Development of advanced tools in computational nonlinear aeroelasticity

Einar Hope Kleppenes

Master's thesis in Energy
Geophysical Institute
University of Bergen

January 4, 2023

Acknowledgements

This 60 ECTS master thesis is written as a part of the Integrated Masters programme in Energy at the Geophysical Institute, University of Bergen.

I would like to thank my supervisors Cristian Guillermo Gebhardt and Bruno Antonio Roccia for the guidance and support during the time working on this thesis. I am grateful for being given the opportunity to write a thesis relevant to my interests. I would also like to thank you for taking the time helping me through all the meetings over the past year, working on this thesis.

Einar Hope Kleppenes, Bergen

Abstract

In this work, the development of a two dimensional computational model for predicting the aeroelastic response is presented. In the aeroelastic model presented here, the aeroedynamic and the structural models are considered elements of a single dynamic system. The aerodynamic model is based on the unsteady vortex particle method , while the structural model is based on the paired Adam Bashforth - Adam Moulton second order numerical predictor corrector method.

The unsteady vortex particle method is used for computing the aerodynamic forces coming from the free stream velocity and for covecting the wake particles. Usually the wake particles are convected using the explicit Euler method in the wake scheme, which is fast and easy to implement. Here, the explicit linear multistep Adam Bashforth methods are implemented in the wake scheme part of the aerodynamic model. Contrary to the explicit Euler method, these methods depends on the velocities from previous time steps.

The nonlinear analysis of the aeroelastic system is more complex and makes it harder to predict the aeroelastic flutter compared to the linear analysis. For this reason, a neural network generated function for computing the correct force and moment terms in the structural model is realized and compared to the analytical linear and nonlinear spring force and moment response. In this thesis, a cubic spring term is added in the structural model of the aeroelastic system. The experimental results using the trained neural network generated spring have shown similar results compared to the case using the analytical expression for the spring, for both the linear and nonlinear structural model.

The results presented here illustrate how one can utilize a relatively simple neural network in order to train and predict the subcritical, critical and supercritical flow for the aeroelastic model, when adding a cubic nonlinearity in the structural model of the system.

Contents

Acknowledgements i

Abstract ii

0.1 Background . . . 1
0.2 Wind energy . . . 2
0.2.1 Development of wind turbines . . . 2
0.2.2 Wind power production . . . 2
0.2.3 Aeroelastic flutter . . . 3
0.2.4 Aerodynamic model . . . 4
0.2.5 Structural model . . . 5
0.3 Objectives . . . 5
0.4 Thesis outline . . . 5

1 General aerodynamics **6**
1.1 The governing equations . . . 6
1.1.1 Navier - Stokes equations . . . 6
1.1.2 Continuity equation . . . 6
1.1.3 Potential flow conditions . . . 7
1.1.4 Laplace's equation . . . 9
1.1.5 Incompressible Euler's equations . . . 9
1.1.6 Boundary conditions . . . 9
1.1.7 Vortex point solution . . . 10
1.1.8 Biot - Savart law . . . 11
1.1.9 Kelvins theorem . . . 11
1.1.10 The Kutta - Joukowski theorem . . . 12
1.1.11 The unsteady Bernoullis equation . . . 13
1.2 Thin airfoil theory . . . 13
1.2.1 Continuous vortex sheet approximation . . . 14
1.2.2 Non penetration boundary condition . . . 14
1.2.3 Kuttas condition . . . 15
1.2.4 Fundamental equation of thin airfoil theory . . . 16
1.2.5 Analytical coefficients . . . 16

2 Aerodynamic model **18**
2.1 Superposition principle . . . 18
2.1.1 Vortex point solution . . . 18
2.2 Airfoil panel discretization . . . 20
2.3 Influence matrix . . . 22
2.4 Computation of the aerodynamic loads . . . 23

2.5 Wake particle convection 25

3 Structural model **27**
3.1 Physical system 27
3.1.1 Plunging airfoil motion 28
3.1.2 Pitching airfoil motion 28
3.2 Equations of motion 29
3.2.1 Neural Network generated spring 32

4 Aeroelastic model **35**
4.1 Simulation scheme 35
4.2 Simulation scheme - steps 35
4.3 Numerical integration schemes 37
4.3.1 Wake model 37
4.3.2 Structural model 37

5 Results and Discussion **40**
5.1 Results 40
5.2 Numerical verification 40
5.2.1 Aerodynamic model 40
5.2.2 Structural model 53
5.2.3 Aeroelastic model 55
5.3 Numerical experiments 58
5.3.1 Aerodynamic model 59
5.3.2 Aeroelastic model 68

6 Conclusions and Future Work **83**

List of Figures

1 The total annual green house gas emissions per year by world regions (*Ritchie et al.* (2017)). . . . 1
2 Figure showing wind power generation in the net zero scenario 2010 - 2030, from *IEA* . . . 3

1.1 Control area for a flow . . . 7
1.2 Illustration of invscid vs non inviscid flow . . . 8
1.3 Boundary conditions at infinity. . . . 10
1.4 Development of circulation for closed curve around airfoil. . . . 12
1.5 Cambered airfoil - description . . . 14
1.6 Approximation of the airfoil by a vortex sheet . . . 14
1.7 The angle of attack for an airfoil, with horizontal free stream velocity . . . 15
1.8 The kutta condition - illustration for an airfoil . . . 15

2.1 Plot of the induced velocity from a vortex singularity as a function of the distance r to the point where the velocity is induced. . . . 19
2.2 Discretization of the airfoil, using discrete vortex particles on the airfoil and the wake points. . . . 20
2.3 airfoil coordinates - vortex point and collocation point. . . . 21
2.4 Pressure force decomposition of the airfoil. . . . 25
2.5 Flow chart of the unsteady vortex method. . . . 26

3.1 Structural model - details. . . . 27
3.2 plunging and pitching airfoil illustration. . . . 28
3.3 Neural network, IL = Input Layer, HL = Hidden Layer and OL = Ouput Layer. . 33

4.1 Coupling between the structural model and the unsteady vortex particle method. 36

5.1 precision quotients q for the wake schemes. . . . 42
5.2 Plot of the numerical lift coefficient vs analytical lift coefficient on the left, plot of the drag coefficient on the right. . . . 43
5.3 Plot of the steady pressure coefficient as a function of the airfoil coordinates in the left figure. The right figure is the steady lift coefficient plotted against the angles of attack in the small angle domain. . . . 44
5.4 Plot of the numerical Cp and the analytical Cp, for both 10 and 20 panels used in the discretization of the airfoil. . . . 44
5.5 Garricks solution vs numerical solution, plunging case, $k_h = 0.1$ . . . 46
5.6 Garricks solution vs numerical solution, plunging case, $k_h = 0.25$. . . . 47
5.7 Garrick's solution vs numerical solution, plunging case, $k_h = 0.5$. . . . 48
5.8 Cl error vs number of panels - different wake schemes, plunging case . . . 49
5.9 Garricks solution vs numerical solution, pitching case, k=0.1 . . . 50

5.10 Garricks solution vs numerical solution, pitching case, k=0.25 51
5.11 Garricks solution vs numerical solution, pitching case, k=0.5 52
5.12 Cl error vs number of panels - different wake schemes, pitching case. 53
5.13 Rotating pendulum. 54
5.14 Numerical vs analytical solution to the linear pendulum and the error plot between the numerical and analytical solution. 55
5.15 Plunging motion - subcritical, critical and supercritical velocities. 56
5.16 Pitching motion - subcritical, critical and supercritical velocities. 57
5.17 Critical velocities for the 20 and 30 number of panels discretization of the suspension bridge 58
5.18 CL vs angles of attack - for the different wake schemes. 59
5.19 Comparison between the wake shapes for the different schemes. 60
5.20 Evolution of the HDD for the wake vortices steady flow - comparison between the wake schemes, with the euclidean norm as the distance function. 62
5.21 Evolution of the HDD for the wake vortices steady flow - comparison between the wake schemes, with the infinity norm as distance function. 63
5.22 Evolution of the HDD for the wake vortices, added pitching motion - comparison between the wake schemes, with the euclidean norm as distance function. . 64
5.23 Evolution of the HDD for the wake vortices, added pitching motion - comparison between the wake schemes, with the infinity norm as distance function. . . 65
5.24 Evolution of the HDD for the wake vortices, added plunging motion - comparison between the wake schemes, with the euclidean norm as distance function. . 66
5.25 Evolution of the HDD for the wake vortices, added plunging motion - comparison between the wake schemes, with the infinity norm as distance function. . . 67
5.26 Plot of the subcritical, critical and pitching motion for the nonlinear aeroelastic model 69
5.27 Plot of subcritical, critical and supercritical plunging motion for the nonlinear aeroelastic model. 70
5.28 Plot showing the linear neural network spring term for pitching and plunging vs the analytical spring term, given some random input data for testing. 71
5.29 Plot showing the nonlinear neural network spring term for pitching and plunging vs the analytical nonlinear spring term, given some random input data for testing. 72
5.30 Plot of the linear analytical spring vs the linear neural network sspring, pitching and pluniging, $\alpha = \frac{\pi}{180}$ 73
5.31 Plot of the linear analytical spring vs the linear neural network sspring, pitching and pluniging, $\alpha = \frac{5\pi}{180}$ 73
5.32 Plot of the linear analytical spring vs the linear neural network spring, pitching and plunging, $\alpha = \frac{10\pi}{180}$ 74
5.33 Plot of the linear analytical spring vs the linear neural network spring, pitching and plunging, $\alpha = \frac{\pi}{180}$ 75
5.34 Plot of the linear analytical spring vs the linear neural network spring, pitching and plunging, $\alpha = \frac{5\pi}{180}$ 75
5.35 Plot of the linear analytical spring vs the linear neural network spring, pitching and plunging, $\alpha = \frac{10\pi}{180}$ 76
5.36 Plot of the linear analytical spring vs the linear neural network spring, pitching and plunging, $\alpha = \frac{\pi}{180}$ 77
5.37 Plot of the linear analytical spring vs the linear neural network spring, pitching and plunging, $\alpha = \frac{5\pi}{180}$ 77

