters. This will make the coupling methodology easier to automatize and to be employed by external users. In addition, also different airfoil models can be tested (for example cambered airfoils), now that the coupling has been validated for elementary flow conditions and geometries.

- Regarding the coupling methodology, its robustness and efficiency could be still improved by avoiding the use of the file input/output medium, at least from the Simpack® side. As the MBD tool is relatively easy to couple via memory sharing and TCP/IP interfaces, a co-simulation engine between Simpack® and the Coupling Interface could also be implemented with the main goal of generating an input table for the PowerFLOW® simulator at each coupling timestep.

- As the long-term goal of the project is the development of a coupling methodology for 3D applications to rotating flexible blades, it is probably more valuable to invest efforts in the development of a 3D version of the coupling methodology than in improving the efficiency of the current one limited to 2D airfoils. However, developing a 3D coupling between PowerFLOW® and Simpack® requires a new Simpack® elastic blade setup (for example by employing the non linear version of the SIMBEAM tool [65]) and the extension of the Coupling Interface to handle the exchange of fluid and structure data on a 3D geometry. For the CFD setup, the work done in [6] is a valuable starting point. The extension of the Coupling Interface is expected to be more challenging to implement than the Simpack® elastic blade setup, as proper models for the matching of the two fluid and structure surface meshes are required, as briefly introduced in sec.2.1.3. This aspect has to be addressed even if the Simpack® blade would be split in a much smaller number of surface elements than in the case of typical FE discretizations. Eventually, a 3D aeroelastic coupling will be much more computationally expensive due to the CFD block. Therefore, this will push also for an improvement of the efficiency of the coupling mechanism, by focusing on what mentioned in the previous point. This could be carried out in parallel or at a later stage, but it is essential for a successful industrial application of the methodology.

A

Derivation of the body-forces expressions

The expressions for the body forces shown in eq.2.10 were obtained in [6] by considering approximate formulations of the rigid-body acceleration vectors of the airfoil at the LE and TE locations. In this section, the derivation of such acceleration terms is carried out. The starting point is the Charles-Mozzi theorem which provides an expression for the velocity vector of a point in a rigid body undergoing pure pitch motion. The time derivative of such expression leads to following formulation:

$$\boldsymbol{v}_P = \boldsymbol{\omega} \times \boldsymbol{r} \qquad \rightarrow \qquad \boldsymbol{a}_P = \dot{\boldsymbol{\omega}} \times \boldsymbol{r} + \boldsymbol{\omega} \times \dot{\boldsymbol{r}} \tag{A.1}$$

where $\boldsymbol{r}$ is the position vector of a generic point P with respect to the pitching axis and $\boldsymbol{\omega}$ is the angular velocity of the body. By recalling the fact that $\dot{\boldsymbol{r}} = \boldsymbol{v}_P$ and by applying the "bac-cab" vector identity, the above expression can be further worked-out as shown below.

$$\boldsymbol{a}_P = \dot{\boldsymbol{\omega}} \times \boldsymbol{r} + \boldsymbol{\omega}(\boldsymbol{\omega} \cdot \boldsymbol{r}) - \boldsymbol{r}(\boldsymbol{\omega} \cdot \boldsymbol{\omega}) \tag{A.2}$$

In the above equation, the second term cancels out, as the angular velocity vector $\boldsymbol{\omega}$ is perpendicular to the in-plain position vector $\boldsymbol{r}$, while the first term and second term can be computed by decomposing the vectors along a coordinate system of x- and y-axes. The coordinate system employed features a horizontal unitary direction-vector $\hat{\boldsymbol{x}}$ positive towards the TE, a vertical unitary direction-vector $\hat{\boldsymbol{y}}$ positive upwards and an angular-velocity vector oriented such that the value of the angular velocity is positive when the airfoil is pitching-up. This leads to the following full expression for the acceleration vector of a generic point of a rigid body undergoing a pure pitch motion, where the terms x and y refer to the horizontal and vertical coordinates of the position vector $\boldsymbol{r}$ of point P.

$$\boldsymbol{a}_P = (+\dot{\boldsymbol{\omega}} y - \boldsymbol{\omega}^{\mathbf{2}} x)\hat{\boldsymbol{x}} + (-\dot{\boldsymbol{\omega}} x - \boldsymbol{\omega}^{\mathbf{2}} y)\hat{\boldsymbol{y}} \tag{A.3}$$

At this point, the model is simplified by considering that for slender bodies - for instance the NACA 0003 airfoil geometry employed in ch.4 and ch.6 - the x-coordinate of point P is much larger than the y-one. This returns to following approximate expression for the acceleration vector.

$$\boldsymbol{a}_P \approx -\omega^2 x \hat{\boldsymbol{x}} - \dot{\omega} x \hat{\boldsymbol{y}} \tag{A.4}$$

Finally, by considering a pitching axis at the location of the quarter-chord and, if the above approximate expression is evaluated at the location of the LE and TE, the above expressions for the acceleration vectors employed to define the body forces $\boldsymbol{b_{LE}}$ and $\boldsymbol{b_{TE}}$ in eq.2.10 are obtained.

$$\boldsymbol{a}_{LE} = -\omega^2 \frac{c}{4} \hat{\boldsymbol{x}} - \dot{\omega} \frac{c}{4} \hat{\boldsymbol{y}} \qquad \text{and} \qquad \boldsymbol{a}_{TE} = \omega^2 \frac{3}{4} c \hat{\boldsymbol{x}} + \dot{\omega} \frac{3}{4} c \hat{\boldsymbol{y}} \tag{A.5}$$

B

Time-domain approximation methods of the Theodorsen model for aeroelastic systems

The frequency-domain character of the Theodorsen model allows to compute elegant closed-form solutions for unsteady flows which are highly appreciated for flutter investigation [83]. However, this model can also be inconvenient for the design of control mechanisms and the calculation time-domain solutions for aeroelastic problems where the reduced frequency of the phenomenon might be unknown beforehand, for instance due to couplings between eigenmodes or the absence of a periodic forcing [91]. This is the case of the problem investigated in ch.3 consisting in the calculation of the time response of the system due to an initial perturbation velocity.

In this appendix chapter, approximation methods of the Theodorsen function based on partial fractions are first described in sec.B.1, then the analytical formulation of the convolution integral approximation of the Theodorsen circulatory lift for a generic dynamic input is derived in sec.B.2, while in sec.B.3 the convolution integral is discretized for the specific input given in sec.3.1.3. Next, the aerodynamic analytical lag-states formulation of the Theodorsen circulatory lift is also derived in sec.B.4. Finally, in sec.B.5, the full time-domain equations for the aerodynamic force vector according to the two previous formulations are given.

B.1. Approximations of the Theodorsen function

A frequently employed solution to overcome the frequency-character of the Theodorsen model consists in approximating the generalized Theodorsen function $C(k)$, shown in eq.2.29, in terms of a summation of rational functions, up to a certain order n, in the domain of the complex Laplace variable $\bar{s}$ [92]:

$$C(k) \approx a_0 + \sum_{i=1}^{n} \frac{a_i}{\bar{s} + b_i} \tag{B.1}$$

The partial fractions $a_i/(\bar{s} + b_i)$ above are called lag terms, as they act as transfer function introducing a delay between the input and the output [92], which is what unsteady aerodynamic effects generally introduce. Besides, each of these partial fraction is associated to an aerodynamic state and the higher is the number of these states, the more accurate is the approximation of $C(k)$, which contains an infinite number of these states. The determination of the coefficients in the partial fractions is the result of a complex optimization procedure, such as the one described by [91], where it is shown that approximations of order (number of lag-states) larger than 4 overlap almost perfectly the true Theodorsen function.

In literature, several rational approximations of the Theodorsen function can be found, as those described in [83]. For the sake of the investigation in ch.3, the low-order two-zero and two-pole Padè approximation is considered:

$$\tilde{C}(k) = \frac{1}{2}\frac{(jk + 0.135)(jk + 0.651)}{(jk + 0.0965)(jk + 0.4555)} \quad \rightarrow \quad \tilde{C}(\bar{s}) = \frac{1}{2}\left(1 + \frac{0.059}{\bar{s} + 0.0965} + \frac{0.175}{\bar{s} + 0.4555}\right) \tag{B.2}$$

As shown in fig.B.1, the Padè approximation mimics very well the behaviour of the Theodorsen function, especially in the highly unsteady range for $k > 0.2$, in spite of its simplicity and the only two (out of

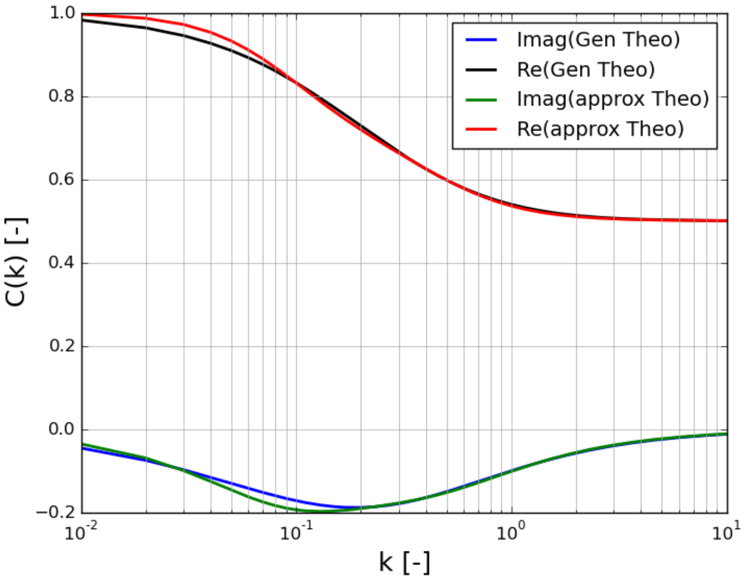


Figure B.1: Comparison between the real "Re()" and imaginary "Imag()" parts of the generalized Theodorsen function "Gen Theo" defined in eq.2.29 and the its Padè approximation "approx Theo" defined in eq.B.2.

the potential infinite number) aerodynamic lag-states considered. In the above equation, the theorem of analytical continuation can be exploited to extend the approximate Theodorsen function from the imaginary axis to the whole complex plain. This allows to express $\tilde{C}(k)$ as $\tilde{C}(\bar{s})$ with the reduced frequency k replaced by the complex variable $\bar{s} = \bar{\alpha} + jk$.

With the above approximation for the Theodorsen function in eq.B.2, it is possible to obtain a pure time-domain formulation for the lift and aerodynamic moment for the Theodorsen model shown in eq.2.28 to compute the time-domain solution of aeroelastic systems. This can be done by employing two different strategies. The first method leads to a convolution integral, while the second one leads to a new second-order ODE describing the time evolution of the only two aerodynamic lag-states retained by the Padè approximation. At the analytical level, there is no difference between the two implementations of the Padè-approximated Theodorsen model, as they are two different ways of transforming the same frequency-dependent expression in the time domain. The only differences appear numerically, but they can be reduced as the integration timestep gets smaller, up to convergence to the same solution for an integration timestep tending to zero.

B.2. Derivation of the convolution integral formulation

To get a convolution integral from the Theodorsen model featuring the Padè-approximated Theodorsen function in eq.B.2, the first step consists in applying the inverse of the Laplace transformation to only $\tilde{C}(\bar{s})$:

$$T(\tau) = \mathcal{L}^{-1}[\tilde{C}(\bar{s})] = \frac{1}{2}\Big(\delta(\tau) + 0.059e^{-0.0965\tau} + 0.175e^{-0.4555\tau}\Big) \tag{B.3}$$

The above equation shows that, by applying the inverse of the Laplace transformation, it is possible to convert the Theodorsen function $C(k)$ from the frequency to the time[1] domain obtaining the response to an impulse. However, since the Delta-Dirac function mathematically modelling the impulse is not the most convenient analytical object to deal with, by performing time integration the function $\delta(\tau)$ can be removed from the model obtaining the time-response to a step, which is what the Wagner function $W(\tau)$ shown below does. Therefore, the time integration of $T(\tau)$ returns the Padè approximation of the Wagner function:

$$\int_0^\tau T(\tau)d\tau = W(\tau) = \frac{1}{2}\Big(1.9956 - 0.6114e^{-0.0965\tau} - 0.3842e^{-0.4555\tau}\Big) \tag{B.4}$$

[1] The time τ is a dimensionless time related to the reduced frequency variable k and its relation to the physical dimensional time t is $t\omega = k\tau$, for a generic angular frequency ω.

The Padè approximation of the Wagner function is only one of the many options available in literature, another one very popular is the Jones function [7].

Due to the parallelism between the Wagner function and the Theodorsen function (shown by both Garrick [93] and Jones [94]) in modelling the circulatory lift over an airfoil, it is possible to extend eq.B.4 from the Wagner model to get the circulatory lift response to a generic input related to a structural dynamic motion, and not just a step function. This is related to the fact that $T(\tau)$ is only related to $\widetilde{C}(k)$. Therefore, if the whole circulatory lift expression of the Theodorsen model in eq.3.18 is considered, and time integration is performed, an approximate time-expression for $\tilde{L}_c$ can be obtained as the time response to the quasi-steady aerodynamic input $v_n^{(3/4)}(t)$ evaluated at the neutral point (three-quarter chord) location[2]. This leads to the convolution integral shown below, which is the mathematical element modeling the memory effect of the shed vorticity:

$$L_c \approx \tilde{L}_c = 2\pi\rho U b \int_0^t T(\hat{\tau}) v_n^{(3/4)}(\hat{\tau})\, d\hat{\tau} = 2\pi\rho U b \left[W(t) v_n^{(3/4)}(t{=}0) + \int_0^t W(t-\hat{\tau}) \dot{v}_n^{(3/4)}(\hat{\tau})\, d\hat{\tau} \right] \tag{B.5}$$

B.3. Discretization of the convolution integral

As mentioned in sec.3.2.1, where the forward/explicit Euler method is employed to discretize in time the Theodorsen problem formulated in terms of the convolution integral approach, the following discrete system is obtained:

$$\boldsymbol{z}_{n+1} = (\boldsymbol{I} - \boldsymbol{\tilde{M}^*_{tot}}^{-1} \boldsymbol{\hat{K}^*_{tot}} \Delta\hat{t}\,)\boldsymbol{z}_n + \boldsymbol{\tilde{M}^*_{tot}}^{-1} \boldsymbol{F}^*(\hat{t}_n)\Delta\hat{t}$$

where the 4x1 vector $\boldsymbol{F}^*(\hat{t}_n)$ contains the 2x1 force vector $\boldsymbol{f}^*(\hat{t}_n)$ containing convolution integral:

$$\boldsymbol{f}^*(\hat{t}_n) = -\frac{\hat{U}}{a} \begin{bmatrix} \left(\dot{h} + \dot{\alpha} + \hat{U}\alpha\right)\big|_{\hat{t}_0} \hat{W}(\hat{t}_n) + \int_0^{\hat{t}_n} \hat{W}(\hat{t}_n - \hat{\tau})\left(\ddot{h} + \ddot{\alpha} + \hat{U}\dot{\alpha}\right) d\hat{\tau} \\ \xi_A \left\{ \left(\dot{h} + \dot{\alpha} + \hat{U}\alpha\right)\big|_{\hat{t}_0} \hat{W}(\hat{t}_n) + \int_0^{\hat{t}_n} \hat{W}(\hat{t}_n - \hat{\tau})\left(\ddot{h} + \ddot{\alpha} + \hat{U}\dot{\alpha}\right) d\hat{\tau} \right\} \end{bmatrix}$$

Given the explicit Euler method, the discretization of the convolution integral can be easily carried out by applying the left-rectangle quadrature rule, also called left Riemann sum. According to this method, within every single discrete integration interval (rectangle), the function is sampled and approximated by its value at the left basis of the rectangle, while the horizontal height can be either uniform or change from an interval to the other depending on the uniform or not discretization of the time axis.

Based on this framework, a general convolution integral can be discretized - for generic function $k(x)$ and $f(y)$ with $T = y_{k+1} - y_k$ - as:

$$\int_0^x k(x-y) f(y)\, dy \;=\; \int_0^{x_n} k(x_n - y) f(y)\, dy \;\approx\; (y_{k+1} - y_k) \sum_{k=0}^{n-1} k(x_n - y_k) f(y_k) \;=\; T \sum_{k=0}^{n-1} k_{n-k} f_k \tag{B.6}$$

When the above result is applied to the convolution integral of interest, the following discrete force vector $\boldsymbol{f}^*(\hat{t}_n)$ is obtained:

$$\boldsymbol{f}^*(\hat{t}_n) = -\frac{\hat{U}}{a} \begin{bmatrix} \left[\dot{h}(\hat{t}_0) + \dot{\alpha}(\hat{t}_0) + \hat{U}\alpha(\hat{t}_0)\right] \hat{W}(\hat{t}_n) + \Delta\hat{t} \sum_{k=0}^{n-1} \hat{W}(\hat{t}_n - \hat{t}_k)\left[\ddot{h}(\hat{t}_k) + \ddot{\alpha}(\hat{t}_k) + \hat{U}\dot{\alpha}(\hat{t}_k)\right] \\ \xi_A \left\{ \left[\dot{h}(\hat{t}_0) + \dot{\alpha}(\hat{t}_0) + \hat{U}\alpha(\hat{t}_0)\right] \hat{W}(\hat{t}_n) + \Delta\hat{t} \sum_{k=0}^{n-1} \hat{W}(\hat{t}_n - \hat{t}_k)\left[\ddot{h}(\hat{t}_k) + \ddot{\alpha}(\hat{t}_k) + \hat{U}\dot{\alpha}(\hat{t}_k)\right] \right\} \end{bmatrix}$$

The employment of the left Riemann sum is firstly convenient, as it does only require the value of the solution on the left side of the discrete integration interval. Other methods, like the midpoint rule, requires interpolation between the left and the right value, although this effort pays-off with a higher-order accuracy. Finally, the order of accuracy of the left-rectangle rule is $\mathcal{O}(\Delta\hat{t}\,)$, consistently with the explicit Euler method.