5.38 Plot of the linear analytical spring vs the linear neural network spring, pitching and plunging, $\alpha = \frac{10\pi}{180}$ 78
5.39 Plot of the nonlinear analytical spring vs the nonlinear neural network spring, pitching and plunging, $\alpha = \frac{\pi}{180}$ 79
5.40 Plot of the nonlinear analytical spring vs the nonlinear neural network spring, pitching and plunging, $\alpha = \frac{\pi}{180}$ 79
5.41 Plot of the nonlinear analytical spring vs the nonlinear neural network spring, pitching and plunging, $\alpha = \frac{\pi}{180}$ 80
5.42 Comparison of the wake shapes for the analytical linear spring and neural network generated linear spring. 81
5.43 Comparison of the wake shapes for the analytical nonlinear spring and neural network generated nonlinear spring. 82

Introduction

0.1 Background

The current global energy demand is increasing rapidly. At the same time, the total net anthropogenic greenhouse gas emissions have kept rising during the period 2010 to 2019 and the cumulative net CO2 emissions have been rising since the year 1850 (*IPCC*). Although the average greenhouse gas emissions were higher in the 2010s compared to any previous decades, the growth rate of greenhouse gas emissions in the 2010s was less when compared to the growth rate in the 2000s (*IPCC*). As one can observe from Fig. 1, the total annual greenhouse gas emissions per year has been increasing rapidly, especially since the 1950s. The transformation of the global energy system needs to accelerate substantially to meet the objectives of the Paris agreement, which aim to keep the rise in average global temperatures closer to 1.5 °C in the present century, when compared to pre-industrial levels. Therefore, to meet todays global energy demand while reducing carbon emissions, we must move to cleaner and greener power generation.

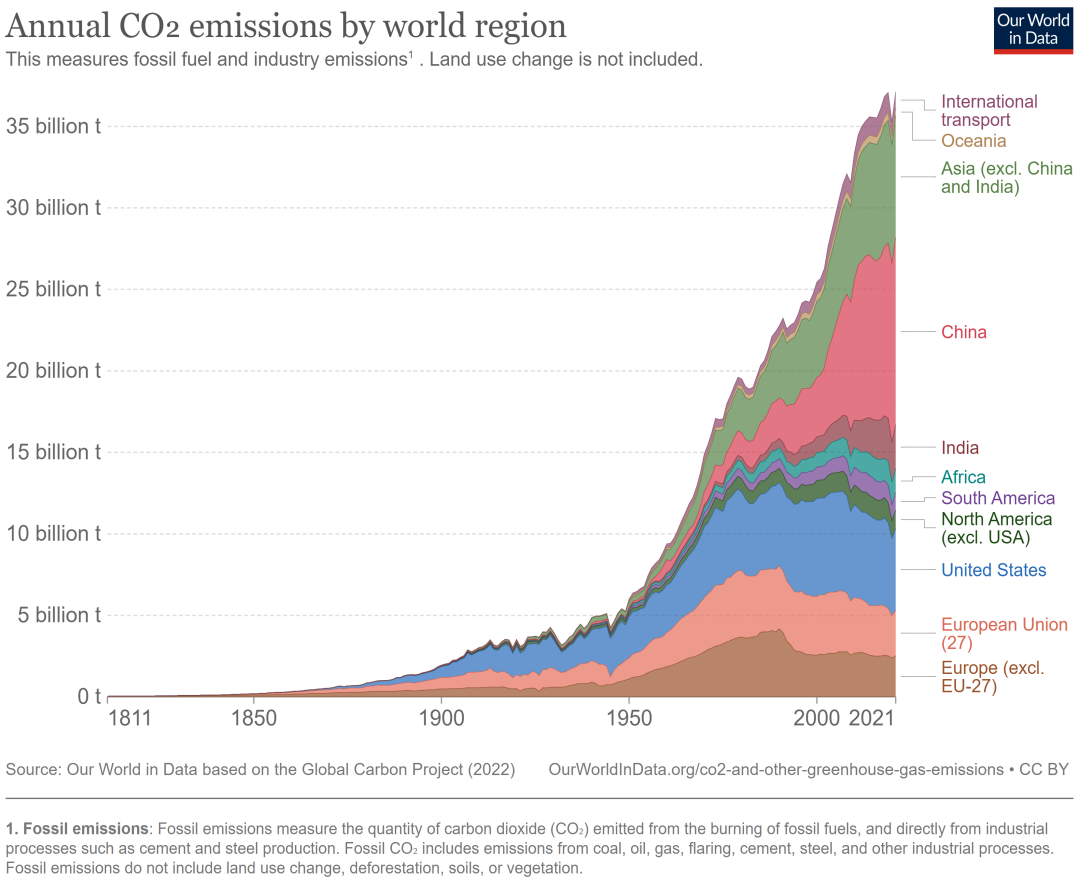

Figure 1: The total annual green house gas emissions per year by world regions (Ritchie et al. (2017)).

The global weighted average cost of electricity (GWACE) from all commercially available renewable power generation technologies continued to fall in 2018. For projects related to onshore wind commissioned in 2018, the GWACE reached a minimum of USD 0.056 per kilowatt-hour (kWh), a 13% decrease from 2017 and a 35% drop in price compared to 2010. The global weighted-average levelised cost of electricity for onshore wind was at USD 0.039 kWh in 2020 (*IRENA*). As the price of electricity produced from onshore wind has been steadily dropping in the 2010s, it is starting to become competitive against fossil-fueled produced electricity (*IRENA* (2019)).

Wind power has the potential to supply more than one third of the total electricity demand by 2050 (*IRENA* (2019)) and is well aligned with energy transformation scenarios conducted by numerous institutions. It highlights the importance of scaling the wind power generation in order to mitigate carbon gas emissions and decarbonise the energy system in the next three decades (*IRENA* (2019)). A combination of decreasing prices in electricity costs, low operating costs (*IRENA*) and the fact that wind energy is an abundant clean natural resource makes it a great candidate in the transition from from fossil fuels to renewable energy.

0.2 Wind energy

0.2.1 Development of wind turbines

The use of wind as a source of energy dates as far back as to the ancient Egypt (*Pasqualetti* (2014)), where the egyptians used the power of the wind to propel sailing crafts (made of linen and papyrus). Around the 10th century AD in Persia, primitive windmills were used to grind grain and lifting water from streams to irrigate gardens. It was not until 1886 that the scientist Charles Brush invented the first practical large-scale wind turbine (*Pasqualetti* (2014)).

Over the years, the development of wind power has followed several trends. In particular, over the past twenty-five years, the size of wind turbines has increased dramatically, from approximately 50 kW of power and rotor diameters of 15 to 20 meters to today's commercial machines of 14 MW and rotor diameters of 185 meters. The increase of the rotor size is mainly driven by cost reductions. Bigger wind turbines are able to extract more energy compared to smaller ones, which makes it possible to extract more energy per turbine, leading to a reduction in manufacturing and installation costs. Such technological progress has meant a substantial change in the context of numerical models for the design of wind turbines. Specifically, Standard aerodynamic calculations, where a profile of constant spatial and temporal velocities was assumed, had to be replaced with unsteady aerodynamic models, capable of being integrated with structural models to simulate the aeroelastic behavior of the entire turbine.