Alternatively, it is also possible to employ the right Riemann sum. However, this variant is more difficult to implement than the left sum method. For example, to get the first value of the convolution

[2] This quantity can also be seen as the effective angle of attack for a pitching and plunging airfoil evaluated at the neutral point, as discussed in sec.2.4.3.

integral $\Delta\hat{t}\, W(0)v_n^{*(3/4)}(\Delta\hat{t}\,)$, the computation of the $\dot{\boldsymbol{z}}_1$ at t_1 state-vector by applying 3.24 is required. Nevertheless, this can only be done after the calculation of $\boldsymbol{z}_1$ at t_1, which requires the knowledge of the first value of the convolution integral leading to an implicit formulation. Instead, the left method just requires $v_n^{*(3/4)}(0)$, which is easy to get as $\boldsymbol{z}_0$ at t_0 is prescribed by the initial conditions. With a little more complication in the discrete equation, the right method can be improved by the trapezoidal rule, which considers both the left and right values within each interval providing third-order accuracy.

B.4. Derivation of the aerodynamic lag-states formulation

The starting point to derive the approximate time-domain Theodorsen airload, according to the lag-states formulation, is a dimensionless expression $\tilde{C}^*(\hat{s})$ in terms of a ratio between a numerator and a denominator:

$$\tilde{C}^*(\hat{s}) = \frac{1}{2}\frac{(\hat{s}+n_1)(\hat{s}+n_2)}{(\hat{s}+d_1)(\hat{s}+d_2)} = \frac{1}{2}\frac{N(\hat{s})}{D(\hat{s})} \tag{B.7}$$

This is done because the goal is to get a dimensionless aeroelastic system, and it is possible to derive a dimensionless airload by starting from a dimensionless formulation of the approximate Theodorsen function in the frequency-domain of the variable $\hat{s} = \hat{\alpha} + j\hat{\omega}$ related to the dimensionless time $\hat{t}$. In this way, both the denominator and the numerator contain only products of the type $(\hat{s} + a_i)$ involving coefficients and the dimensionless Laplace variable $\hat{s}$.

Based on the scaling rules defined in eq.3.9, an approximate expression for the dimensionless lift in the case of a pitching axis $\tilde{a} = -1/2$, can be reconstructed in terms of $\tilde{C}^*(\hat{s})$, where the term $\dot{v}_n^{(1/2)}$ is the time derivative of the normal component of the relative velocity wind-airfoil evaluated at the middle of the airfoil:

$$L^* = \frac{1}{2a}\left(\ddot{h} + \frac{1}{2}\ddot{\alpha} + \hat{U}\dot{\alpha}\right) + \frac{\hat{U}}{a}\tilde{C}^*(\hat{s})\left(\dot{h} + \dot{\alpha} + \hat{U}\alpha\right) = \frac{1}{2a}\dot{v}_n^{*(1/2)}(\hat{t}\,) + \frac{\hat{U}}{a}\tilde{C}^*(\hat{s})v_n^{*(3/4)}(\hat{t}\,) \tag{B.8}$$

The transformation of the above equation from the time to the frequency domain of the complex variable $\hat{s}$ allows to split the previous equation into two separate equations in the Laplace variable.

$$\tilde{L}^* = \frac{1}{2a}\dot{\tilde{v}}_n^{*(1/2)}(\hat{s}) + \frac{\hat{U}}{a}N(\hat{s})\tilde{r}(\hat{s}) \qquad \text{and} \qquad D(\hat{s})\tilde{r}(\hat{s}) = \tilde{v}_n^{*(3/4)}(\hat{s})$$

The complex quantity $\tilde{r}$ plays the role of a lag transfer function, as described in sec.B.1 between the $\tilde{v}_n^{*(3/4)}$ input and the resulting lift. This leads to the introduction of two additional lag-states of aerodynamic nature into the aeroelastic systems, whose dynamics needs to be determined in the same way as the structural states. Therefore, by applying the inverse of the Laplace transformation to the two above equations, a second-order ODE is obtained describing the time evolution of the airload and involving structural (h, α) and aerodynamic lag-states $(r,\dot{r})$.

$$L^* = \frac{1}{2a}\left(\ddot{h} + \frac{1}{2}\ddot{\alpha} + \hat{U}\dot{\alpha}\right) + \frac{\hat{U}}{a}\left(\ddot{r} + \hat{U}(n_1+n_2)\dot{r} + \hat{U}^2 n_1 n_2 r\right)$$

$$\ddot{r} + \hat{U}(d_1+d_2)\dot{r} + \hat{U}^2 d_1 d_2 r = \left(\dot{h} + \dot{\alpha} + \hat{U}\alpha\right) \tag{B.9}$$

B.5. Computation of the time-dependent aerodynamic force vectors

For both the convolution-integral and the aerodynamic lag-states formulations of the time-domain approximated Theodorsen aeroelastic system, it is required to define the relations that allow to compute the two components of the dimensional aerodynamic force vector $\boldsymbol{F} = \left[-\hat{L},\, \hat{M}_A - x_A\tilde{L}\right]^T$ from the knowledge of the structural and - where required - computed aerodynamic lag-states. This is required for both the Simpack® verification activity discussed in ch.3 and for the coupling demonstrator discussed in ch.5.

B.5.1. Convolution-integral method

Given the expressions for the approximate lift and aerodynamic moment for a quarter chord pitching axis location ($\tilde{a}=-1/2$):

$$\hat{L} = \pi\rho b^2\left(\ddot{h}_O+\frac{1}{2}b\ddot{\alpha}+U_\infty\dot{\alpha}\right)+2\pi\rho U_\infty b\left[\left(\dot{h}_O+b\dot{\alpha}+U_\infty\alpha\right)\Big|_{t=0}W(t)+\int_0^t W(t-\tau)\left(\ddot{h}_O+b\ddot{\alpha}+U_\infty\dot{\alpha}\right)d\tau\right]$$

$$\hat{M}_A = -\pi\rho b^3\left(\frac{1}{2}\ddot{h}_O+\frac{3}{8}b\ddot{\alpha}+U_\infty\dot{\alpha}\right)$$

the following dimensional aerodynamic force vector is obtained, where the expression for the dimensional Padè approximation of the Wagner function $W(\tau)$ is shown in eq.B.4.

$$\boldsymbol{F} = -\pi\rho b^2\begin{bmatrix} 1 & \frac{b}{2} \\ \frac{b}{2}+x_A & \frac{3}{8}b^2+x_A\frac{b}{2}\end{bmatrix}\begin{bmatrix}\ddot{h}_O \\ \ddot{\alpha}\end{bmatrix} - \pi\rho b^2 U\begin{bmatrix}0 & 1 \\ 0 & b+x_A\end{bmatrix}\begin{bmatrix}\dot{h}_O \\ \dot{\alpha}\end{bmatrix}+$$

$$-2\pi\rho U_\infty b\begin{bmatrix}\left(\dot{h}_O+b\dot{\alpha}+U_\infty\alpha\right)\big|_{t=0}W(t)+\int_0^t W(t-\hat{\tau})\left(\ddot{h}+b\ddot{\alpha}+U_\infty\dot{\alpha}\right)d\hat{\tau} \\ x_A\left\{\left(\dot{h}_O+b\dot{\alpha}+U_\infty\alpha\right)\big|_{t=0}W(t)+\int_0^t W(t-\hat{\tau})\left(\ddot{h}+b\ddot{\alpha}+U_\infty\dot{\alpha}\right)d\hat{\tau}\right\}\end{bmatrix}$$

B.5.2. Aerodynamic lag-states method

In the case of the second method, the expression for the aerodynamic moment is the same as the one shown in the previous subsection, while the dimensional lift is different due to the presence of the lag-states r which do not have a physical dimension. The initial conditions for the two aerodynamic lag-states both are set to zero, as the airfoil starts moving from $t_0 = 0$ and no wake is present at that time. Therefore, the expression for the approximate lift is:

$$\hat{L} = \pi\rho b^2\left(\ddot{h}_O+\frac{1}{2}b\ddot{\alpha}+U_\infty\dot{\alpha}\right)+2\pi\rho bU_\infty\left(\frac{b\omega_\alpha}{2}\ddot{r}+\frac{U_\infty}{2}(n_1+n_2)\dot{r}+\frac{1}{2}\frac{U_\infty^2}{b\omega_\alpha}n_1n_2r\right)$$

This allows to derive the following dimensional force vector:

$$\boldsymbol{F} = -\pi\rho b\begin{bmatrix} b & \frac{b^2}{2} & U_\infty b\omega_\alpha \\ \frac{b^2}{2}+x_Ab & \frac{3}{8}b^3+\frac{b^2}{2}x_A & U_\infty b\omega_\alpha x_A\end{bmatrix}\begin{bmatrix}\ddot{h}_O \\ \ddot{\alpha} \\ \ddot{r}\end{bmatrix} - \pi\rho bU_\infty\begin{bmatrix}0 & b & U_\infty(n_1+n_2) \\ 0 & b^2+x_Ab & U_\infty(n_1+n_2)x_A\end{bmatrix}\begin{bmatrix}\dot{h}_O \\ \dot{\alpha} \\ \dot{r}\end{bmatrix}$$

$$-\pi\rho bU_\infty\begin{bmatrix}0 & 0 & \frac{U_\infty^2}{b\omega_\alpha}n_1n_2 \\ 0 & 0 & \frac{U_\infty^2}{b\omega_\alpha}n_1n_2x_A\end{bmatrix}\begin{bmatrix}h_O \\ \alpha \\ r\end{bmatrix}$$

In the above equation, the values for the zeros coefficients are $n_1 = 0.135$ and $n_2 = 0.651$, while those for the poles coefficients are $d_1 = 0.0965$ and $d_2 = 0.4555$.

C

Dimensionalization of the Simpack® simulation parameters

The scaling rules in eq.3.9 are applied in ch.3 to nondimensionalize the EoM of the aeroelastic systems. In addition, they can also be employed to find the corresponding dimensional aerodynamic and structural simulation parameters given the quantities in tab.3.1. This leads to:

$$\begin{aligned} h_O &= \tfrac{h}{b}; & t &= \tfrac{\hat{t}}{\omega_\alpha};\\ m &= a\rho b^2 c_{l_\alpha} = 1.5 b^2 2\pi; & U &= \hat{U} b \omega_\alpha\\ k_h &= \Omega^2 m \omega_\alpha^2 = 0.8^2 m \omega_\alpha^2; & k_\alpha &= r_\alpha^2 m b^2 \omega_\alpha^2 = 0.5_\alpha^2 m b^2 \omega_\alpha^2;\\ J_E &= r_\alpha^2 m b^2 = 0.5_\alpha^2 m b^2; & x_E &= \xi_E b;\\ x_A &= \xi_A b; & x_G &= \xi_G b \end{aligned}$$

As it is apparent from the above relations, to find values for the dimensional parameters on the lhs, some of the quantities on the rhs need to be prescribed. To begin with, by employing a value for the airfoil semichord $b = 1$[m] and for the pitch angular frequency $\omega_\alpha^2 = 1$[rad/s], a perfect equivalence between the dimensionless analytical problem and the dimensional Simpack® simulation (or a dimensional version of the aeroelastic analytical model) can be obtained in the values for time, kinematic states (α,h) and airfoil point coordinates. The only parameter that is still left is the value of the fluid flow density, which is required to determine the body mass, and so the structural properties. Flow density is set to the reference value of $\rho = 1.225\,[\mathrm{kg/m^3}]$ in ch.3. These assumptions lead to the following values for the dimensional simulation parameters to apply in the Simpack® setup and employed to get the results shown in sec.3.4:

$$\begin{aligned} &x_A = 0.0\,[\mathrm{m}]; && x_E = 0.1\,[\mathrm{m}]; && x_G = 0.5\,[\mathrm{m}]; && U = \hat{U}\,[\mathrm{m/s}];\\ &m = 11.545353\,[\mathrm{kg/m}]; && k_h = 7.389026\,[\mathrm{N/m^2}]; && k_\alpha = 2.886338\,[\mathrm{N/rad}]; && J_E = 2.886338\,[\mathrm{kg{\cdot}m}]; \end{aligned}$$

where $\hat{U} = 0.9\hat{U}_f$. Finally, due to the 3D nature of the Simpack® simulation, by employing a depth $d = 1$[m], dimensions can be adjusted to a 3D geometry by retaining the equivalence with the analytical and numerical 2D models.

D

Simpack®-PowerFLOW® Coupling Interface documentation

This appendix chapter serves as an extensive description of the Coupling Interface code developed to couple PowerFLOW® and Simpack® by means of a two-way explicit FSI scheme.

D.1. Development approach and essential aspects

In the development of a coupling strategy to couple Simpack® and PowerFLOW®, in addition to FSI aspects, properties of the solvers and physical/numerical modelling, also software issues need to be addressed. Regarding the latter, the most significant aspect is the need for a structured, easy to extend and to maintain computer program. Besides, this has to be combined with the two main functions required by the Coupling Interface, namely the handling of the ASCII input/output coupling files and the execution of the tools.

0-th level:		InterfaceManager	
1-st level:	TransientIterationManager		CouplingIterationManager
2-nd level:	[Coupling-files libraries]	[PowerFLOW® libraries]	[Simpack® libraries]
	tot_iter_listManager	init_trans_dataManager	AFSManager
	cpl_setupManager	PFinManager	SubvarManager
	cpl_forceManager	PFoutManager	SpckOManager
	kin_statesManager	InitTransChecker	SpckPostProcManager
3-rd level:	TriggerManager		ASCIIManager
4-th level:	Viewer	Model	FileHandler

Table D.1: Table showing the structure of the several libraries implemented for the PowerFLOW®-Simpack® coupling.

Among the plethora of possible software architecture models, the Model-Viewer-Controller method is employed, due to its simplicity, flexibility and adequacy for the two previously mentioned main functions of the Interface. As described in [95], the MVC is a software-design pattern commonly used to develop interfaces with a clear distinction between the data structure logic (Model), display logic (Viewer) and processing logic (Controller). In the context of the coupling, the Viewer logic is secondary, as it only relates to notify the user about the internal working of the Interface and it is addressed by the "Viewer" library in sec.D.2. On the other hand, the Model logic and Controller one are essential. Concerning the data structuring function, the Model logic is implemented by means of two python objects commonly used as data containers, namely lists and ordered dictionaries[1], while the Controller logic is the core of

[1] Lists are very general objects, as they can contain several different data-types (floats, doubles, strings etc.) but they do not allow to keep track of the nature of the information stored. For this purpose, dictionaries are more suitable, as they can contain an arbitrary number of lists each one labeled by means of a key. In this way, the user can quickly identify/define the nature of the data handled. An evolved version of standard dictionaries are ordered dictionaries, as they allow to prescribe an order in which keys are gathered.

the Interface, because the processing of fluid and structural data, as well as the triggering of the tools, are tasks carried out by this logic unit.

The data-processing function of the Controller logic is based on the tasks of reading, generating and writing the ASCII coupling files. If the reading and writing tasks can be made quite general, once common features between the files are identified, the generation process is rather file-specific. This suggests the adoption of a pyramidal structure consisting of a few hierarchical levels for the Controller logic of the Coupling Interface, as shown in tab.D.1. In this framework, the lowest (4-th) level is occupied by the "FileHandler" (see sec.D.2) library dedicated to the reading and writing of the ASCII files, as these tasks are rather general and depend only on the type of separator used (only comma, space and new-line separators are considered). Next, as ASCII files can generally be decomposed into headers, data and tail sections, the lower-intermediate 3-rd level is occupied by the "ASCIIManager" library (discussed in sec.D.3) which takes care of the splitting of these files into such three sections. Finally, the upper-intermediate 2-nd level of ASCII-file handling is occupied by the file-specific libraries described in sec.D.4. The latter contain functions specialized in extracting the necessary information from the datasections and in generating each coupling file. In these routines, the Model logic is exploited to generate the containers storing the data necessary to write the coupling files. Such data-containers consist of a main list with as many lists as lines in the file itself and each line-list stores as many items as columns in the file. Regarding the data-extraction function, ordered dictionaries collect the data from the each line of the datasection lists allowing to keep track of the nature of the information for processing and debugging reasons.

To implement the other main function of the Interface (execution of the tools) the Model logic and the Controller one work in unison with the first one providing the commands that are employed by the Controller logic to execute the tools. This is carried out by the "TriggerManager" library and the "SpckSimManager.sjs" script for the automatic handling of the MBD simulation. Besides, the Controller logic includes also the "InitTransChecker" library to process the PowerFLOW® force signals to assess the conclusion of the CFD initial transient. Finally, the higher 1-st and 0-th level libraries are described in sec.5.4.4, as they are related to the handling of the two phases of the aeroelastic simulation (initial transient and proper coupling) and to the control of the overall coupling mechanism.

D.2. Fourth (lowest) level libraries

At the lowest level of the Coupling Interface, there are three libraries implementing the lowest-level of the controller operations for file handling, as well as the Model and Viewer logic of the MVC architecture.

Viewer

This library consists in a set of several functions generating all the output messages that are displayed to the user and that are collected in the "simulator.o" file, which is a file automatically generated by the PowerFLOW® simulator during the simulation. The "simulator.o" file contains information on the case size (number of voxels, surfels and timesteps), simulation parameters (number of processors, solver precision, fluid solver etc), as well as the timestep of table import and the outputs from the command executed before importing the table. Therefore, this file can be exploited as an emulator of a GUI for the user to monitor the internal working of the Coupling Interface and of the Simpack® simulation in all their steps. The functions in this library return all the string messages that are printed in the "simulator.o" file by the libraries responsible for each task in the coupling or initial-transient mode of the Interface discussed in sec.5.4.4.

Model

The "Model" library is responsible for giving structure to the coupling data that are exchanged between the PowerFLOW®, Interface and Simpack® blocks before they are written to the coupling files by generating empty lists and ordered dictionaries based on the input keys and lists. Such data containers are returned by two functions that are directly called by all the second-level libraries whenever files are read or written by the "FileHandler" library. In addition, this library directly stores essential data for the working of the coupling mechanism, such as the full commands for the execution of PowerFLOW® CLI for the binary-to-ASCII conversion of the CFD output file ("<exaritool path> forces.ri PF_out.csnc -sec -units mks -moment") and for the execution of the Simpack® solver to run the MBD simulation ("<Simpack-slv path> –silent -s <Spck.spck path>"), as well as paths and headers of the coupling files.

FileHandler

This library takes care of reading and writing ASCII files based on the delimiter employed to separate the data items in the file itself. In python, these tasks can be carried out by importing the "csv" package and by using its "reader" and "writer" functions. In the "reader" function, the delimiter is an input, in addition to the path to the file of interest, while the output is an iterable object which is passed to the functions in the "ASCIIManager" library. On the other hand, the "writer" function requires as input also a list (initialized by the data-structuring functions of the "Model" library and filled with data by the 2-nd level libraries) containing the full content for the generation of the file. If the delimiter is a new-line symbol "\n", each line is treated as a single-string list, while, if the delimiter is a comma or space, then each line is a list containing as many strings as columns in the file. Regarding the output of the "reader" function, this is passed to the higher-level functions in the "ASCIIManager" library for further file handling operations, namely the splitting of the iterable output object into a header, a datasection and a tail list.

D.3. Third level (lower-intermediate) libraries

This lower-intermediate level the Coupling Interface features two libraries focused on either executing the tools or handling ASCII files in general

Trigger Manager

This library is characterized by two classes, one for the PowerFLOW® CLI and the other one for the Simpack® simulation. In both classes, the full commands defined in the "Model" library are loaded and passed to the python *subprocess.call* function to actually execute the CLI commands on the Linux terminal. In the case of the Simpack® solver, as the goal is to just execute commands without caring about the output of the same, the "subprocess.call" function is invoked with only the required command as input. On the other hand, for the PowerFLOW® CLI class, the "subprocess.call" function is executed within a "with ... as" python statement in order to write the output of the "exaritool forces.ri" function from the PowerFLOW® CLI to the "PFout.txt" file.

ASCIIManager

Within the framework of Controller operations for the reading of coupling files, the intermediate level is embodied by the "ASCIIManager" library. This library is primarily dedicated to the splitting of a coupling file into a header, a datasection and, where required, a tail list, starting from the iterable object returned by the "reader" function in the "FileHandler" library. This is carried out by employing a main function called "file_splitter" which performs the splitting by looping through the items of the iterable input. Next, based on the input number of lines in the header and tail, when the value of the line counter is below the number of header lines, then the "header_extractor" function is called to store such a line into the header list, in the other case, all the other lines are appended to the datasection list. If the number of tail lines is greater than zero, then the "tail_extractor" function is invoked to fill also the tail list. An important aspect to point our is that all the coupling files, except for "PFout.txt", "PFin.txt", "tot_iter_list.txt" and "cpl_setup.txt", are treated as either space or comma-separated values ASCII files already obtaining both the line-wise and column-wise splitting of their datasection at the level of "ASCIIManager" library. On the other hand, the previous four ".txt" files feature a more complex structure in their datasection not allowing columns-wise data extraction at the level of the "ASCIIManager" library. For this reason, such files are read by using a new-line delimiter to just split them line-wise. Next, each string in the datasection or header list is passed to the dedicated 2-nd level library depending on the coupling file at stake, where specific functions for the extraction of the individual column items per line are employed.

D.4. Second level (upper-intermediate) libraries

The main purpose of the libraries in the second upper-intermediate level of the Interface is to either take care of the generation of all the coupling files, to perform the PowerFLOW® initial transient check and to execute a Simpack® simulation for every coupling iteration.

D.4.1. Coupling files libraries

tot_iter_listManager

This library is essentially devoted to the handling of the "tot_iter_list.txt" file, which is the file storing the number of total iterations carried out by the Coupling Interface in both its initial transient and coupling mode. This library contains two main functions: one to update the file at the end of each iteration, and one to restart it only in the case of tot_it=1 by deleting its old datasection content. Furthermore, the "updater" function calls other secondary functions to carry out specific subtasks. The first one reads the "tot_iter_list.txt" file itself and splits it into a header and datasection by calling the "file_splitter" function in the ASCIIManager library. Then, the extracted datasection is provided to a second function which updates it by appending a new line, namely the subsequent "tot_it" value. Next, another function assembles the updated datasection and header to generate a full list storing the whole content of the file. This object is then supplied to the "writer" function in the "FileHandler" library to update the "tot_iter_list.txt" file. On the other hand, the "restart" function performs similar operations but the datasection is now represented by a single item - namely the integer 1 - as this is required to start the Coupling Interface for the very first total coupling iteration. The header of this file is instead obtained from the class storing the headers of the coupling files in the "Model" library.

cpl_setupManager

This library is dedicated only to the reading of the data in the "cpl_setup.txt" file, which have to be manually defined by the user before running the aeroelastic simulation. Therefore, the main function in this library is aimed at extracting the content of the "cpl_setup.txt" file and its first task is the splitting of the file into a header and a datasection list. Next, there are there minor functions specialized in the extraction of the names, values and units from the datasection and header lists depending on the version of the coupling. This library is called by the library responsible for the generation of the "Spck.subvar" file for the very first coupling iteration.

cpl_forceManager

This library is responsible for the update and reset of the "cpl_force.csv" file, which is the data-container for the time-history of the PowerFLOW® forces employed during the coupling phase of the global aeroelastic simulation in its 1 or 2 DoFs versions. The working of this library is similar to "tot_iter_listManager", as there is a main "generator" function dedicated to the update of the file, and one focused on the reset of the file. In addition, also a function for the extraction of the complete time-history of each force component in the datasection of the file is present. The latter function is directly invoked by the "AFSManager" library to generate the "Spck.afs" input file. Regarding the "cpl_force.csv" file, in addition the time-history of the PowerFLOW® forces after the initial transient, this file stores also the coupling timestep duration, the coupling iteration number and global time at which each PowerFLOW® force is computed. These quantities are read by the Coupling Interface and employed in the generation of the "Spck.afs" file.

kin_statesManager

This library takes care of the update, reset and data-extraction for the "kin_states.csv" file, which is the data-container of the time-history of the kinematic states returned by the aeroelastic simulation. These three main tasks are carried out by two specific main functions, namely the "generator" and the "restarter". The "generator" function follows the same steps carried out by the analogue routine in "cpl_setupManager" and "tot_iter_listManager", with the only difference that a new line with 3N·DoF columns is added. The "restarter" function is also very similar to the one used by both libraries for the same purpose, just a different header is loaded from the "Model" library depending on the number of DoF.

D.4.2. PowerFLOW® libraries

init_trans_dataManager

This library is invoked at the end of the CFD initial transient to generated the "init_trans_data.csv" file collecting important data related to the converged very first airload to employ in the Simpack® simulation for cpl_it=0. The main function in this library is the "generator" one, which calls the other three secondary functions in this library. The first is dedicated to the extraction of the header from the "Model" library, depending on the number of DoF. The second takes care of the datasection generation and the third assembles the list where the full content of the file is stored. In the library, there is also another primary function, which is called "data_reader" and it is called at the beginning of the "CouplingIterationManager" 1-st level library to derive the value of the current "cpl_it" from "tot_it" and the data related to the end of the initial transient.

PFinManager

This function generates the PowerFLOW® input table by reading the same "PFin_<tot_it>.txt" file related to the previous iteration to extract its header (the header of this file doesn't change during the coupling). On the other hand, the datasection generation depends on the phase of the aeroelastic simulation. If this is still in its initial transient phase, then the new datasection is a copy of the old one, as PowerFLOW® needs to compute a more accurate airload for the same conditions of the previous "tot_it" iteration. Instead, if the aeroelastic simulation is in its true coupling phase, then the "kin_states.csv" and "init_trans_data.csv" files are read to extract the global time at which the initial transient is over and those kinematic states required by each version of the coupling and pitch motion modelling method in PowerFLOW®. In addition, also a flag prescribing if the initial transient is over or not is included in the datasection of the new "PFin_<tot_it>.txt" file. The other secondary functions in the library are always those responsible for the file splitting and assembly, in addition to a function generating the name for the new "PFin_<tot_it>.txt", namely "PFin_<tot_it+1>.txt". Regarding the structure of the latter file, it consist in a single-line datasection where all quantities are positioned one next to the other after three numbers related to the coordinates (0,0,0) of a single point "fake" grid required by the table-layout type employed among those available in [52]. The header of this file contains information related to the 3D grid where all data are virtually assigned, the number of data and their units, as discussed in sec.2.2.2.

PFoutManager

The "PFoutManager" library contains a set of functions processing data from the "PFout.txt" file. Given the only 1 and 2 DoFs versions of the coupling implemented, only the lift force and the out-of-plain aerodynamic moment are considered and the last value of each signal is returned to the "CouplingIterationManager". Such quantities are extracted by employing the standard procedure already described. The first step is the splitting of the "PFout.txt" file into its header and its datasection lists (where, in addition to a time column, there are 3 columns for 2D simulations) by the "splitter" function. Then, the datasection list is processed to extract the two lists related to the aerodynamic forces time-histories considered by the function "datasection_handler". Then, the third and last step is the extraction of the last item (current force) from each list by means of the function "curforce_extractor".

InitTransCheck

This library contains the functions necessary to carry out the signal processing of an arbitrary force component from the PowerFLOW® simulation. This task is crucial to check weather the initial CFD transient is over or not at the beginning of the full aeroelastic simulation, so that the proper coupling with Simpack® can start. The main function is called "transient_checker" and it calls the other sub-functions responsible for all the required sub-tasks. The first sub-function extracts the lift (for the 1 and 2 DoFs coupling) and moment (only for 2 DoFs) force-histories up to the current total iteration by calling a dedicated function from the "PFoutManager" library. Next, each force-component time-history is passed to the function performing the calculation of the running-average and root-mean-square of the signal based on the input size of the sliding window and by employing the equations and variables outlined in sec.5.4.1. Eventually, the last sub-function, evaluates weather the initial transient is over or not by checking if there is at least one signal with a value for its scaled root-mean-square smaller than the input tolerance. In this function, the user can chose to adopt a more stringent criterion by considering the convergence of all the force signals processed. In any case, the convergence check always returns a non-convergence result if the current "tot_it" value is less then or equal to the window size, and such an outcome is passed to the "TransientIterationManager" library which then reacts as discussed in sec.5.4.4.

D.4.3. Simpack® libraries

AFSManager

This library is responsible for all the operations related to data-extraction and generation of the Simpack® input file "Spck.afs" employed in the coupling to supply the aerodynamic forces from PowerFLOW® to the MBD simulation. For both tasks, there are two main functions: the "extract_datasection" and the "generator" one. The first extracts the quantities of interest from the datasection of the file, namely the coupling timestep used as duration for the Simpack® simulation and the two input force-values for each force-component. The two force-values are the same according to what explained in sec.5.2.2. The second main function generates the new "Spck.afs" file in the same exact way as the other coupling files. The "old Spck.afs" file from the previous iteration is loaded and split into a header and a datasection. The header is kept to generate the header of the new file, while the datasection is updated with the last force value in the "PFout.txt" file for each component. Then the updated datasection and the old header are assembled to get a new list which is then provided to the "writer" function of the "FileHandler" library to generate the new file. To mention a few properties of the "Spck.afs" file, in its header the *format* of the datasection, the name and units of each column as well as the type of interpolation (*Akima spline*) to be applied by the solver are specified. The datasection format employed is *array of univariate functions*, which means that the datasection features one time column and as many other columns as *input-functions* considered. Due to the way the coupling methodology is developed (see sec.5.1), there are always two lines in the datasection, the first is for the airload components to apply for $t_n = 0$ and the second contains the same for $t_{n+1} = \Delta_T$, as every coupling iteration is characterized by a single dedicated Simpack® simulation.

SubvarManager

The "SubvarManager" library is the one taking care of the generation of a new "Spck.subvar" input file for the MBD simulation. As the data to employ in the setup of the MBD simulation are different if "cpl_it" is either zero or not and depending on the DoF value, a specific function - called "datasection_generator" is implemented to handle all these different situations. After splitting the "Spck.subvar" file, if cpl_it=0 and DoF=1, then this function extracts the initial conditions for the plunge DoF, as well as the amplitude and frequency of the prescribed pitch motion from the "cpl_setup.txt" file by invoking the functions from the "cpl_setupManager" library. If cpl_it=0 and DoF=2, then the initial conditions for the plunge and pitch DoF are all extracted from the "cpl_setup.txt" file. In any case, only when cpl_it=0, the initial conditions of the Simpack® simulation coincide with the initial conditions of the global aeroelastic simulation. On the other hand, when cpl_it≠0, there are always two further conditions related to the number of DoF. If DoF=1, then the amplitude and frequency of the prescribed pitch motion are always extracted from the "cpl_setup.txt" file, while the values for the plunge motion initial conditions of the Simpack® simulation are obtained from the "Spck.mat" file by means of the functions in the "SpckOManager" library. When DoF=2, then both the position and velocity states for the pitch and plunge motions are obtained from the "Spck.mat" file. In both cases, the *subvars* related to the global simulation time and the Simpack® simulation duration (coupling timestep) are obtained from the "PFout.txt" file via the functions in the "PFoutManager" library. More in detail, the actual extraction is carried out by the functions in the class "DatasectionGenerator" and once updated datasection is generated, then the other tasks are the same as for every other coupling file (assemble the header and the updated datasection lists, and write to file the merged list). Finally, the datasection of the "Spck.subvar" file contains for every *subvar* name, value and unit, while the header of this section has to respect the rules outlined in [65] for its correct reading by the Simpack® solver.

SpckOManager

This library is only dedicated to the reading of the data in the binary "Spck.mat" output file generated by Time Integration Solver of the MBD simulation tool. To access the data in this file, the h5py python package [90] is exploited in a dedicated "Extractor" class which extracts all the required kinematic states in the file. The lolgical order followed when this library is employed is the following. The "Extractor" class is first called and its outputs are the two lists emulating the header and datasection of the old "SpckOUT.csv" file employed in the coupling demonstrator to keep its framework. Then, the initial and final values for the position, velocity and acceleration states of each DoF are extracted and stored in three corresponding lists. Each one of these lists contains as many ordered dictionaries as input DoF considered.

SpckSimManager.sjs

As already discussed in sec.5.2.5, most of the Simpack® MBD simulation needs to be manually set-up depending on the problem (1 or 2 DoFs) considered, while the "SpckSimManager.sjs" script just carries-out those operations necessary to fill-in the new data required by each iteration. This allows to significantly reduce the time required by each coupling iteration, leading to a significant reduction in the total time for the coupled aeroelastic simulation. Therefore, such script essentially takes care of three tasks. Firstly, the update of the initial conditions with the position and velocity states for each DoF coming from the previous iteration is carried-out. This is possible because the "Spck.subvar" file containing the *subvars* for the initial conditions is automatically loaded when the script opens the "Spck.spck" simulation file, as well as the updated "Spck.afs" file corresponding to the current iteration. Secondly, the reset to zero of the Simpack® simulation time and thirdly the execution of the *Time Integration Solver* from $t = 0$ up to $t = \Delta_T$ in order to obtain a new binary ".mat" result file. The execution of the "SpckSimManager.sjs" script is triggered by the Coupling Interface by calling the Simpack® solver with the full command defined in the "Model" library.

D.5. Folders structure

The coupled aeroelastic simulation between PowerFLOW® and Simpack® is entirely contained in a single folder, which is labelled as "<>" in tab.D.2 where both the PowerFLOW® and the Simpack® simulation files, as well as all the libraries related to the Coupling Interface are stored in separate subfolders. First of all, the coupling simulation folder "<>" is the location where the PowerFLOW® "<>.cdi"[2] file is generated. Based on this file, the "exaqsub" command is launched by the bash "start_sim.sh" script which executes then the PowerFLOW® simulation. In "<>", also other files are located, namely the "tot_iter_list.txt" coupling file and the bash script "trigger_interface.sh", which is called by the command executed before each "PFin_<tot_it>.txt" input table reading by the PowerFLOW® simulator to trigger the Coupling Interface libraries. The latter are all stored in the "<>/Interface" subfolder, in addition to the "FileCleaner.py" python script. The "FileCleaner.py" is invoked by the "start_sim.sh" bash script in order to clean or reset the coupling files generated by a previous run of the coupling simulation before the "exaqsub" command.

<>	<>/Interface
PF_sim.cdi	[all libraries
tot_iter_list.txt	shown in
trigger_interface.sh	tab.D.1]
start_sim.sh	FileCleaner.py

<>/PFin	<>/PFout	<>/SpckIN	<>/SpckOUT	<>/SpckSIM
coupling_setup.txt	PFout.txt	Spck.afs	Spck.mat	Spck.spck
PFin_*.txt	cpl_forces.txt	Spck.subvar	kin_states.csv	
	init_trans_data.csv			

Table D.2: Table showing the folders structure employed for the PowerFLOW®-Simpack® coupling.