0.2.2 Wind power production

According to the global wind report (*GWEC*), the global cumulative wind power capacity had risen to 837 GW, where 93.6 GW of new installations were added since 2021, showing a year over year growth of 12%. However, reaching the goal of net zero emissions by 2050 requires having installed approximately 7900 TWh of wind power by 2030 (see Fig. 2). This progression means increasing average annual capacity to 250 GW, which is more than double the record growth of 2020 (*IEA*). Achieving 7900 TWh by 2030 requires increasing wind power generation by 18% per year during 2022-2030 (*IEA*). Although large-scale wind turbines have increased their efficiency over the past 40 years, the cost and maintenance of these turbines are still relatively high.

A wind turbine is a huge rotating machine that converts the kinetic energy contained in the wind (air in motion) into a torque acting on the rotor shaft. Clearly, there is a theoretical limit to the amount of power that a wind turbine is capable of producing, the so-called Betz's limit, $C_p = 0.59$ (*Hansen* (2008)). Although such a boundary sets a limit on how much energy wind turbines can extract from the wind, the actual efficiency is lower than Betz's limit. Wind turbines are typically installed in places characterized by highly unstable environment conditions. As consequence, the aerodynamic loads acting on the blades and the tower are unsteady in nature, showing a very complex temporal and spatial distribution. For all these reasons, an accurate and reliable prediction of the aerodynamic/aeroelastic behavior of large horizontal-axis wind-turbines is currently a challenge in computational mechanics (*Leishman* (2002)).

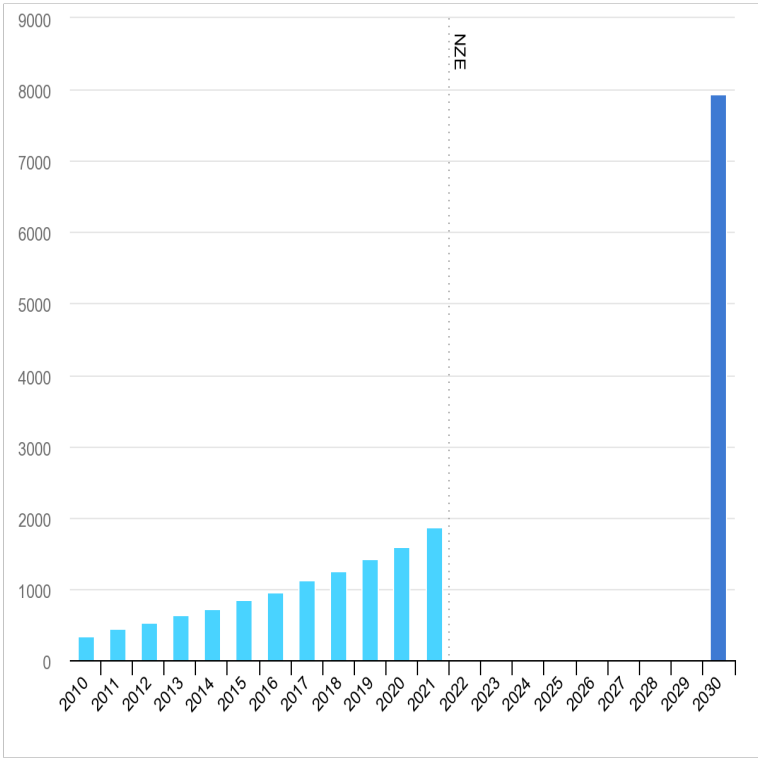

Figure 2: Figure showing wind power generation in the net zero scenario 2010 - 2030, from IEA

0.2.3 Aeroelastic flutter

A flexible body immersed in a fluid flow can experience oscillations as a consequence of interactions between inertial, elastic, dissipative, control, and aerodynamic forces (*Hodges and Pierce* (2011)). Among the different aeroelastic phenomena, flutter has been extensively studied and pinpointed as the cause of many catastrophic civil and aeronautical events. A historical incident related to this aeroelastic mechanism is associated with the collapse of the famous Tacoma Narrows bridge in 1940 (*Billah and Scanlan* (1990)).

Although fluttering is typically classified as a destructive phenomenon, recent studies have shown that it is possible to take advantage of it as a means for electric energy generation. By using piezoelectric or electromagnetic devices, one can convert self-sustained mechanical vibrations into usable energy. Piezoelectric materials can accumulate electric charge as consequence of mechanical deformations (through the direct piezoelectric effect). Erturk et. al. (*Erturk et al.* (2010)) considered a single piezoelectric harvester to develop an experimental and analytical model for predicting the electrical power produced at flutter speed. Moreover, Erturk's team used a thin, symmetrical, rigid, and a large aspect-ratio wing linked, through torsional springs, to a supporting piezoelectric structure. Although the power output of such

systems is much lower when compared to traditional horizontal wind turbines, these devices are intended for low power generation to be located in hostile or inaccessible places, such as the Antarctica.

In the field of energy harvesting, a large number of analytical, numerical and experimental investigations were carried out to understand the basic physics involved in flutter-based piezoelectric harvesters (*Bryant and Garcia* (2009),*Dias et al.* (2015), *Erturk et al.* (2010)). Although those models are mostly linear, they allowed us to get important insights regarding the aeroelastic behaviour, energy generation and how the flutter boundary changes as a function of system parameters. Despite the relevance of these works, further studies are needed to fully understand the role of nonlinearities in harvester power production. The nonlinear response of a flutter-based aeroelastic system generally involves a combination of very different nonlinearities, such as: geometric, material, inertial, electrical, and those from the flowfield (*Roccia et al.* (2020)).

Nonlinear analysis is often more complex and computationally more expensive than linear approaches (*Roccia et al.* (2020)), depending on the number of nonlinearities present in the system. A good prediction of the nonlinear response is important in order to improve the prediction of the system's output. Along this path, artificial neural networks (ANNs) have been gaining ground for surrogate modeling based on a data-driven machine learning approach, for instance: computer vision (*Krizhevsky et al.* (2017)), natural language processing (*Kumar et al.* (2016)) and for identifying nonlinear systems of equations (*Genc* (2017)).

0.2.4 Aerodynamic model

A number of different aerodynamic models can be used for computing the loads from the surrounding flow acting on the structure. Among the large number of methods, beam element moment methods (BEMMs), boundary element methods (BEMs), and computational fluid dynamics (CFD) techniques are the most popular depending on the problem to be solved. BEMM-based solvers are computationally efficient, but lack accuracy when compared to CFD methods. Moreover, CFD solvers are the right choice from a precision point of view, but their computational cost is extremely high, often prohibitive for studying fluid-structure interaction problems. Because of this, BEM approaches, such as the well-known unsteady vortex-lattice method (UVLM), are frequently used in the context of computational aeroelasticity; actually, it presents an excellent trade-off between precision and computational cost (*Lee and Lee* (2019)). Different versions of the UVLM have been used to investigate flapping-wing kinematics (*Roccia et al.* (2013)), rotorcraft (*Rados et al.* (2001)), and wind-turbines (*Gebhardt et al.* (2010); *Strganac and Mook* (1990)). Many works about the aeroelastic behavior of planes were carried out by researchers at Virginia Polytechnic Institute and State University. They extended the UVLM capabilities and coupled it with several structural models in order to investigate, mostly, different aircrafts configurations (*de Souza et al.* (2012); *Wang et al.* (2010)). More recently, Gebhardt and Roccia used a modified version of the unsteady vortex-lattice method coupled with a geometrically non-linear structural model to study the aeroelastic behavior of a large horizontal-axis wind-turbine (*Gebhardt and Roccia* (2014)).

Two-dimensional versions of the UVLM were also used to study different aerodynamic and aeroelastic problems. In particular, Roccia et al. used a modified version of a 2D UVLM, hereafter called unsteady vortex-particle method (UVPM), for predicting the aeroelastic response of piezoelectric energy harvesting devices (*Roccia et al.* (2020)).

0.2.5 Structural model

Although several structural models are possible, such as those based on lumped- or distributed-parameter models, researchers often select reduced-order models to gain insight into the underlying physics of a given problem. On this basis, *Roccia et al.* (2020) considered a 3 degrees-of-freedom (DOFs) lumped-parameter model for studying energy harvesting devices. They used the four-order predictor-corrector method of Hamming to numerically integrate all the governing equations in the time domain. In this thesis, a 2-DOFs nonlinear structural model was considered to predict the response of the system. In addition, several numerical schemes were implemented to integrate the set of ordinary differential equations (ODEs) describing the temporal evolution of the aeroelastic system.