In the lower part of tab.D.2, it is possible to visualize the two subfolders related to PowerFLOW® ("<>/PFin" and "<>/PFout") and the three ones related to Simpack® ("<>/SpckIN", "<>/SpckOUT" and "<>/SpckSIM"). The PowerFLOW®-related subfolders contain the aerodynamic coupling files ("coupling_setup.txt", "init_trans_data.csv" and "cpl_forces.csv") as well as the input and output files required by the CFD simulation ("PFin_<tot_it>.txt" and "PFout.txt" respectively). The three subfolders required for the Simpack® simulation store the dedicated input files ("Spck.afs" and "Spck.subvar"), output ("Spck.mat"), coupling-related files ("kin_states.csv") and also the Simpack® simulation "Spck.spck" file, which is located in the "<>/SpckSIM" subfolder.

[2] The PowerFLOW® ".cdi" file is a mixed binary and ASCII file and it is essential to start the CFD simulation. It is generated by PowerCASE based on the simulation setup and is read by both the PowerFLOW® discretizer, decomposer and simulator to compute the CFD solution.

E

The V-g method

The V-g method is a technique employed to determine the flutter velocity and reduced frequency of an aeroelastic system regardless of its dynamic response. This is done by considering the system in the domain of the Fourier reduced-frequency variable, determining its properties in the neighbourhood of the flutter boundary. In this chapter of the appendix, the formulation of this method is derived for the 2 DoFs semi-rigid airfoil problem introduced in sec.3.1.1 and employed in sec.6.3 for the 2 DoFs PowerFLOW®-Simpack® coupling.

The derivation for the problem considered is inspired to the work reported in [84]. It starts by considering the dimensionless EoMs for the structural part of the pitching and plunging airfoil obtained in sec.3.1.2 coupled with the dimensionless aerodynamic lift L^* and moment M_A^* terms provided by the original Theodorsen model for a pitching axis location at the quarter-chord:

$$\begin{bmatrix} 1 & \xi_G \\ \xi_G & r_\alpha^2 - \xi_E^2 + 2\xi_E\xi_G \end{bmatrix} \begin{bmatrix} \ddot{h} \\ \ddot{\alpha} \end{bmatrix} + \Omega^2 \begin{bmatrix} 1 & \xi_E \\ \xi_E & \frac{r_\alpha^2}{\Omega^2} + \xi_E^2 \end{bmatrix} \begin{bmatrix} h \\ \alpha \end{bmatrix} = \begin{bmatrix} -L^* \\ M_A^* - \xi_A L^* \end{bmatrix}$$

$$L^* = \frac{1}{2a}\left(\ddot{h} + \frac{1}{2}\ddot{\alpha} + \hat{U}\dot{\alpha}\right) + \frac{\hat{U}}{a}C(k)\left(\dot{h} + \dot{\alpha} + \hat{U}\alpha\right) \quad \text{and} \quad M_A^* = -\frac{1}{2a}\left(\frac{\ddot{h}}{2} + \frac{3}{8}\ddot{\alpha} + \hat{U}\dot{\alpha}\right) \tag{E.1}$$

For an explanation of the quantities employed in the above equations and the dimensionalization procedure, the reader is reference to sec.3.1.3. If the expressions for the dimensionless lift and aerodynamic moment are plugged into the structural system, the following hybrid time-reduced frequency aeroelastic system of equations is obtained:

$$\begin{bmatrix} 1 & \xi_G \\ \xi_G & r_\alpha^2 - \xi_E^2 + 2\xi_E\xi_G \end{bmatrix} \begin{bmatrix} \ddot{h} \\ \ddot{\alpha} \end{bmatrix} + \Omega^2 \begin{bmatrix} 1 & \xi_E \\ \xi_E & \frac{r_\alpha^2}{\Omega^2} + \xi_E^2 \end{bmatrix} \begin{bmatrix} h \\ \alpha \end{bmatrix} =$$

$$- \begin{bmatrix} \frac{1}{2a} & \frac{1}{4a} \\ \frac{1}{4a} + \frac{\xi_A}{2a} & \frac{3}{16a} + \frac{\xi_A}{4a} \end{bmatrix} \begin{bmatrix} \ddot{h} \\ \ddot{\alpha} \end{bmatrix} - \begin{bmatrix} \frac{\hat{U}C(k)}{a} & \frac{\hat{U}}{2a} + \frac{\hat{U}C(k)}{a} \\ \frac{\hat{U}\xi_A C(k)}{a} & \frac{\hat{U}}{2a} + \frac{\xi_A\hat{U}}{2a} + \frac{\xi_A\hat{U}C(k)}{a} \end{bmatrix} \begin{bmatrix} \dot{h} \\ \dot{\alpha} \end{bmatrix} - \begin{bmatrix} 0 & \frac{\hat{U}^2C(k)}{a} \\ 0 & \frac{\xi_A\hat{U}^2C(k)}{a} \end{bmatrix} \begin{bmatrix} h \\ \alpha \end{bmatrix} \tag{E.2}$$

The first step of the derivation consists in rewriting the complete aeroelastic system above in the domain of the Fourier reduced frequency variable by applying the following substitutions $h = \tilde{h}$, $\dot{h} = j\hat{\omega}\tilde{h}$ and $\ddot{h} = -\hat{\omega}^2\tilde{h}$ for the plunge DoF, as well as $\alpha = \tilde{\alpha}$, $\dot{\alpha} = j\hat{\omega}\tilde{\alpha}$ and $\ddot{\alpha} = -\hat{\omega}^2\tilde{\alpha}$ for the pitch DoF. The variable $\hat{\omega}$ is the dimensionless reduced frequency, defined as $\hat{\omega} = \omega/\omega_\alpha$ and related to the reduced frequency k by means of the relation $\hat{\omega} = k\hat{U}$. This leads to the following system, completely defined in the complex domain:

$$-\hat{\omega}^2 \begin{bmatrix} 1 & \xi_G \\ \xi_G & r_\alpha^2 - \xi_E^2 + 2\xi_E\xi_G \end{bmatrix} \begin{bmatrix} \tilde{h} \\ \tilde{\alpha} \end{bmatrix} + \Omega^2 \begin{bmatrix} 1 & \xi_E \\ \xi_E & \frac{r_\alpha^2}{\Omega^2} + \xi_E^2 \end{bmatrix} \begin{bmatrix} \tilde{h} \\ \tilde{\alpha} \end{bmatrix} =$$

$$\hat{\omega}^2 \begin{bmatrix} \frac{1}{2a} & \frac{1}{4a} \\ \frac{1}{4a} + \frac{\xi_A}{2a} & \frac{3}{16a} + \frac{\xi_A}{4a} \end{bmatrix} \begin{bmatrix} \tilde{h} \\ \tilde{\alpha} \end{bmatrix} - j\hat{\omega} \begin{bmatrix} \frac{\hat{U}C(k)}{a} & \frac{\hat{U}}{2a} + \frac{\hat{U}C(k)}{a} \\ \frac{\hat{U}\xi_A C(k)}{a} & \frac{\hat{U}}{2a} + \frac{\xi_A\hat{U}}{2a} + \frac{\xi_A\hat{U}C(k)}{a} \end{bmatrix} \begin{bmatrix} \tilde{h} \\ \tilde{\alpha} \end{bmatrix} - \begin{bmatrix} 0 & \frac{\hat{U}^2C(k)}{a} \\ 0 & \frac{\xi_A\hat{U}^2C(k)}{a} \end{bmatrix} \begin{bmatrix} \tilde{h} \\ \tilde{\alpha} \end{bmatrix} \tag{E.3}$$

At this point, the above system is manipulated by keeping all the structural $\hat{\omega}$-dependent terms on the lhs and by collecting all the aerodynamic k-dependent terms on the rhs into a unique $\boldsymbol{E^*}(k)$ dimensionless aerodynamic-matrix. The matrix $\boldsymbol{E^*}(k)$ features both terms resulting from combinations of polynomials in the k variable and complex elements due to the $C(k)$ Theodorsen function, which introduces an infinite number of aerodynamic lag-states related to the wake memory-effect. The later aspect is not an issue for the V-g method, as the goal is to investigate the flutter boundary of the system, and not its dynamic response or root-locus.

As carried out in sec.3.1.3, the two structural matrices on the lhs are labelled as $\boldsymbol{M^*}$ (dimensionless mass-matrix) and $\boldsymbol{K^*}$ (dimensionless stiffness-matrix), in addition to a dimensionless vector of the pitch and plunge DoFs $\boldsymbol{\tilde{x}}$:

$$\Big(- \hat{\omega}^2 \boldsymbol{M^*} + \boldsymbol{K^*} \Big) \boldsymbol{\tilde{x}} = \frac{\hat{U}^2}{2a} \boldsymbol{E^*}(k) \boldsymbol{\tilde{x}} \quad \text{with} \quad \boldsymbol{K^*} = \Omega^2 \begin{bmatrix} 1 & \xi_E \\ \xi_E & \frac{r_\alpha^2}{\Omega^2} + \xi_E^2 \end{bmatrix} \quad \boldsymbol{M^*} = \begin{bmatrix} 1 & \xi_G \\ \xi_G & r_\alpha^2 - \xi_E^2 + 2\xi_E \xi_G \end{bmatrix}$$

$$\text{and} \quad \boldsymbol{E^*}(k) = \begin{bmatrix} k^2 - 2jkC(k) & \frac{k^2}{2} - jk - 2jkC(k) - 2C(k) \\ \frac{k^2}{2} - \xi_A k^2 - 2jkC(k)\xi_A & \frac{3}{8}k^2 + \frac{k^2}{2}\xi_A - jk - jk\xi_A - 2jkC(k)\xi_A - 2C(k)\xi_A \end{bmatrix} \tag{E.4}$$

Next, several further manipulations steps are required to convert the above compact formulation of the system into the final formulation of the V-g method. The first step consists in multiplying the above matrix equation by the factor[1] $1/\hat{\omega}^2 \boldsymbol{M^*}^{-1}$ and in bringing all the resulting terms to the left. This leads to the equation below:

$$\Big(- \boldsymbol{I} + \frac{1}{\hat{\omega}^2} \boldsymbol{M^*}^{-1} \boldsymbol{K^*} - \frac{\hat{U}^2}{2a\hat{\omega}^2} \boldsymbol{M^*}^{-1} \boldsymbol{E^*}(k) \Big) \boldsymbol{\tilde{x}} = \boldsymbol{0} \tag{E.5}$$

In the second step, the above expression is multiplied by the inverse of the $\boldsymbol{\Omega}^2$ matrix[2], defined as $\boldsymbol{\Omega}^2 = \boldsymbol{M^*}^{-1} \boldsymbol{K^*}$, obtaining the below result:

$$\Big(- \boldsymbol{\Omega}^{2^{-1}} + \frac{1}{\hat{\omega}^2} \boldsymbol{I} - \frac{\hat{U}^2}{2a\hat{\omega}^2} \boldsymbol{\Omega}^{2^{-1}} \boldsymbol{M^*}^{-1} \boldsymbol{E^*}(k) \Big) \boldsymbol{\tilde{x}} = \boldsymbol{0} \tag{E.6}$$

By rearranging the terms in the above equation, the final expression of the V-g method is obtained:

$$\Bigg[\frac{1}{\hat{\omega}^2} \boldsymbol{I} - \boldsymbol{\Omega}^{2^{-1}} \Big(\boldsymbol{I} + \frac{\hat{U}^2}{2a\hat{\omega}^2} \boldsymbol{M^*}^{-1} \boldsymbol{E^*}(k) \Big) \Bigg] \boldsymbol{\tilde{x}} = \boldsymbol{0} \qquad \rightarrow \qquad \Bigg[\frac{1}{\hat{\omega}^2} \boldsymbol{I} - \boldsymbol{A^*}(k) \Bigg] \boldsymbol{\tilde{x}} = \boldsymbol{0} \tag{E.7}$$

The final system is a typical eigenvalue problem for the $\boldsymbol{A}^*(k)$ matrix. However, the elements in the matrix are not constant and they need to be real positive numbers. Therefore, finding the solution to the problem $det[\boldsymbol{I}/\hat{\omega}^2 - \boldsymbol{A^*}(k)] = 0$ is equivalent to find the couple $(k_{flt}, \hat{\omega}_{flt})$ which leads to eigenvalues for the matrix $\boldsymbol{A}^*(k)$ equal to $1/\hat{\omega}^2$, for a non-zero eigenvector $\boldsymbol{\hat{z}}$.

In order to find the couple $(k_{flt}, \hat{\omega}_{flt})$, an iterative process is required where the eigenvalues of $\boldsymbol{A}^*(k)$ are computed for a decreasing value of k. This means that the first value of k should be sufficiently large to make sure that the search for a flutter velocity starts from very low inflow velocity values. The flutter condition is met when there is at least one real positive eigenvalue among those returned by $\boldsymbol{A}^*(k)$ for the given k. Once the $k = k_{flr}$ reduced frequency is found, then the flutter velocity is found as well, namely $\hat{U} = \hat{U}_f = \hat{\omega}_{flt}/k_{flt}$ with $\hat{\omega}_{flt} = \sqrt{1/\lambda}$ and λ the real-positive eigenvalue found.

The search algorithm implemented consists in a "for-loop" where the value of k is changed from the initial guess $k_0 = 5.0$ (which corresponds to a dimensionless velocity $\hat{U} = 0.1308$) by a step factor 0.001 for 4900 iterations. This leads to a maximum detectable value for the dimensionless velocity $\hat{U} = 7.9582$. The flutter velocity has been found within this range[3] at the 4838-th iteration corresponding to a value for the reduced frequency $k_{flt} = 0.162$ which eventually leads to $\hat{\omega}_{flt} = 0.8374$ and $\hat{U}_{flt} = 5.1695$.

[1] The matrix $\boldsymbol{M^*}$ can be inverted due to its symmetric and positive-defined character. This is related to the fact that the mass matrix is linked to the kinetic energy.

[2] The matrix $\boldsymbol{\Omega}^2$ is a matrix of squared frequencies. In fact, if the elastic and the mass center coincide, then the diagonal terms of $\boldsymbol{\Omega}^2$ get precisely $\omega_\alpha^2 = k_\alpha/J$ and $\omega_h^2 = k_h/m$. Besides, this matrix can be inverted, since also $\boldsymbol{K^*}$ is symmetric and positive defined due to the close relation to the elastic energy.

[3] This range is employed as it works for the given values of the dimensionless problem parameters. For different conditions, the range will most likely need to be modified.

F

Remaining results

F.1. Simpack® MBD setup verification

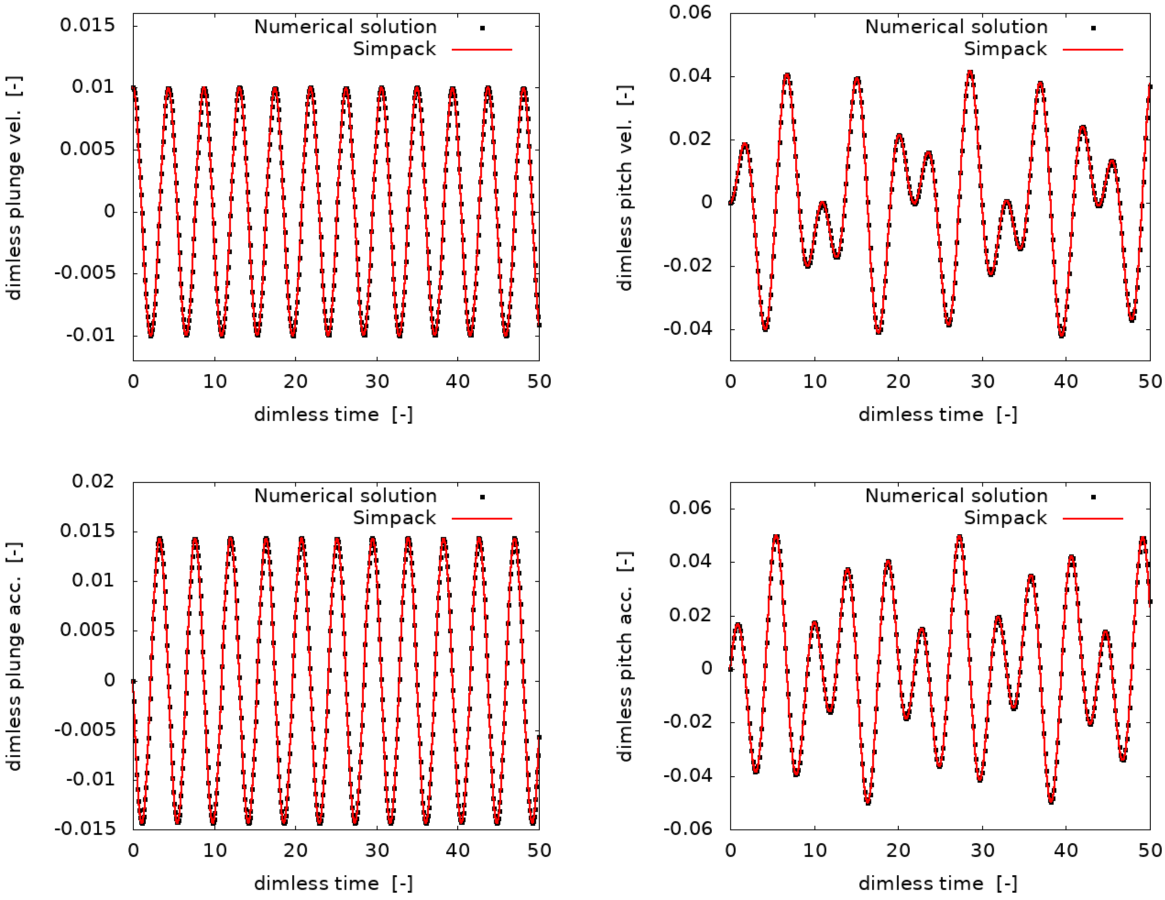


Figure F.1: Comparison between the time-histories of the dimensionless plunge velocity (top left), pitch velocity (top right), plunge acceleration (bottom left) and pitch acceleration (bottom right) returned by the numerical reference solution and the Simpack® MBD simulation for the steady aeroelastic problem. Both solutions are computed by employing a timestep $d\hat{t} = 0.0001s$ under a dimensionless freestream velocity $\hat{U} = 0.6849$.

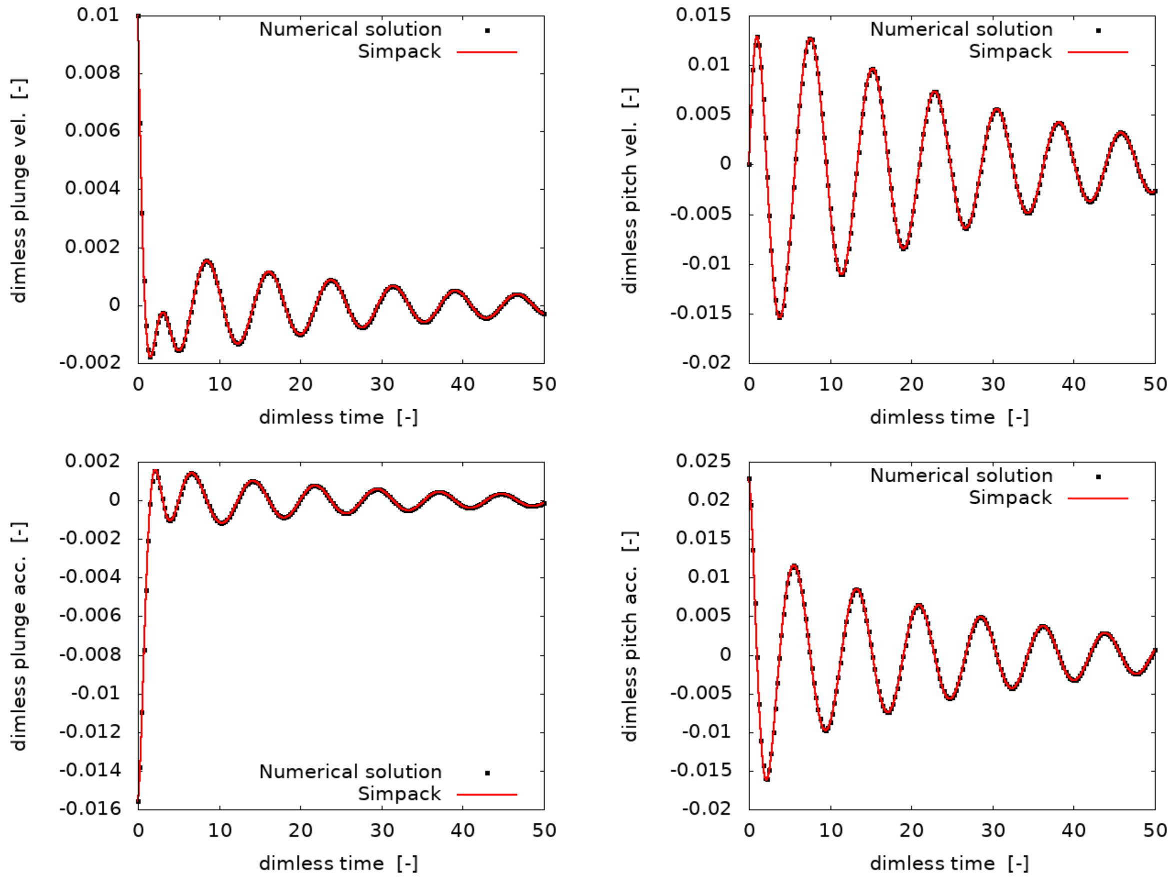


Figure F.2: Numerical and Simpack® kinematic states for the quasi-steady aeroelastic problem under a dimensionless freestream velocity $\hat{U}$=0.6174.

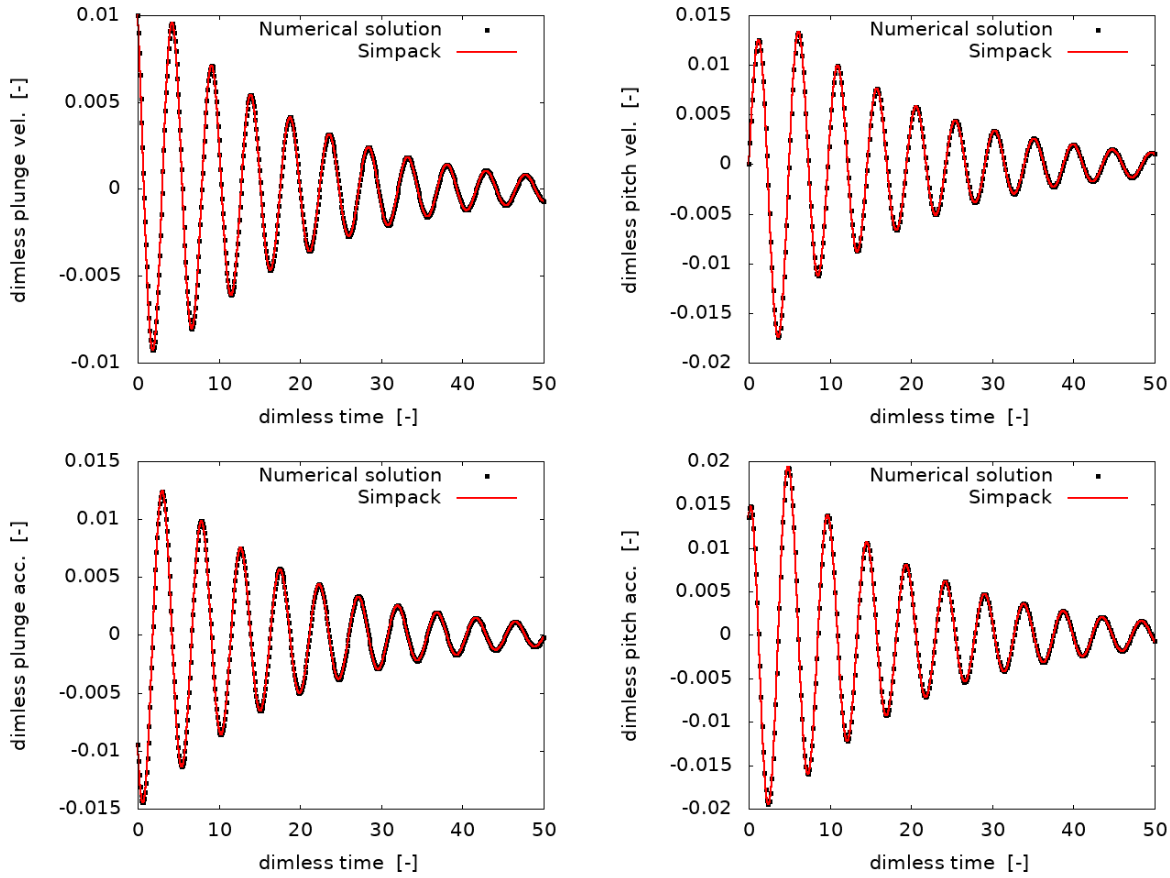


Figure F.3: Numerical and Simpack® kinematic states for the Theodorsen model with lag-states method under a dimensionless freestream velocity $\hat{U}$=1.0746.

F.2. Coupling demonstrator, 1 DoF problem

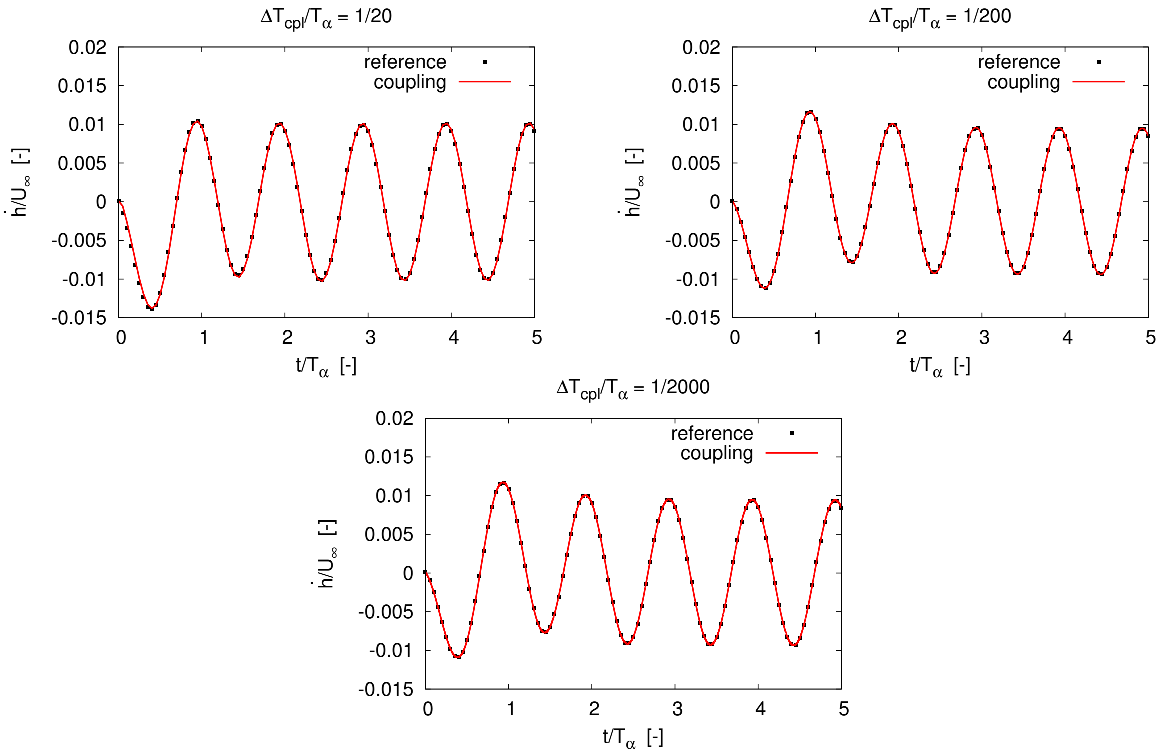


Figure F.4: Plunge velocity $\dot{h}$ for the demonstrator coupling (red lines) and reference numerical solution (black squares) for three values of the coupling time-step. Freestream flow velocity $U_\infty = 100$m/s and $T_\alpha = 0.2$s.

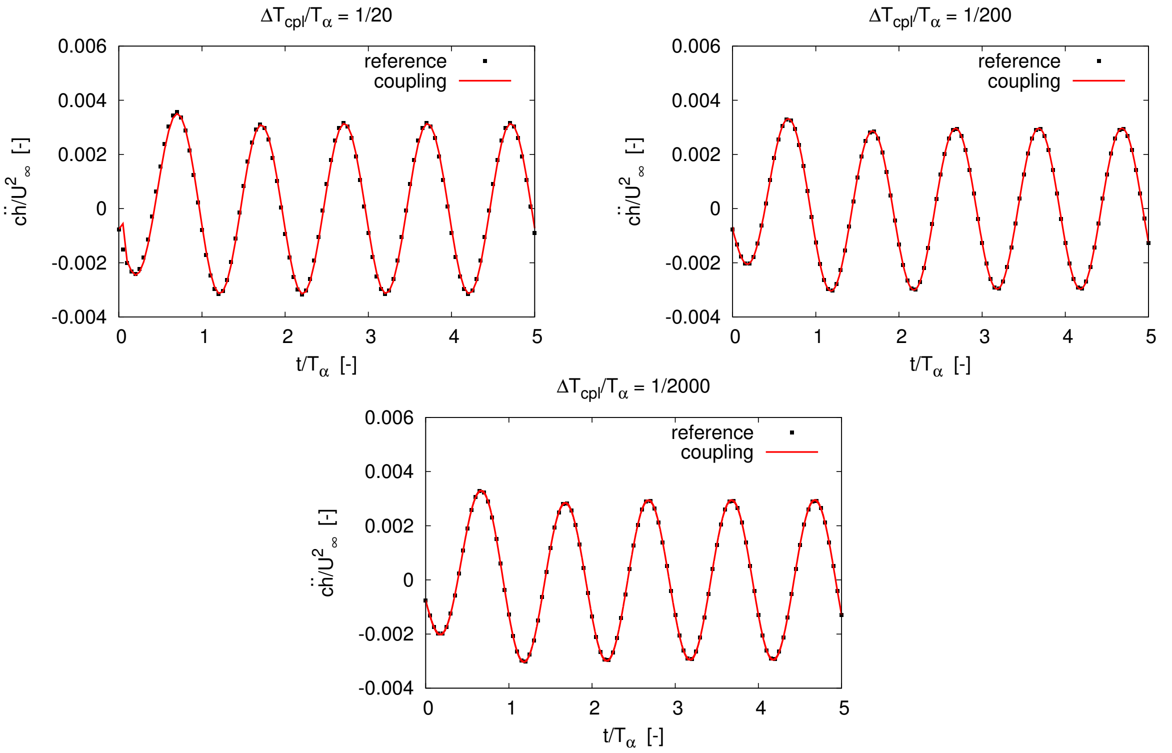


Figure F.5: Plunge acceleration $\ddot{h}$ for the coupling demonstrator (red lines) and reference numerical solution (black squares) for three values of the coupling time-step Chord $c = 1$m, velocity $U_\infty = 100$m/s and $T_\alpha = 0.2$s.

F.3. Coupling demonstrator, 2 DoFs problem

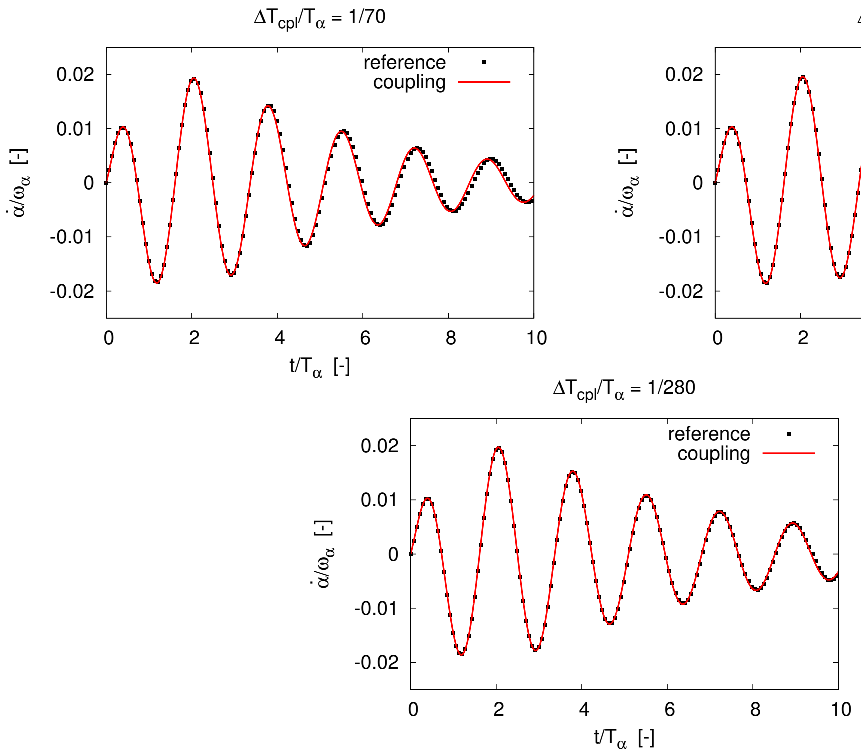


Figure F.6: Pitch velocity $\dot{\alpha}$ for the coupling demonstrator (red line) against the reference numerical solution (black squares) for three values of the coupling time-step, $T_\alpha = 0.1405$s and pitch angular frequency $\omega_\alpha = 44.72$rad/s.

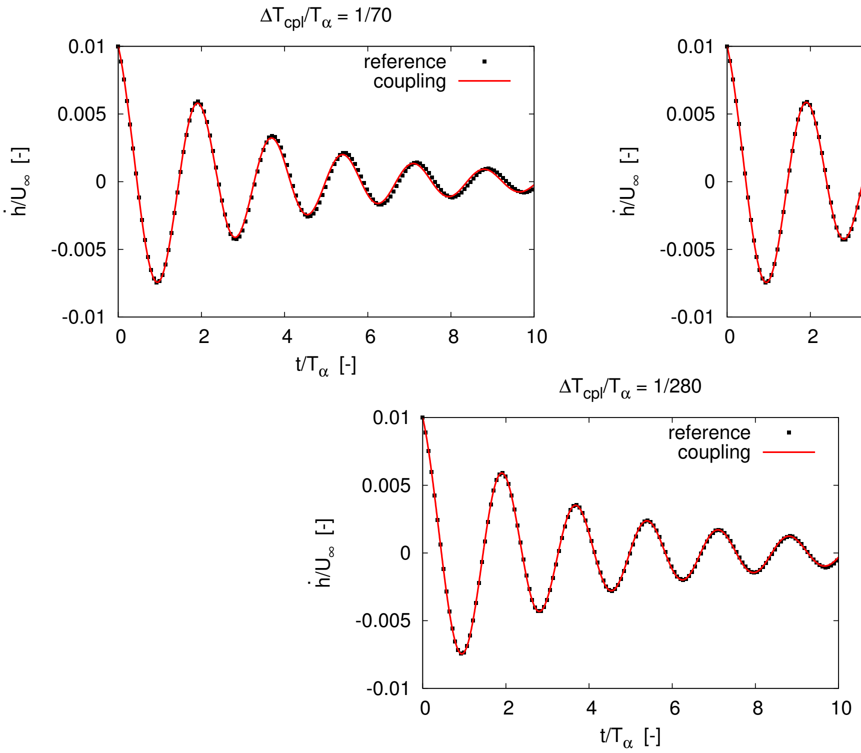


Figure F.7: Plunge velocity $\dot{h}$ for the coupling demonstrator (red line) against the reference numerical solution (black squares) for three values of the coupling time-step with $T_\alpha = 0.1405$s and free-stream velocity $U_\infty = 52$m/s.