0.3 Objectives

The main objective of this thesis is to develop a nonlinear two-dimensional aeroelastic model well-suited to investigate the fluid-structure interaction of a typical airfoil. Such a main goal will be achieved by considering the following specific objectives:

- Implement a modified version of the unsteady vortex-particle method by including several numerical schemes to propagate the wakes.
- Develop a two-dimensional (plunge and pitching) nonlinear structural model to predict the airfoil response.
- Implement an inter-connection model to exchange information between the aerodynamic and structural models.
- Implement an ANNs to replace the constitutive law associated with the nonlinear springs.
- verification of the aerodynamic, structural, and aeroelastic numerical implementations by considering classic results available in the literature.
- Use the Hausdorff distance as an additional comparison criteria to measure to what extent two wakes are in agreement or not.
- Assess the performance of the designed ANN in the context of computational aeroelasticity.
- Study flutter for the proposed aeroelastic problem.

0.4 Thesis outline

The rest of the thesis will be structured as follows. Chapter 1 will cover the necessary background theory. Then, in chapter 2, the discrete numerical implementation of the UVPM will be presented, where all the steps of the method are carefully explained. In chapter 3 the structural model of the airfoil is presented and the equations of motion are derived. Then, in chapter 4, the aeroelastic model for the system is described and the coupling simulation scheme between the aerodynamic and structural models is explained in detail. The numerical results are presented in chapter 5, which is divided into two parts. The first part contains the verification of the different models implemented, and the second part contains the new cases of study. Finally, in chapter 6, concluding remarks are presented.

Chapter 1

General aerodynamics

1.1 The governing equations

In this section, the governing equations for the two dimensional continuous model are presented. For a more in depth explanation of the theory and the full derivation of the equations, see *Kundu and Cohen* (2003) and *Katz and Plotkin* (2001).

1.1.1 Navier - Stokes equations

The fundamental equations governing a flow field are given by the Navier - Stokes equations. For two dimensional flows, the velocity vector at a given position and time is defined as $\mathbf{u} = \mathbf{u}(\mathbf{r},t)$, where $\mathbf{r} = \mathbf{r}(x,y) \in \mathbb{R}^2$. Here, $\mathbf{r}(x,y) = x\mathbf{e}_x + y\mathbf{e}_y$, where $\mathbf{e}_x, \mathbf{e}_y$ are Cartesian basis vectors and $x \in \mathbb{R}$ and $y \in \mathbb{R}$ are scalar coefficients corresponding to the position in the two dimensional xy plane. In the most general form, the Navier-Stokes equations can be written as

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = \rho\mathbf{g} + \nabla\cdot\mathbf{T}. \tag{1.1}$$

In equation (1.1) above, ρ is the mass density $[\frac{kg}{m^2}]$, $\frac{\partial \mathbf{u}}{\partial t}$ is the time derivative of the velocity $[\frac{m}{s^2}]$, $\mathbf{u}\cdot\nabla\mathbf{u}$ is the convective acceleration term of the flow $[\frac{m}{s^2}]$, $\mathbf{g} = [0,-g]$ is the two dimensional gravitational acceleration vector, where $g = 9.81m/s^2$, $\nabla = \frac{\partial}{\partial x}\mathbf{e}_x + \frac{\partial}{\partial y}\mathbf{e}_y$ is the two dimensional gradient operator and $\nabla\cdot\mathbf{T}$ is the divergence of the stress tensor $\mathbf{T}$ $[\frac{N}{m^2}]$. For a Newtonian flow with constant viscosity, the divergence of the stress tensor can be written as $\nabla\cdot\mathbf{T} = -\nabla p + \mu\nabla^2\mathbf{u} + \frac{\mu}{3}\nabla(\nabla\cdot\mathbf{u})$ (*Kundu and Cohen* (2003)), where p is the pressure $[\frac{N}{m}]$, μ is the dynamic viscosity $[\frac{kg}{ms}]$

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = \rho\mathbf{g} - \nabla p + \mu\nabla^2\mathbf{u} + \frac{\mu}{3}\nabla(\nabla\cdot\mathbf{u}). \tag{1.2}$$

The Navier-Stokes equations in equation (1.2) is written in vector form, and is a statement of the conservation of momentum of the flow.

1.1.2 Continuity equation

The total mass of a fixed control area can be computed by integrating the differential mass $dm = \rho dA$, over the entire fixed control area. A fixed control area is defined as a stationary area, in which the total mass is conserved at all times. In figure 1.1, the total mass of the area enclosed by the boundary curve S.

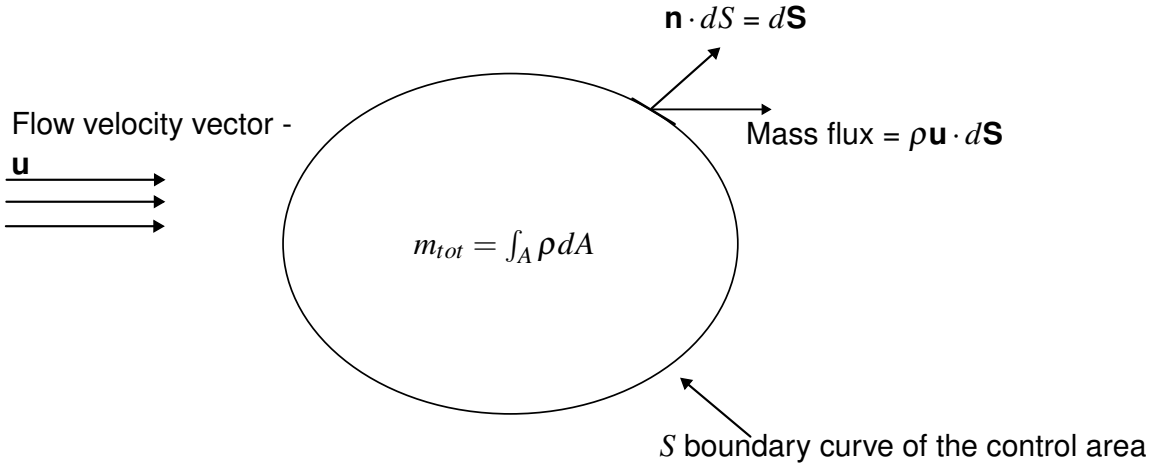

Figure 1.1: Control area for a flow

$$m_{tot} = \int_A \rho dA. \tag{1.3}$$

The transport of mass due to the flow velocity $\mathbf{u}$ is given as $\rho\mathbf{u} \cdot d\mathbf{S}$, which is the mass flux through the boundary S, illustrated in figure 1.1. $d\mathbf{S}$ is the normal surface vector on the surface of the boundary S. Since mass must be conserved, the time derivative of the total mass of the fixed control area be equal to zero, which leads to

$$\frac{dm_A}{dt} = \frac{\partial}{\partial t}\int_A \rho dA + \int_A \rho(\mathbf{u}\cdot\mathbf{n})dS = 0. \tag{1.4}$$

The second integral $\int_A \rho(\mathbf{u}\cdot\mathbf{n})dS$ in equation (1.4) is the line integral of the mass flux of the entire boundary surface, where $\mathbf{n}$ is the normal vector to the surface of the control area. It computes the total mass flux leaving and entering the boundary surface of the area. From the divergence theorem, the expression for the line integral can be transformed to the area integral (*Katz and Plotkin* (2001) pg.8)

$$\frac{dm_A}{dt} = \frac{\partial}{\partial t}\int_A \rho dA + \int_A \nabla\cdot(\mathbf{u}\rho)dA = \int_A (\frac{\partial \rho}{\partial t} + \nabla\cdot(\mathbf{u}\rho))dA = 0. \tag{1.5}$$

The time derivative can be taken inside the integral since the control area is assumed to be stationary. In order for the integral to be valid for an arbitrary control area, the sum of the terms inside the integral in equation (1.5) must be zero

$$\frac{\partial \rho}{\partial t} + \nabla\cdot(\mathbf{u}\rho) = 0. \tag{1.6}$$

This equation is the continuity equation for the flow, which states the conservation of mass in differential form.

1.1.3 Potential flow conditions

By making a few assumptions of the flow, the governing equations for the flow can be simplified to a great extent. From the Helmholtz decomposition theorem, the velocity vector of the flow can be decomposed into a rotational part $\nabla\times\Psi$ and an irrotational part $\nabla\phi$ (*Joseph* (2006)), such that

$$\mathbf{u} = \nabla\times\Psi + \nabla\phi. \tag{1.7}$$

In the equation above, $\Psi = \Psi(x,y)$ and $\phi = \phi(x,y)$ are scalar functions in the two dimensional flow field.

Incompressible flow condition

An incompressible flow is defined as a flow where the mass density is constant (*Kundu and Cohen* (2003) pg.81). By imposing this condition on the flow, the expression for the continuity equation can be simplified to

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\mathbf{u}\rho) = 0 + (\nabla \cdot \mathbf{u})\rho + (\nabla \cdot \rho)\mathbf{u} = (\nabla \cdot \mathbf{u})\rho = 0. \tag{1.8}$$

Since ρ is nonzero and constant, dividing both sides by ρ gives the following expression

$$\nabla \cdot \mathbf{u} = 0. \tag{1.9}$$

Inviscid flow condition

Another simplification of the governing equations of motion is to assume the flow to be inviscid. An inviscid flow is a flow where the viscous effects from the viscous forces are not present (*Kundu and Cohen* (2003) pg.149). Thus, the dynamic viscosity constant must be zero for an inviscid flow, $\mu = 0$. The viscosity of the flow is a measure of the flows resistance to move. Such an inviscid flow will have a nonzero tangential velocity at a solid surface.

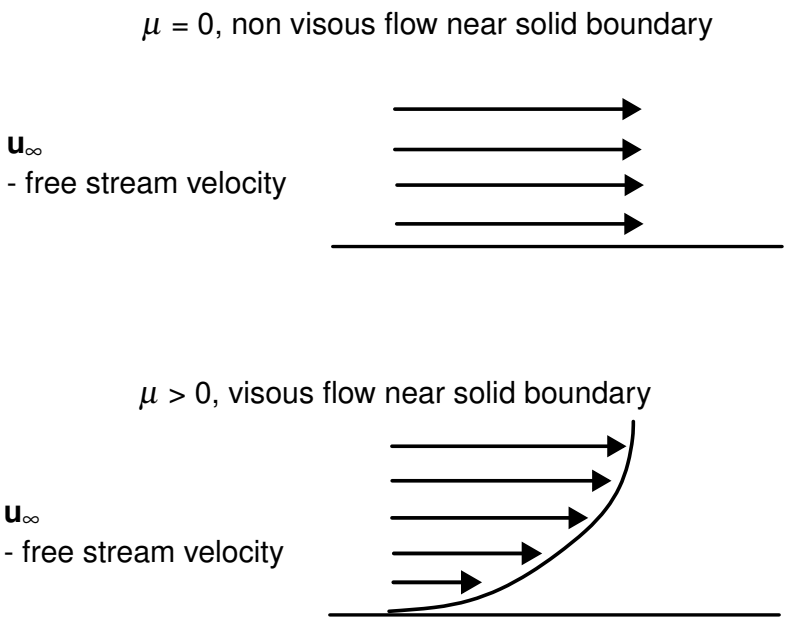

Figure 1.2: Illustration of invscid vs non inviscid flow

In figure 1.2, the difference between an insvicid flow and non inviscid flow is illustrated. As one can see, near the solid boundary the insvicid flow maintains the same horizontal velocity, contrary to the non inviscid flow, where the flow slows down near the solid surface due to viscous effects.

Irrotational flow

For a flow with velocity vector $\mathbf{u}$, the vorticity vector is defined as the curl of the velocity vector

$$\omega = \nabla \times \mathbf{u}. \tag{1.10}$$

For an irrotational flow, this quantity is zero everywhere in the flow domain: $\omega = 0$. Thus, if a flow satisfies both the irrotational, inviscid and the incompressible condition, the flow can be defined as a potential flow, where the velocity vector can be defined as $\mathbf{u} = \nabla\phi$.

1.1.4 Laplace's equation

From the irrotational flow condition, the rotational part of the decomposed velocity vector has to be zero, $\nabla \times \Psi = 0$ and the velocity vector becomes $\mathbf{u} = \nabla\phi$. By replacing this expression for the velocity in the incompressible continuity equation of the flow, leads to the following expression:

$$\nabla \cdot \mathbf{u} = \nabla \cdot \nabla\mathbf{u} = \nabla^2\phi = 0. \tag{1.11}$$

This is known as the Laplace's equation of the flow, which is a linear second order differential equation.

1.1.5 Incompressible Euler's equations

From the incompressible flow condition, $\nabla \cdot \mathbf{u} = 0$. Based on this condition, the Navier-Stokes equations can be simplified to

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla\mathbf{u} = \mathbf{g} - \frac{\nabla p}{\rho} + \mu\nabla^2\mathbf{u}. \tag{1.12}$$

If the insvicid condition for the flow is met as well, then $\mu = 0$, which leads to the viscous term $\mu\nabla^2\mathbf{u} = 0$ and the final equations for the flow becomes the incompressible Euler's equations

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla\mathbf{u} = \mathbf{g} - \frac{\nabla p}{\rho}. \tag{1.13}$$

As one can see in equation (1.13), the viscous terms disappear in the Euler's equation and the only force terms left are the pressure gradient and the gravitational body force.

1.1.6 Boundary conditions

By solving the Laplace's equation, one can obtain the expression for the velocity potential for the velocity field. However, there exists an infinite number of different solutions to the problem $\nabla^2\phi = 0$. In order to obtain a unique solution for the problem, a set of boundary conditions need to be specified first.

Non-penetration condition

The flow velocity component normal to the boundary surface needs to be equal to the normal velocity of the boundary surface itself. This is to ensure the fluid doesn't penetrate the solid boundary surface, which is physically not possible. If the boundary surface is stationary, the normal component of the flow must be zero here

$$\mathbf{u} \cdot \mathbf{n} = \nabla\phi \cdot \mathbf{n} = \mathbf{u}_{boundary} \cdot \mathbf{n} = 0. \tag{1.14}$$

Here, $\mathbf{u} \cdot \mathbf{n}$ is the normal velocity of the flow, which must be equal to the normal velocity of the boundary wall $\mathbf{u}_{boundary} \cdot \mathbf{n}$ at this location.

Boundary condition at infinity

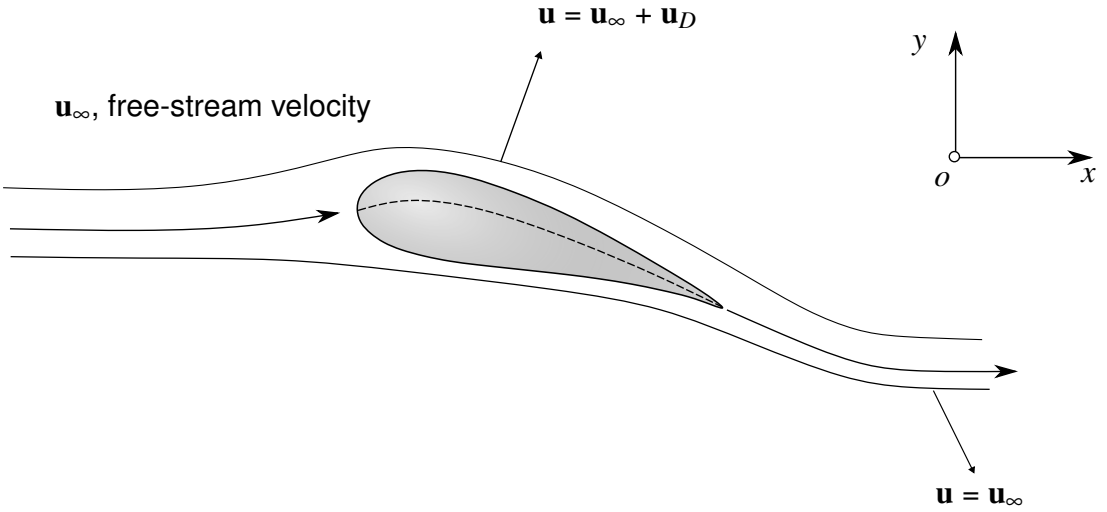

Figure 1.3: Boundary conditions at infinity.

For an the airfoil immersed in a flow with uniform horizontal velocity in the x direction, the velocity vector near the airfoil is given by: $\mathbf{u} = \nabla\phi_D + \nabla\phi_\infty$. Here, $\nabla\phi_\infty = \mathbf{u}_\infty$ is the velocity coming from the free stream, while $\nabla\phi_D = \mathbf{u}_D$ is the velocity coming from the disturbance due to the airfoil. As the free stream flow leaves the trailing edge of the airfoil, the velocity gets restored to its initial velocity again $\mathbf{u} = \mathbf{u}_\infty$. This condition is know as the boundary condition at infinity and is given by

$$\lim_{r\to\infty} \nabla\phi_D = \lim_{r\to\infty} \mathbf{u}_D = 0. \tag{1.15}$$

In equation (1.15), the velocity disturbance coming from the airfoil $\mathbf{u}_D$ decreases as the distance r goes to infinity. The r value in the subscript of the limit is the Euclidean distance from a point on the airfoil and to an arbitrary point in the flow domain. Thus, as the distance from the airfoil grows, the free stream velocity turns back to its original state.