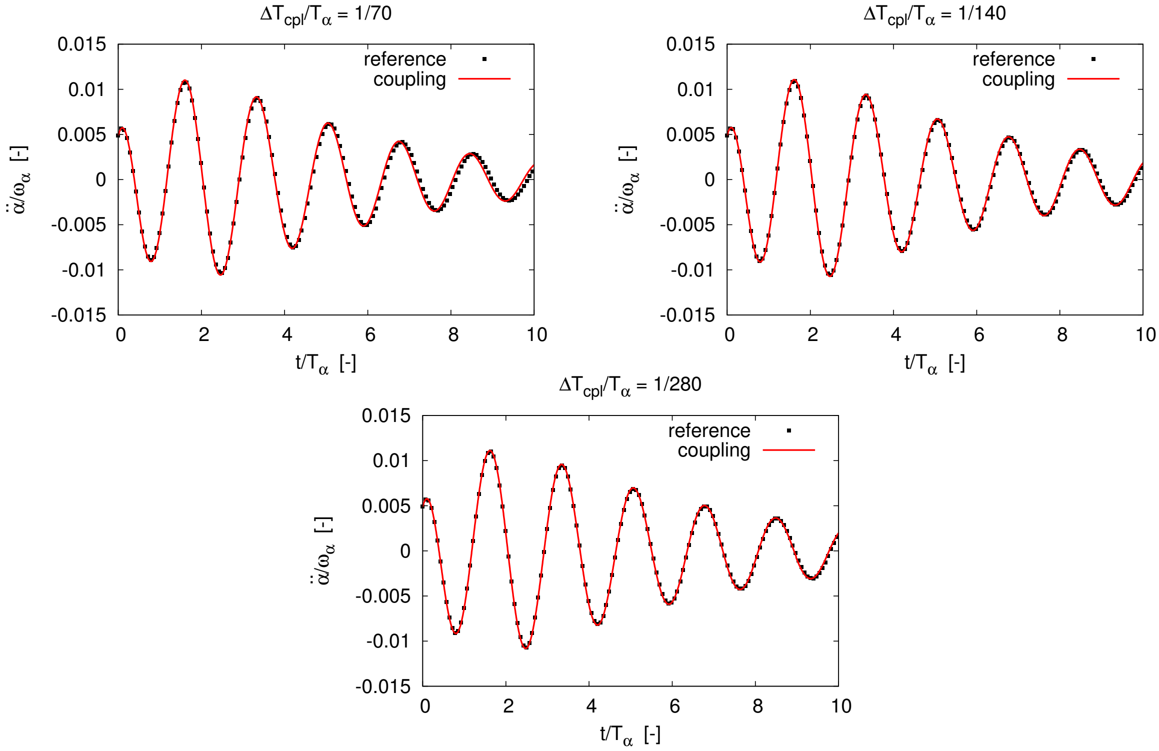


Figure F.8: Pitch acceleration $\ddot{\alpha}$ for the coupling demonstrator (red line) against the reference numerical solution (black squares) for three values of the coupling time-step with $T_\alpha = 0.1405$s and pitch angular frequency $\omega_\alpha = 44.72$rad/s.

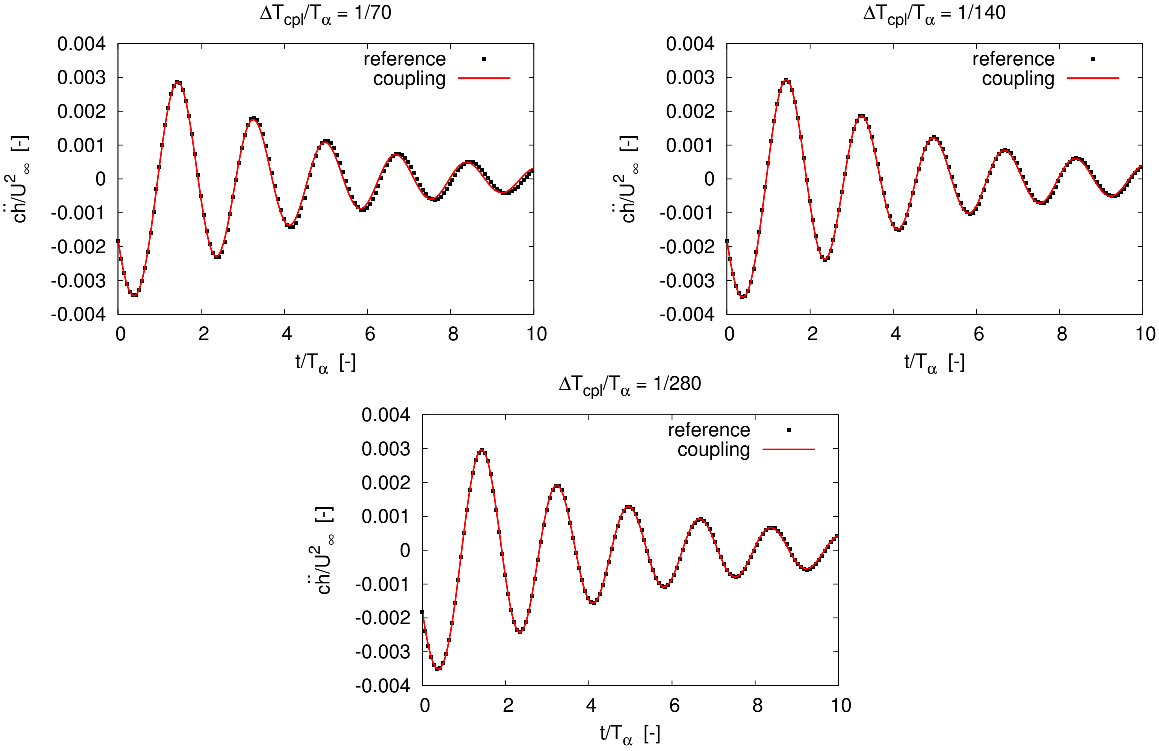


Figure F.9: Plunge acceleration $\ddot{h}_O$ for the coupling demonstrator (red line) against the reference numerical solution (black squares) for three values of the coupling time-step with $T_\alpha = 0.1405$s, $c = 1$m and $U_\infty = 52$m/s.

F.4. PowerFLOW®-Simpack® coupling, 1 DoF pitch-LRF

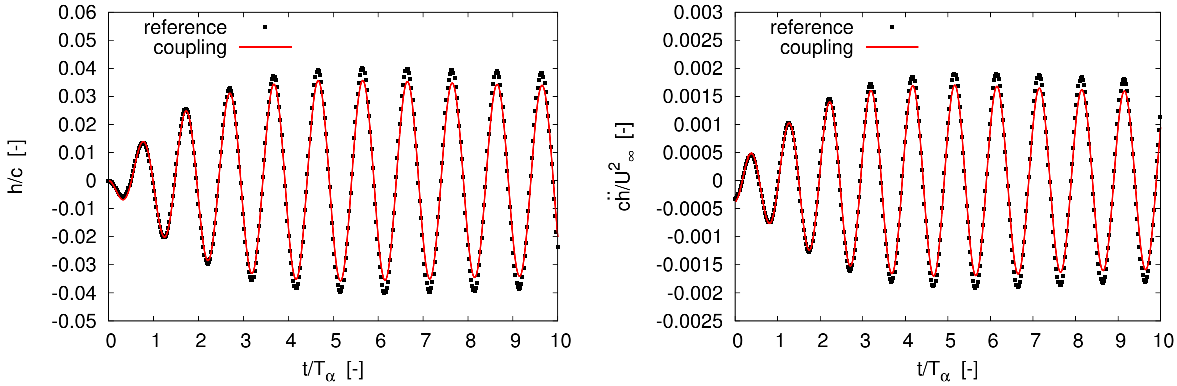


Figure F.10: Time-histories of the plunge position h ($c = 0.46$), and acceleration $\ddot{h}$ ($U_\infty = 34.7223$m/s) for test case #1: amplitude of pitch motion $A = 1°$, reduced frequency $k = 0.11$ and $T_\alpha = 0.385$s.

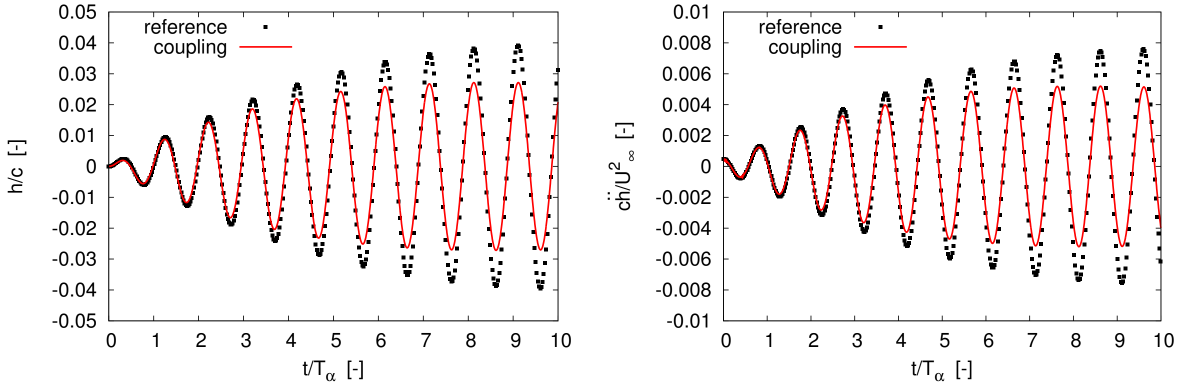


Figure F.11: Test case #2: $A = 1°$, $k = 0.22$ and $T_\alpha = 0.192$s.

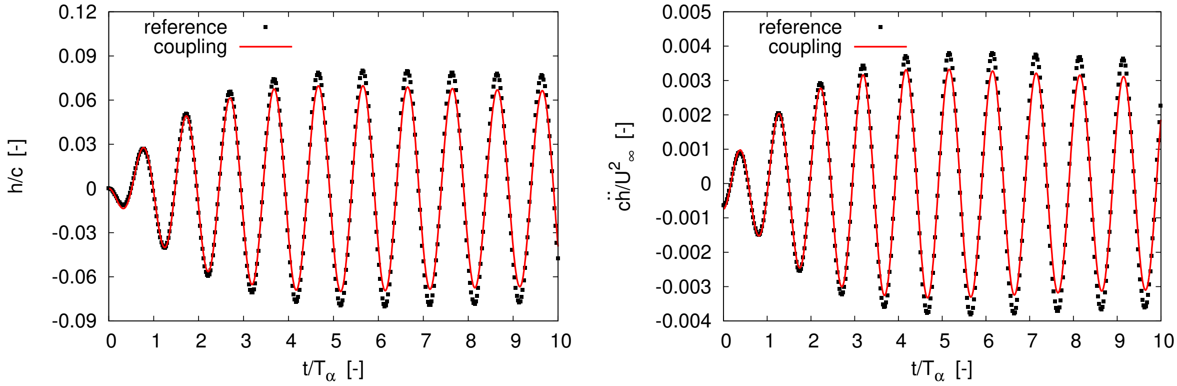


Figure F.12: Test case #3: $A = 2°$, $k = 0.11$ and $T_\alpha = 0.385$s.

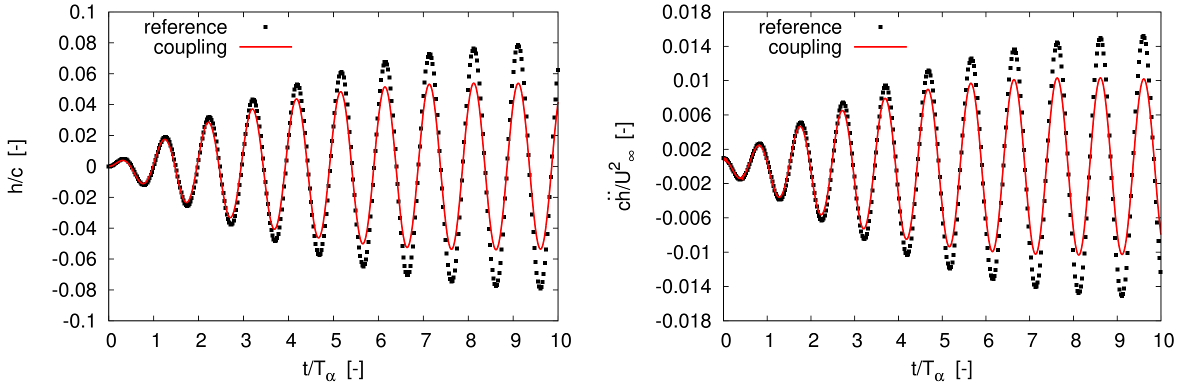


Figure F.13: Test case #4: $A = 2°$, $k = 0.22$ and $T_\alpha = 0.192$s.

F.5. PowerFLOW®-Simpack® coupling, 1 DoF pitch body-forces

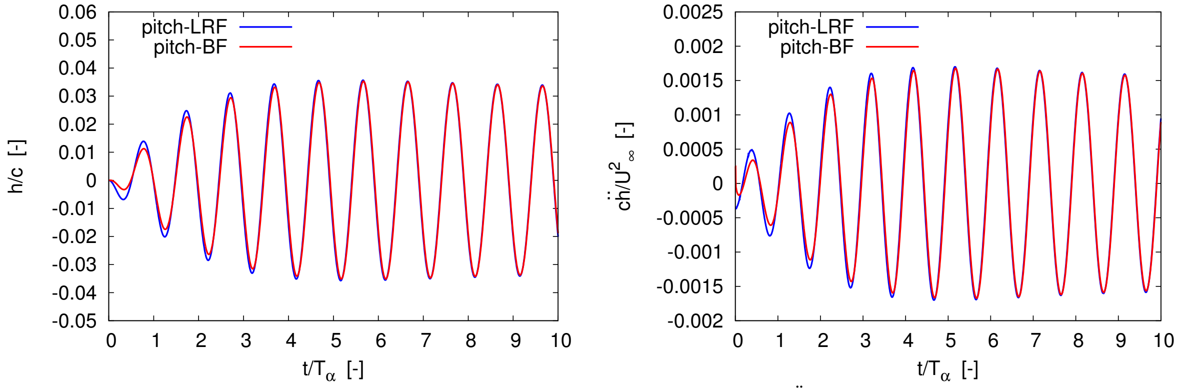


Figure F.14: Time-histories of plunge position h ($c = 0.46$), and acceleration $\ddot{h}$ ($U_\infty = 34.7223$m/s) for test case #1: amplitude of pitch motion $A = 1°$, reduced frequency $k = 0.11$ and $T_\alpha = 0.385$s.

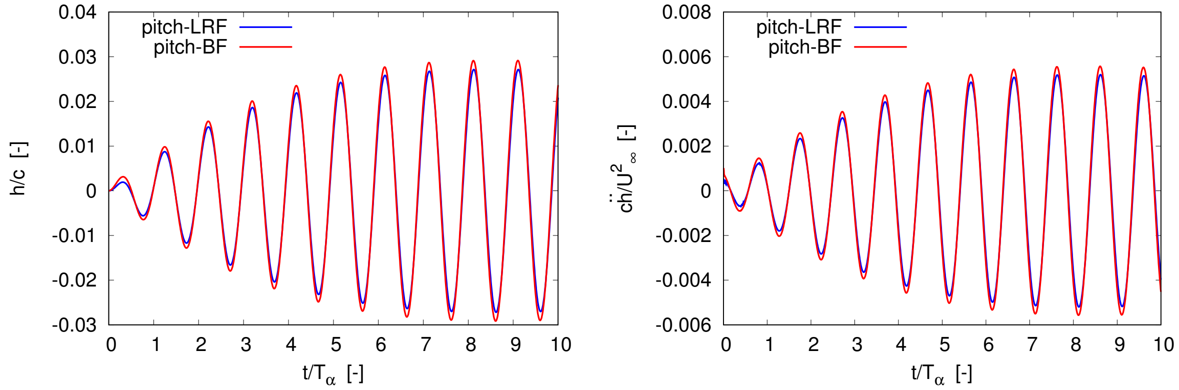


Figure F.15: Test case #2: $A = 1°$, $k = 0.22$ and $T_\alpha = 0.192$s.

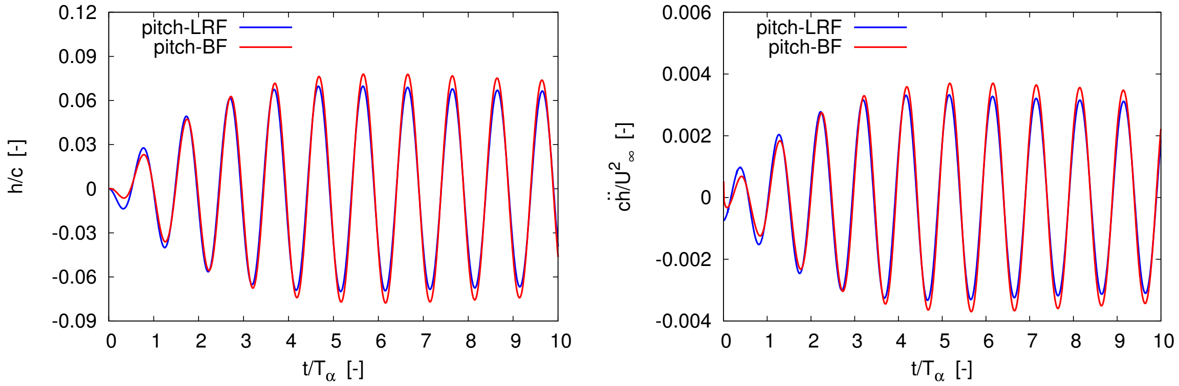


Figure F.16: Test case #3: $A = 2°$, $k = 0.11$ and $T_\alpha = 0.385$s.

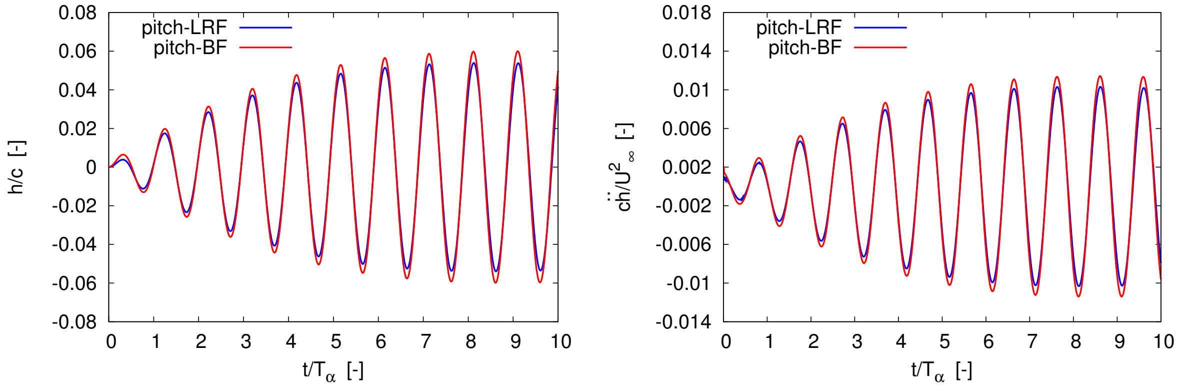


Figure F.17: Test case #4: $A = 2°$, $k = 0.22$ and $T_\alpha = 0.192$s.

F.6. PowerFLOW®-Simpack® coupling, 2 DoFs, #1 and #2

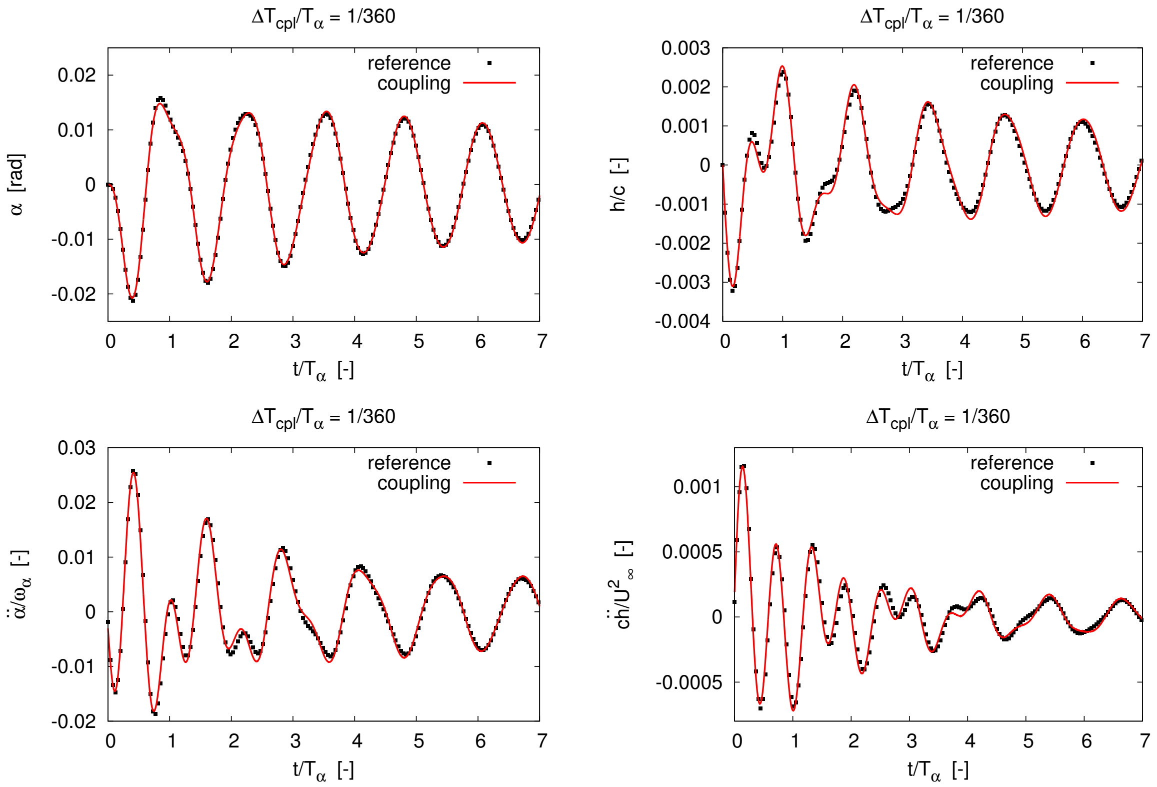


Figure F.18: Case $U_\infty = 0.9U_{flt}$ with $T_\alpha = 0.1795$s, $c = 0.46$m, $\omega_\alpha = 35$rad/s, $U_\infty = 37.45$m/s.

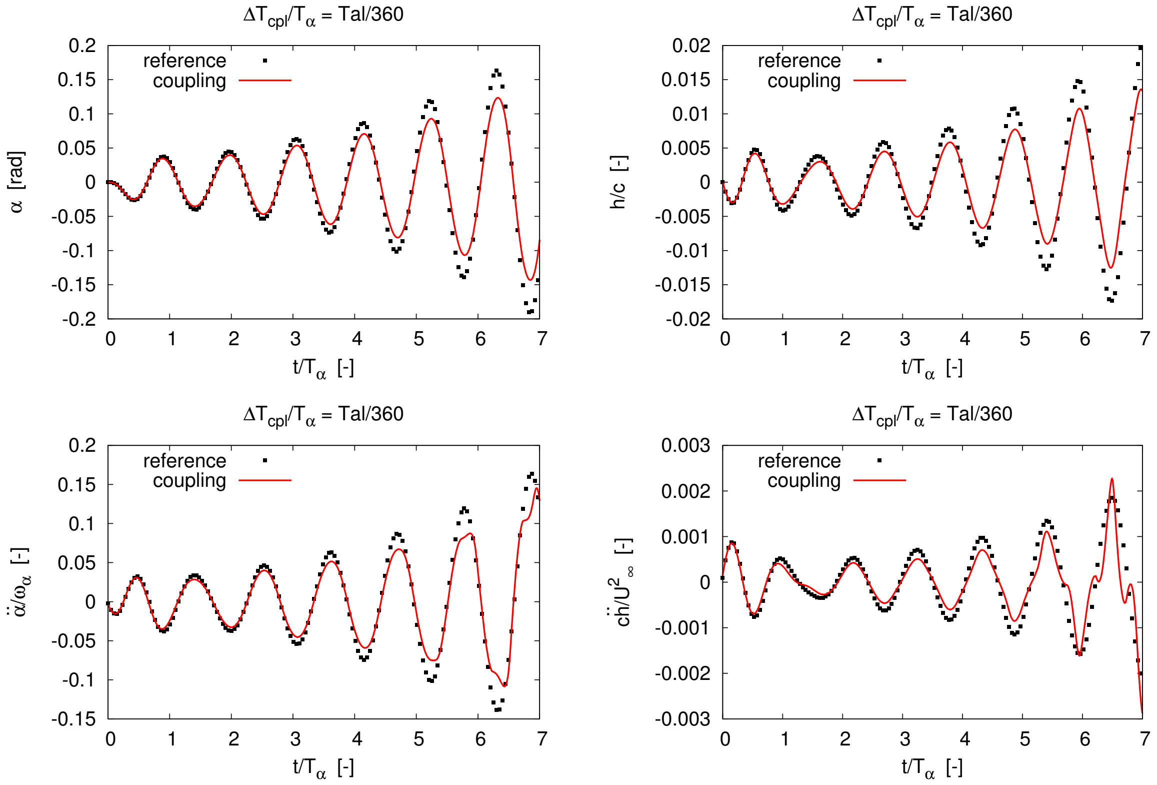


Figure F.19: Case $U_\infty = 1.1U_{flt}$ with $T_\alpha = 0.1795$s, $c = 0.46$m, $\omega_\alpha = 35$rad/s, $U_\infty = 45.77$m/s.

F.7. PowerFLOW®-Simpack® coupling, 2 DoFs, #3 and #4

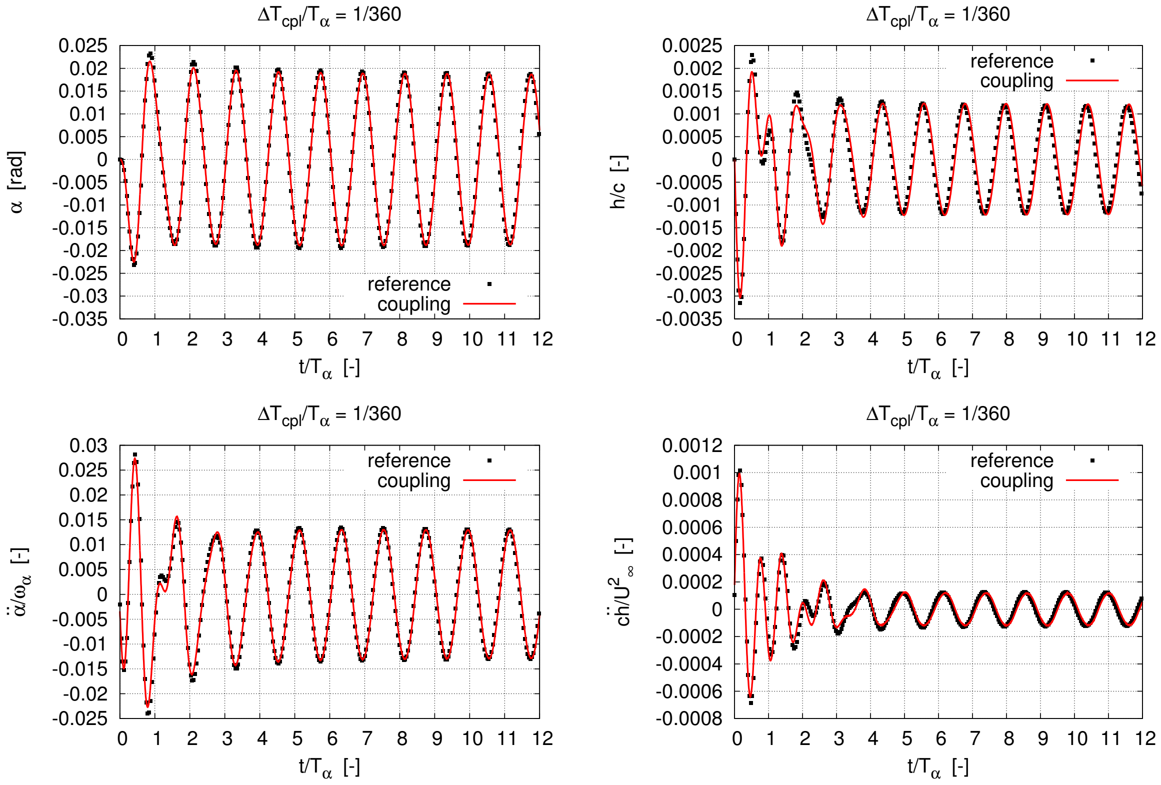


Figure F.20: Case 99% the of V-g flutter velocity with $T_\alpha = 0.1795\mathrm{s}$, $c = 0.46\mathrm{m}$, $\omega_\alpha = 35\mathrm{rad/s}$, $U_\infty = 41.20\mathrm{m/s}$.

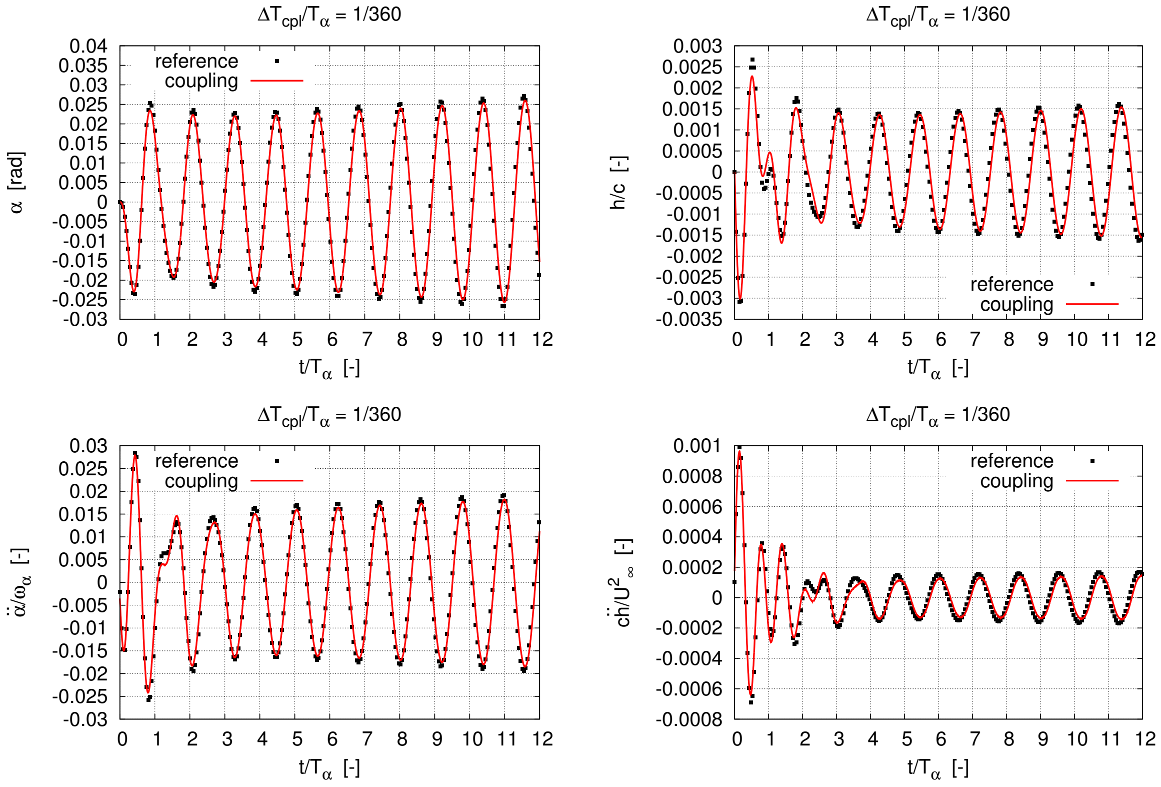


Figure F.21: Case 101% he of V-g flutter velocity with $T_\alpha = 0.1795\mathrm{s}$, $c = 0.46\mathrm{m}$, $\omega_\alpha = 35\mathrm{rad/s}$, $U_\infty = 42.03\mathrm{m/s}$.

References

[1] L. Wang, X. Liu, and A. Kolios, *State of the art in the aeroelasticity of wind turbine blades: Aeroelastic modelling,* Renewable and Sustainable Energy Reviews **64**, 195 (2016).

[2] H. Bijl, A. v. Zuijlen, A. d. Boer, and D. J. Rixen, *Fluid-structure interaction (wb1417) - An introduction to numerical coupled simulation. Lecture Notes* (TU Delft, Faculty of Aerospace Engineering (Aerodynamics Department) and Faculty of Mechanical, Maritime and Materials Engineering, 2007).

[3] C. Farhat and M. Lesoinne, *Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems,* Computer Methods in Applied Mechanics and Engineering **182**, 499 (2000).

[4] C. Farhat, M. Lesoinne, P. Stern, and S. Lantéri, *High performance solution of three-dimensional nonlinear aeroelastic problems via parallel partitioned algorithms: methodology and preliminary results,* Advances in Engineering Software **28**, 43 (1997).

[5] D. Casalino, *Lattice-Boltzmann – Very Large Eddy simulations for airframe and engine noise application, presentation slides (unpublished)* (Exa Corporation, Roma 3 University, 2015).

[6] G. Romani and D. Casalino, *Powerflow simulations of helicopters in blade-vortex interaction conditions,* in *75th Vertical Flight Society Annual Forum and Technology Display* (2019).

[7] J. G. Leishman, *Principles of Helicopter Aerodynamics*, Cambridge Aerospace Series (Cambridge University Press, 2006).

[8] S. J. Hulshoff, *AE4930 Aeroelasticity, course Reader* (Faculty of Aerospace Engineering, TU Delft, Delft, 2012).

[9] M. Potsdam, H. Yeo, and W. Johnson, *Rotor airloads prediction using loose aerodynamic/structural coupling,* Journal of Aircraft **43**, 732 (2006).

[10] T. Burton, N. Jenkins, D. Sharpe, and E. Bossanyi, *Aerodynamics of horizontal axis wind turbines,* in *Wind Energy Handbook*, Wiley Online Books (Wiley, 2011).