1.1.7 Vortex point solution

One of the fundamental elementary solutions to the Laplace's equation is the two dimensional vortex point. The vortex point is a singularity element, which means the Laplace's equation is only satisfied for the flow outside the core of the vortex point. The full derivation of the the velocity potential for vortex point located at position (x_0, y_0) can be found in *Katz and Plotkin* (2001), pg.70, and is given by

$$\phi(x,y) = -\frac{\Gamma}{2\pi}\arctan(\frac{y-y_0}{x-x_0}). \tag{1.16}$$

In equation (1.16), (x_0, y_0) is the center coordinate of the vortex point, given in Cartesian coordinates. Γ is the circulation strength of the vortex point, which is defined as $\Gamma = \int_C \mathbf{u}\cdot d\mathbf{l}$, where $\mathbf{u}$ is the velocity vector of the flow and $d\mathbf{l}$ is an infinitesimal length element of the curve c. Thus, the circulation $\Gamma\ [\frac{m^2}{s}]$ of the flow is the closed line integral of the flow field around the curve c, enclosing the vortex point. (x,y) is an arbitrary point in the flow. The velocity components can be found by taking the gradient of the velocity potential, where $\mathbf{u} = \nabla\phi(x,y)$, and the expression for the velocity components in polar form is given as $\mathbf{u}_r = 0$ and $\mathbf{u}_\theta = -\frac{\Gamma}{2\pi r}$. $r = \sqrt{(x-x_0)^2 + (y-y_0)^2}$ is the Euclidean distance between vortex point and point (x,y) in the

flow. Based on the velocity components, the induced velocity from the vortex point decreases as the distance r from the vortex point increases.

1.1.8 Biot - Savart law

By using the Biot - Savart law, one can obtain the velocity field for any points in the two dimensional flow domain. For a full derivation for the formula, see (*Katz and Plotkin* (2001) pg.44). The expression for the velocity vector at a position $\mathbf{r}$ and at time t is given as

$$\mathbf{u}(\mathbf{r},t) = \frac{1}{2\pi}\int_{A(\mathbf{r},t)} \frac{\omega(\mathbf{r},t)\times(\mathbf{r}-\mathbf{r}_1)}{r^2} dA(\mathbf{r},t) + \mathbf{u}_\infty. \tag{1.17}$$

In the equation above, $\mathbf{u}_\infty$ is the velocity contribution from the free stream velocity of the flow, which is assumed to be constant everywhere in the flow domain. The integral term is the induced velocity from a region of with vorticity in the flow. The value $\omega(\mathbf{r},t)$ is the vorticity vector at $\mathbf{r}$, $A(\mathbf{r},t)$ is the flow domain region, $\mathbf{r}_1$ is the point where the induced velocity is being computed, $r = ||\mathbf{r}-\mathbf{r}_1||$ is the Euclidean distance between the two points $\mathbf{r}$ and $\mathbf{r}_1$. In the flow domain where the flow is irrotational, the integrand term in equation (1.17) is zero, since by definition $\omega(\mathbf{r},t) = 0$ when the flow is irrotational.

1.1.9 Kelvins theorem

For a closed curve surrounding a body immersed in an potential flow, the expression of the total derivative of the circulation for this curve is given by

$$\frac{D\Gamma}{Dt} = \frac{D}{Dt}\int_C \mathbf{u}\cdot d\mathbf{l} = \int_C \frac{D\mathbf{u}}{Dt}\cdot d\mathbf{l} + \int_C \mathbf{u}\cdot\frac{D}{Dt}d\mathbf{l}. \tag{1.18}$$

In the equation above, Γ is the total circulation around the closed curve C, $\mathbf{u}$ is the velocity vector and $d\mathbf{l}$ is an infinitesimal curve element. Its clear from the expression above that $\frac{D\mathbf{u}}{Dt} = \mathbf{a}$, which is the acceleration term and that $\frac{D}{Dt}d\mathbf{l} = d\mathbf{u}$ is an infinitesimal velocity element. Based on this, the equation becomes

$$\frac{D\Gamma}{Dt} = \int_C \mathbf{a}\cdot d\mathbf{l} + \int_C \mathbf{u}\cdot d\mathbf{u} = \int_C \mathbf{a}\cdot d\mathbf{l}. \tag{1.19}$$

The second term $\int_C \mathbf{u}\cdot d\mathbf{u} = \int_C d(\frac{\mathbf{u}\cdot\mathbf{u}}{2}) = \int_C d(\frac{u^2}{2})$ in the equation is zero since the closed integral of an exact differential which is a function of the coordinates and time only must be zero (*Katz and Plotkin* (2001) pg.31). Here, $\mathbf{u}\cdot\mathbf{u} = u^2$ is the dot product of the velocity vector with itself, where $u = |\mathbf{u}|$ is the absolute value of the velocity vector. From the incompressible Euler's equation, the expression for the acceleration can be replaced by

$$\frac{D\Gamma}{Dt} = -\int_C \nabla(\frac{p}{\rho})\cdot d\mathbf{l} + \int_C \mathbf{g}\cdot d\mathbf{l} = -\int_C d(\frac{p}{\rho}) + \int_C \mathbf{g}\cdot d\mathbf{l} = 0. \tag{1.20}$$

The expression for $\frac{D\Gamma}{Dt}$ is zero since $\int_C d\frac{p}{\rho}$ is just $\frac{p}{\rho}$ evaluated at the end points of the curve, which are the same since the curve is closed and thus must be zero.The second term is zero sine the force is gravity and the work done by any conservative force around a closed curve must be zero.

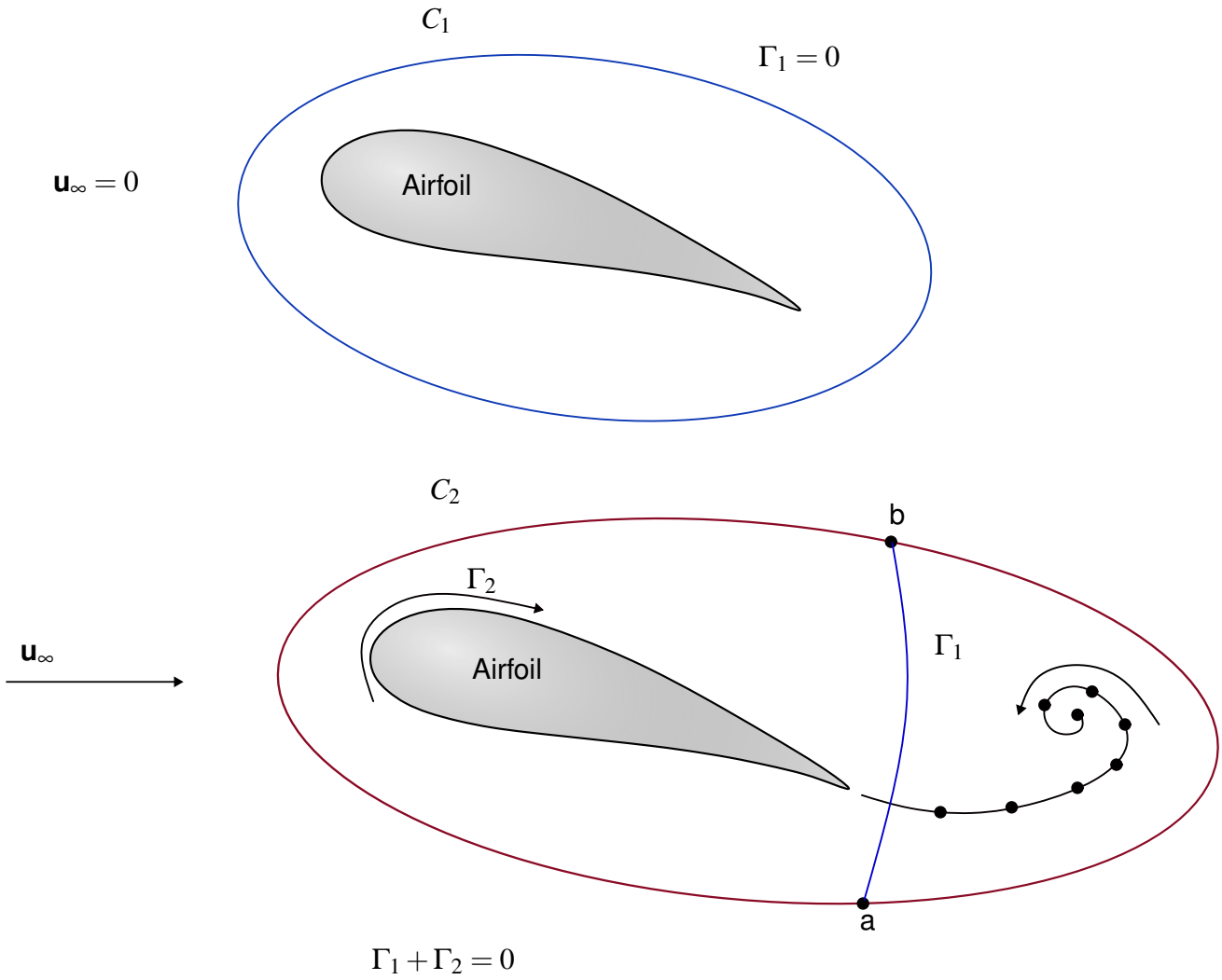

Figure 1.4: Development of circulation for closed curve around airfoil.