[11] M. O. L. Hansen and H. A. Madsen, *Review paper on wind turbine aerodynamics,* Journal of Fluids Engineering **133**, 114001 (2011).

[12] B. Sanderse, S. P. Pijl, and B. Koren, *Review of computational fluid dynamics for wind turbine wake aerodynamics,* Wind Energy **14**, 799 (2011).

[13] J. Hofmann, L. Krause, M. Mindt, S. Surrey, and M. Graser, *Rotor simulation and multi body systems: Coupling of helicopter aerodynamics with simpack®,* in *63. Deutscher Luft- und Raumfahrtkongress* (2014).

[14] G. Romani, *Sviluppo di una tecnica di soluzione per l'analisi aeroacusto-elastica di rotori di elicottero in avanzamento (Development of a solution method for the aeroacustic and aeroelastic analysis of helicopters in forward flight)*, Msc thesis (2017).

[15] B. G. van der Wall, J. W. Lim, M. J. Smith, S. N. Jung, J. Bailly, J. D. Baeder, and D. D. Boyd, *The hart ii international workshop: an assessment of the state-of-the-art in comprehensive code prediction,* CEAS Aeronautical Journal **4**, 223 (2013).

[16] M. J. Smith, J. W. Lim, B. G. van der Wall, J. D. Baeder, R. T. Biedron, D. D. Boyd, B. Jayaraman, S. N. Jung, and B.-Y. Min, *The hart ii international workshop: an assessment of the state of the art in cfd/csd prediction,* CEAS Aeronautical Journal **4**, 345 (2013).

[17] S. G. Horcas, M. H. A. Madsen, N. N. Sørensen, and F. Zahle, *Suppressing vortex induced vibrations of wind turbine blades with flaps,* in *Recent Advances in CFD for Wind and Tidal Offshore Turbines*, edited by M. A. Ferrer, E. (2019) Book section 2, pp. 11–24.

[18] M. Sayed, T. Lutz, E. Krämer, S. Shayegan, and R. Wüchner, *Aeroelastic analysis of 10 mw wind turbine using cfd–csd explicit fsi-coupling approach,* Journal of Fluids and Structures **87**, 354 (2019).

[19] G. Romani and D. Casalino, *Rotorcraft blade-vortex interaction noise prediction using the lattice-boltzmann method,* Aerospace Science and Technology **88**, 147 (2019).

[20] H.-J. Bungartz and M. Schäfer, *Fluid-Structure Interaction. Modelling, Simulation, Optimisation*, Lecture Notes in Computational Science and Engineering, Vol. 53 (Springer-Verlag Berlin Heidelberg, Springer-Verlag Berlin Heidelberg, 2006).

[21] P. M. Michaelsen, *A Fluid-Structure-Interaction Simulation tool for application in rotating machinery*, Phd thesis (2015).

[22] C. A. Felippa, K. C. Park, and M. R. Ross, *A classification of interface treatments for fsi,* in *Fluid Structure Interaction II*, Lecture Notes in Computational Science and Engineering„ Vol. 73, edited by M. M. Bungartz, H.-J. and M. Schäfer (Springer Berlin Heidelberg, Berlin, Heidelberg, 2010) pp. 27–51.

[23] C. A. Felippa, K. C. Park, and C. Farhat, *Partitioned analysis of coupled mechanical systems,* Computer Methods in Applied Mechanics and Engineering **190**, 3247 (2001).

[24] W. A. Wall, A. Gerstenberger, P. Gamnitzer, C. Förster, and E. Ramm, *Large deformation fluid-structure interaction – advances in ale methods and new fixed grid approaches,* in *Fluid-Structure Interaction*, Lecture Notes in Computational Science and Engineering, Vol. 53, edited by H.-J. Bungartz and M. Schäfer (Springer Berlin Heidelberg, Berlin, Heidelberg, 2006) pp. 195–232.

[25] A. H. van Zuijlen and H. Bijl, *Multi-level accelerated sub-iterations for fluid-structure interaction,* in *Fluid Structure Interaction II*, Lecture Notes in Computational Science and Engineering, Vol. 73, edited by M. M. Bungartz, H.-J. and M. Schäfer (Springer Berlin Heidelberg, Berlin, Heidelberg, 2010) pp. 1–25.

[26] C. Wood, A. J. Gil, O. Hassan, and J. Bonet, *A partitioned coupling approach for dynamic fluid–structure interaction with applications to biological membranes,* International Journal for Numerical Methods in Fluids **57**, 555 (2008).

[27] Q. Zhang and T. Hisada, *Studies of the strong coupling and weak coupling methods in fsi analysis,* International Journal for Numerical Methods in Engineering **60**, 2013 (2004).

[28] C. Kassiotis, A. Ibrahimbegovic, R. Niekamp, and H. G. Matthies, *Nonlinear fluid–structure interaction problem. part i: implicit partitioned algorithm, nonlinear stability proof and validation examples,* Computational Mechanics **47**, 305 (2011).

[29] C. Farhat, *Parallel and distributed solution of coupled nonlinear dynamic aeroelastic response problems,* in *Parallel Solution Methods in Computational Mechanics*, edited by M. Papadrakakis (J. Wiley Sons, Chichester„ 1997).

[30] S. Piperno, C. Farhat, and B. Larrouturou, *Partitioned procedures for the transient solution of coupled aroelastic problems part i: Model problem, theory and two-dimensional application,* Computer Methods in Applied Mechanics and Engineering **124**, 79 (1995).

[31] L. D. Moerloose, L. Taelman, P. Segers, J. Vierendeels, and J. Degroote, *Analysis of several subcycling schemes in partitioned simulations of a strongly coupled fluid-structure interaction,* International Journal for Numerical Methods in Fluids **89**, 181 (2019).

[32] D. R. J. Owen, C. R. Leonardi, and Y. T. Feng, *An efficient framework for fluid–structure interaction using the lattice boltzmann method and immersed moving boundaries,* International Journal for Numerical Methods in Engineering **87**, 66 (2013).

[33] S. Geller, S. Kollmannsberger, M. E. Bettah, M. Krafczyk, D. Scholz, A. Düster, and E. Rank, *An explicit model for three-dimensional fluid-structure interaction using lbm and p-fem,* in *Fluid Structure Interaction II*, Lecture Notes in Computational Science and Engineering, Vol. 73, edited by M. M. Bungartz, H.-J. and M. Schäfer (Springer Berlin Heidelberg, Berlin, Heidelberg, 2010) pp. 285–325.

[34] A. de Boer, A. H. van Zuijlen, and H. Bijl, *Review of coupling methods for non-matching meshes,* Computer Methods in Applied Mechanics and Engineering **196**, 1515 (2007).

[35] S. Chen and G. D. Doolen, *Lattice boltzmann method for fluid flows,* Annual Review of Fluid Mechanics **30**, 329 (1998).

[36] R. Benzi, S. Succi, and M. Vergassola, *The lattice boltzmann equation: theory and applications,* Physics Reports **222**, 145 (1992).

[37] Y. B. Bao and J. Meskas, *Lattice Boltzmann method for fluid simulations*, Report Report 930 (Department of Mathematics, Courant Institute of Mathematical Sciences, New York University, 2011).

[38] D. A. Perumal and A. K. Dass, *A review on the development of lattice boltzmann computation of macro fluid flows and heat transfer,* Alexandria Engineering Journal **54**, 955 (2015).

[39] H.-Q. Xie, Z. Zeng, and L.-Q. Zhang, *Three-dimensional multi-relaxation-time lattice boltzmann front-tracking method for two-phase flow,* Chinese Physics B **25**, 014702 (2016).

[40] J. Wang, L. Chen, Q. Kang, and S. S. Rahman, *The lattice boltzmann method for isothermal micro-gaseous flow and its application in shale gas flow: A review,* International Journal of Heat and Mass Transfer **95**, 94 (2016).

[41] X. Nie, X. Shan, and H. Chen, *A lattice-boltzmann / finite-difference hybrid simulation of transonic flow,* in *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, Aerospace Sciences Meetings (American Institute of Aeronautics and Astronautics).

[42] P. L. Bhatnagar, E. P. Gross, and M. Krook, *A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component systems,* Physical Review **94**, 511 (1954).

[43] H. Chen, S. Kandasamy, S. Orszag, R. Shock, S. Succi, and V. Yakhot, *Extended boltzmann kinetic equation for turbulent flows,* Science **301**, 633 (2003).

[44] C. M. Teixeira, *Incorporating turbulence models into the lattice-boltzmann method,* International Journal of Modern Physics C **09**, 1159 (1998).

[45] G. Brès, F. Pérot, and D. Freed, *Properties of the lattice boltzmann method for acoustics,* in *15th AIAA/CEAS Aeroacoustics Conference (30th AIAA Aeroacoustics Conference)*, Aeroacoustics Conferences (American Institute of Aeronautics and Astronautics, 2009).

[46] A. T. Hsu, T. Yang, I. Lopez, and A. Ecer, *A review of lattice boltzmann models for compressible flows,* in *Parallel Computational Fluid Dynamics 2003*, edited by S. N. P. J. Ecer, A. and P. Fox (Elsevier, Amsterdam, 2004) pp. 19–28.

[47] S. Chapman, T. G. Cowling, D. Burnett, and C. Cercignani, *The Mathematical Theory of Non-uniform Gases: An Account of the Kinetic Theory of Viscosity, Thermal Conduction and Diffusion in Gases*, Cambridge Mathematical Library (Cambridge University Press, 1990).

[48] H. Chen, C. Teixeira, and K. Molvig, *Digital physics approach to computational fluid dynamics: Some basic theoretical features,* International Journal of Modern Physics C **08**, 675 (1997).

[49] D. Casalino, F. Avallone, I. Gonzalez-Martino, and D. Ragni, *Aeroacoustic study of a wavy stator leading edge in a realistic fan/ogv stage,* Journal of Sound and Vibration **442**, 138 (2019).

[50] V. Yakhot, S. A. Orszag, S. Thangam, T. B. Gatski, and C. G. Speziale, *Development of turbulence models for shear flows by a double expansion technique,* Physics of Fluids A: Fluid Dynamics **4**, 1510 (1992).

[51] C. G. Alexander, H. Chen, S. Kandasamy, R. A. Shock, and S. R. Govindappa, *Simulations of engineering thermal turbulent flows using a lattice boltzmann based algorithm,* American Society of Mechanical Engineers, Pressure Vessels and Piping Division **424**, 115 (2000).

[52] *PowerCASE User's Guide, Release 6.0* (Dassault Systemes).

[53] *PowerFLOW Command-Line Reference, Release 6.0* (Dassault Systemes).

[54] A. Bedford and W. L. Fowler, *Engineering Mechanics, Dynamics*, 5th ed. (Prentice Hall, New Jersey, 2008).

[55] J. L. Meriam and L. G. Kraige, *Engineering Mechanics volume 2 Dynamics* (John Wiley and Sons, New York, 1987).

[56] S. I. Newton, *Philosophiae Naturalis Principia Mathematica* (Royal Society, London, 1687).

[57] P. Masarati, P. Mantegazza, F. Resta, and A. Curami, *Dinamica dei Sistemi Aerospaziali, dispense del corso (Dynamics of aerospace systems, lecture notes)* (Dipartimento di Scienze e Tecnologie Aerospaziali, Politecnico di Milano, Milan, 2015).

[58] M. Chasles, *Note sur les propriétés générales du système de deux corps semblables entr'eux et placés d'une manière quelconque dans l'espace; et sur le déplacement fini ou infiniment petit d'un corps solide libre (note on the general properties of the system of two similar bodies between them and placed in some way in space; and on the finite or infinitely small displacement of a free solid body),* Bulletin des Sciences Mathematiques, Astronomiques, Physiques et Chimiques **14**, 321–326 (1830).

[59] P. Mantegazza and P. Masarati, *Analysis of Systems of Differential-Algebraic Equations (DAE)*, Graduate Course on "Multibody System Dynamics" PhD (Aerospace Engineering Department. Politecnico di Milano., Milan, 2015).

[60] A. Martin, *DAE aspects of multibody system dynamics*, Report 1 (Martin Luther University Halle-Wittenberg, Institut fuer Mathematik, 2016).

[61] *Simpack Basic Training, Course Material* (Simpack GmbH, a Dassault Systemes company).

[62] R. L. Huston, L. Y. Sheng, and L. chengqun, *Use of absolute coordinates in computational multibody dynamics,* Computers Structures **52**, 17 (1994).

[63] A. Quarteroni and F. Saleri, *Calcolo scientifico: Esercizi e problemi risolti con MATLAB e Octave (Scientific computing: exercises and solved problems with Matlab and Octave)*, UNITEXT (Springer Milan, 2009).

[64] T. J. Ypma, *Historical development of the newton–raphson method,* SIAM review **37**, 531 (1995).

[65] *Simpack 2019.x User Guide* (Dassault Systemes, 2019).

[66] *Description and Use of LSODE, the Livermore Solver for Ordinary Differential Equations*, Report 1327 (NASA, 1993).

[67] J. M. Jonkman, G. J. Hayman, B. J. Jonkman, and R. R. Damiani, *AeroDyn v15 User's Guide and Theory Manual* (NREL).

[68] K. McCloghrie and M. Rose, *Management Information Base for network management of TCP/IP-based internets: MIB-II*, Report 2070-1721 (1991).

[69] *Simpack User Defined Routines Training, Course Material* (Simpack GmbH, a Dassault Systemes company).

[70] *Simpack Scripting Training, Course Material* (Simpack GmbH, a Dassault Systemes company).

[71] S. Stefanov, *JavaScript Patterns: Build Better Applications with Coding and Design Patterns* (O'Reilly Media, 2010).

[72] M. Robinson, S. D. R. Galbraith, and M. Miller, *Unsteady aerodynamics of wind turbines,* in *33rd Aerospace Sciences Meeting and Exhibit*, Aerospace Sciences Meetings (American Institute of Aeronautics and Astronautics, 1995).

[73] W. J. McCroskey, *The Phenomenon of Dynamic Stall*, Report 81264 (NASA, 1981).

[74] E. Buckingham, *On physically similar systems; illustrations of the use of dimensional equations,* Physical Review **4**, 345 (1914).

[75] H. Wagner, *Über die entstehung des dynamischen auftriebes von tragflügeln (about the origin of the dynamic lift of wings),* ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik **5**, 17 (1925).

[76] H. G. Küssner, *Zusammenfassender bericht tiber den instationären auftrieb von flügeln (summary report about the unsteady lift of wings),* Luftfahrtforschung **13**, 410 (1935).

[77] T. H. V. Karman and W. R. Sears, *Airfoil theory for non-uniform motion,* Journal of the Aeronautical Sciences **5**, 379 (1938).

[78] T. Theodorsen, *General Theory of Aerodynamic Instability and the Mechanism of Flutter*, Report 496 (NACA, 1935).

[79] R. G. Loewy, *A two-dimensional approximation to the unsteady aerodynamics of rotary wings,* Journal of the Aeronautical Sciences **24**, 81 (1957).

[80] J. P. Jones, *The influence of the wake on the flutter and vibration of rotor blades,* Aeronautical Quarterly **9**, 258 (1958).

[81] I. H. A. Abbott and A. E. V. Doenhoff, *Theory of wing sections, including a summary of airfoil data* (Dover Publications, 1959).

[82] N. Tatomir, *Efficacy of Linearized Aerodynamic CFD Modeling for Flutter Computation*, Msc thesis (2019).

[83] S. L. Brunton and C. W. Rowley, *Empirical state-space representations for theodorsen's lift model,* Journal of Fluids and Structures **38**, 174 (2013).

[84] G. Romani and C. Poggi, *Aeroelasticità, esercitazioni I-II (Aeroelasticity assignments I-II) [unpublished]*, Report (Roma 3 university, 2015).

[85] R. L. Halfman, *Experimental aerodynamic derivatives of a sinusoidally oscillating airfoil in two-dimensional flow*, Report 1108 (NACA, 1952).

[86] V. Motta, A. Guardone, and G. Quaranta, *Influence of airfoil thickness on unsteady aerodynamic loads on pitching airfoils,* Journal of Fluid Mechanics **774**, 460 (2015).

[87] *PowerFLOW User's Guide, Release 6.0* (Dassault Systemes).

[88] M. M. Arnold, *Simulation and Evaluation of the Hydroelastic Responses of a Tidal Current Turbine*, Phd thesis (2016).

[89] G. Romani and D. Casalino, *Application of lattice-boltzmann method to rotorcraft aerodynamics and aeroacoustics,* in *43rd European Rotorcraft Forum, ERF 2017*, Vol. 2 (2017) pp. 816–828.

[90] A. Collette, *Python and HDF5, Unlocking Scientific Data* (O'Reilly Media, 2013).

[91] E. A. Tal, N. T. Nguyen, and E. Ting, *Comparison of unsteady aerodynamics approximations for time-domain representation of frequency-independent aeroelastic state-space models,* in *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA SciTech Forum (American Institute of Aeronautics and Astronautics).

[92] P. Omenzetter, K. Wilde, and Y. Fujino, *Suppression of wind-induced instabilities of a long span bridge by a passive deck-flaps control system: Part ii: Numerical simulations,* Journal of Wind Engineering and Industrial Aerodynamics **87**, 81 (2000).

[93] I. E. Garrick, *Operational treatment of the non-uniform lift theory in airplane dynamics*, Report 667 (NASA, 1938).

[94] R. T. Jones, *On some reciprocal relations in the theory of nonstationary flows*, Report 629 (NACA, 1938).

[95] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice* (Addison-Wesley Professional, 2012) p. 640.