As one can see from figure 1.4, the total circulation around the airfoil enclosed by curve C_1 is zero at the beginning, since the free stream velocity $\mathbf{u}_\infty = 0$ and thus $\Gamma_1 = \int_{C_1} \mathbf{u} \cdot d\mathbf{l} = 0$. In the airfoil below in figure 1.4, the circulation around the airfoil is positive, but the sum of the wake circulations cancel the positive contribution and thus total circulation for the curve C_2 which encloses the airfoil and the wake is still 0. The line connected by the points b and d divides the circulation part around the airfoil and the circulation values of the wakes. Thus, for any curve enclosing both the wake and the airfoil, the total derivative of the circulation enclosed by this curve must satisfy

$$\frac{D\Gamma(t)}{Dt} = \frac{\Gamma_{airfoil}(t) + \Gamma_{wake}(t)}{\Delta t} = 0. \tag{1.21}$$

1.1.10 The Kutta - Joukowski theorem

From the results of Kelvins theorem, the total circulation from the airfoil and the wake must add to zero, thus the circulation around the airfoil has different sign to the circulation for the wake. If the airfoil produces lift, the circulation $\Gamma = \int_S \mathbf{u} \cdot d\mathbf{l}$ must be finite (*Anderson* (2011) pg.245), where S is a curve enclosing the airfoil. The lift of airfoil is given by the Kutta-Joukowski theorem, where

$$L = \rho u_\infty \Gamma_{airfoil}. \tag{1.22}$$

The Kutta-Joukowski theorem states that the lift of the airfoil is directly proportional to the circulation around the airfoil (*Anderson* (2011) pg.245). In the equation, L is the total lift $[\frac{kg \cdot m}{s^2}]$ of the airfoil, perpendicular to the free stream velocity u_∞ and $\Gamma_{airfoil}$ is the total circulation around the airfoil.

1.1.11 The unsteady Bernoullis equation

From the flow is assumed to be a potential flow, the velocity vector for the flow field can be written in terms of the velocity potential of the flow, where $\mathbf{u} = \nabla\phi$. Based on this, the incompressible Euler's equation can be rewritten as

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = \frac{\partial \mathbf{u}}{\partial t} + \frac{\nabla u^2}{2} - \mathbf{u} \times \omega = \mathbf{g} - \nabla \frac{P}{\rho}, \tag{1.23}$$

where $u^2 = \mathbf{u} \cdot \mathbf{u}$. Since the flow is irrotational, $\omega = 0$ and the velocity vector $\mathbf{u} = \nabla\phi$, the equation becomes

$$\nabla \frac{\partial \phi}{\partial t} + \nabla \frac{p}{\rho} + \nabla \frac{u^2}{2} - \mathbf{g} = \nabla \left(E + \frac{p}{\rho} + \frac{u^2}{2} + \frac{\partial \phi}{\partial t} \right) = 0. \tag{1.24}$$

The term $\mathbf{g}$ is written in terms of $\mathbf{g} = -\nabla E$ since its assumed to be a conservative body force, where E is the gravity potential (*Katz and Plotkin* (2001) pg.34). The equation is valid if the sum of the terms inside the brackets only depend on time, and the final expression turns into:

$$E + \frac{p}{\rho} + \frac{u^2}{2} + \frac{\partial \phi}{\partial t} = C(t), \tag{1.25}$$

where $C(t)$ is some time dependent term, since the left hand side of eq.(1.24) only depends on time. eq.(1.24) is the unsteady Bernoulli equation for the potential flow. By considering two different points in the flow, the unsteady Bernoulli equation must satisfy

$$\left(E + \frac{p}{\rho} + \frac{u^2}{2} + \frac{\partial \phi}{\partial t} \right) = \left(E + \frac{p}{\rho} + \frac{u^2}{2} + \frac{\partial \phi}{\partial t} \right)_\infty, \tag{1.26}$$

where left handside of eq.(1.24) is some arbitrary point point and the right handside is a reference point at infinity. If one assumes that $E_\infty = 0$, $\phi_\infty = const$ and $\mathbf{u}_\infty = 0$, the pressure jump or the pressure difference between the two points can be found by

$$\frac{p_\infty - p}{\rho} = \frac{\partial \phi}{\partial t} + E + \frac{u^2}{2}. \tag{1.27}$$

1.2 Thin airfoil theory

For a thin airfoil, one can approximate the flow field around this body by modelling the camber line of the airfoil as a vortex sheet. The purpose of placing a vortex sheet on the camber line is to transform the airfoil to a streamline of the flow, which satisfies the boundary conditions and the Laplace's equation.

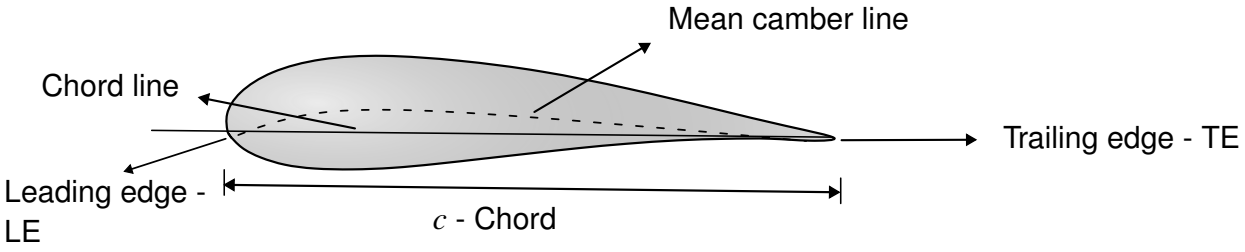

Figure 1.5: Cambered airfoil - description

Figure 1.5 shows a general cambered airfoil. The mean camber line is the line connected between the leading edge and trailing edge with equal distance to the upper and lower part of the airfoil. The chord line is the straight line connected from the leading edge to the trailing edge. For an symmetric airfoil, the mean camber line is equal to the straight chord line (see figure 1.6).

1.2.1 Continuous vortex sheet approximation

A vortex sheet is defined as a continuous segment in which a vortex density function $\gamma = \gamma(s)$ is defined over some interval.

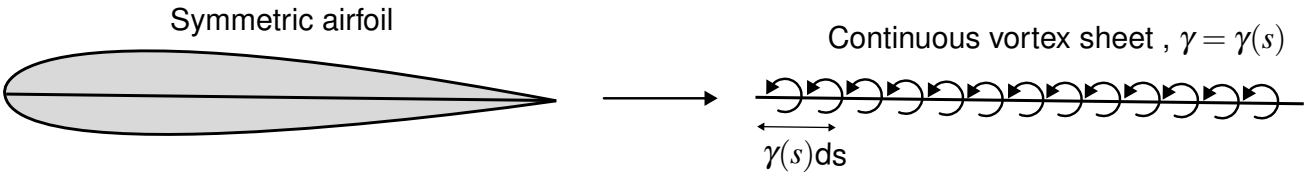

Figure 1.6: Approximation of the airfoil by a vortex sheet

Figure 1.6 shows how the thin symmetric airfoil can be modelled by approximating the camber line of the airfoil as a continuous vortex sheet.The induced velocity of a small segment of this vortex sheet can be found using Biot - Savarts law, such that

$$du(\mathbf{x}) = -\frac{\gamma(s)ds}{2\pi r}. \tag{1.28}$$

Here, r is the Euclidean distance from the infinitesimal vortex segment γds to the position vector $\mathbf{x} = (x, y)$, where the velocity is induced from the vortex segment. For the rest of this chapter, the model is assumed to be continuous and only valid for small angle approximations.

1.2.2 Non penetration boundary condition

In order for the camber line to be a streamline, the normal velocity component must be zero, in which

$$u_{\infty,n} + w(\mathbf{s}) = 0. \tag{1.29}$$

Here, $u_{\infty,n}$ is the normal component of the free stream velocity and $w(\mathbf{s})$ is the induced normal component from the vortex sheet at the position vector $\mathbf{s}$ on the vortex sheet along the chord line. The normal free stream velocity expression for an symmetric airfoil with an incoming flow at an angle of attack α is then

$$u_{\infty,n} = \mathbf{u}_\infty \cdot \mathbf{n} = u_\infty \sin\alpha. \tag{1.30}$$

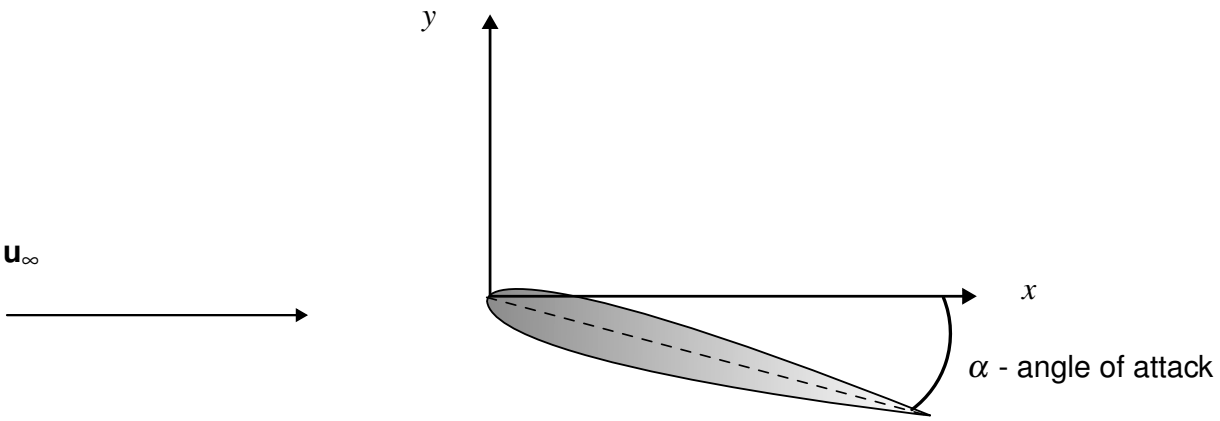

Figure 1.7: The angle of attack for an airfoil, with horizontal free stream velocity

In figure 1.7, the angle of attack is defined as the angle between the free stream velocity and the chord line of the thin symmetric airfoil. For small angles, $\sin\alpha \approx \alpha$, and equation (1.30) becomes

$$u_{\infty,n} = u_\infty \alpha. \tag{1.31}$$

In order to calculate the total velocity at position $\mathbf{x}$ induced from the vortex sheet, one can integrate $du(\mathbf{x})$ along the whole vortex sheet, which is from the leading edge to the trailing edge (*Anderson* (2011), pg.301)

$$u(\mathbf{x}) = -\int_0^c \frac{\gamma(s)ds}{2\pi r(\mathbf{x})}, \tag{1.32}$$

where $r(\mathbf{x})$ is the Euclidean distance from $\mathbf{x}$ to ds on the chord line.

1.2.3 Kuttas condition

For a given airfoil at a small angle of attack relative to the steady free stream velocity, the flow needs to leave the trailing edge of the airfoil smoothly at the top and the bottom (*Anderson* (2011) pg.291). In order for this to happen, both the upper and lower velocities at the trailing edge must cancel each other, such that this point becomes a stagnation point. Thus, the circulation strength at the trailing edge must satisfy $\gamma(TE) = 0$.

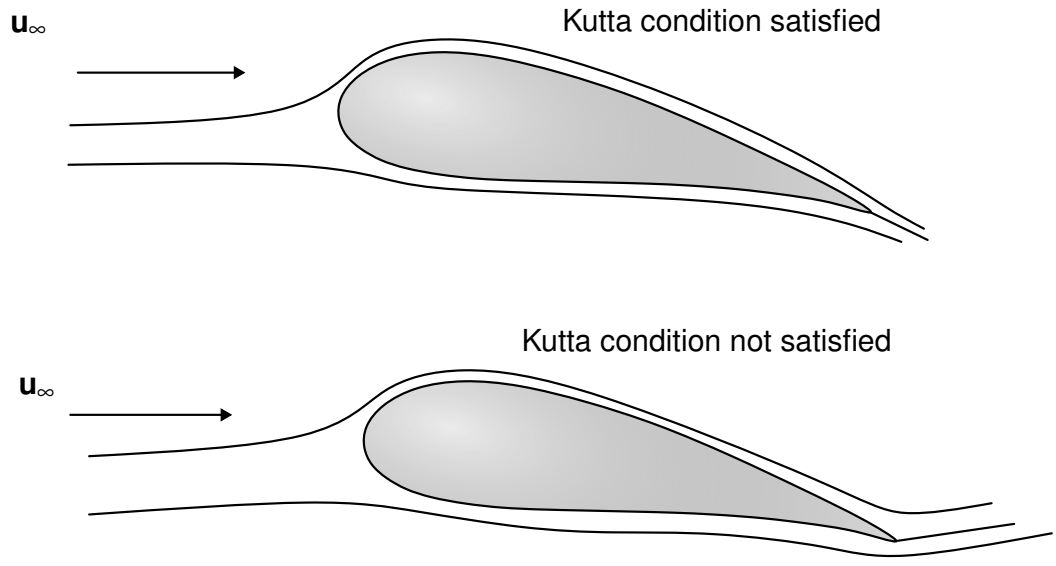

Figure 1.8: The kutta condition - illustration for an airfoil

In figure 1.8, the top airfoil satisfies the Kutta condition at the trailing edge, while the lower airfoil does not, since the flow leaves the trailing edge smoothly for the airfoil on the top.

1.2.4 Fundamental equation of thin airfoil theory

The value of $w(\mathbf{s})$ can be computed using equation (1.32), which leads

$$w(\mathbf{s}) = -\int_0^c \frac{\gamma(s)ds}{2\pi r(\mathbf{s})}. \tag{1.33}$$

Based on this new expression for $w(\mathbf{s})$, the non penetration condition for the continuous model becomes

$$-\frac{1}{2\pi}\int_0^c \frac{\gamma(s)ds}{r(\mathbf{s})} = u_\infty \alpha. \tag{1.34}$$

This expression is known as the fundamental equation of the thin airfoil theory, assuming the airfoil is symmetric. The integral can be solved by a variable transformation (*Anderson* (2011) pg.302) given by

$$s = \frac{c}{2}(1-\cos\theta), \tag{1.35}$$

and substituting for this in equation (1.31) leads to the integral

$$\frac{1}{2\pi}\int_0^\pi \frac{\gamma(\theta)\sin\theta d\theta}{\cos\theta - \cos\theta_0} = u_\infty \alpha. \tag{1.36}$$

For the limits in the integral, $\theta = 0$ corresponds to the position at the leading edge and $\theta = \pi$ corresponds to the position at the trailing edge of the airfoil. θ_0 corresponds to the given x value in terms of θ. The solution of the integral can be found in *Anderson* (2011), page 302. The explicit solution for $\gamma(\theta)$ becomes

$$\gamma(\theta) = 2\alpha u_\infty \frac{(1+\cos\theta)}{\sin\theta}. \tag{1.37}$$

1.2.5 Analytical coefficients

In this part, the analytical lift and moment coefficients are derived for the continuous model. The analytical coefficients for the continuous model will become important for verifying the discrete numerical method discussed in chapter 3.

Analytical lift coefficient

The analytical lift coefficient can be found by first finding the total circulation around the airfoil, which is done by integration of the expression for $\gamma(s)$ along the vortex sheet

$$\Gamma = \int_0^c \gamma(s)ds = \frac{c}{2}\int_0^\pi \gamma(\theta)\sin\theta d\theta = \alpha c u_\infty \int_0^\pi (1+\cos\pi)d\theta = \pi\alpha c u_\infty. \tag{1.38}$$

The expression for the circulation can be replaced in the Kutta- Joukowski equation for the lift L, which leads to the following expression

$$L = \rho_\infty u_\infty \Gamma = \pi\alpha c \rho_\infty u_\infty^2. \tag{1.39}$$

Thus, the expression for the nondimensional lift coefficient turns into

$$c_l = \frac{L}{\frac{1}{2}\rho u_\infty^2 c} = \frac{\pi\alpha c\rho u_\infty^2}{\frac{1}{2}\rho u_\infty^2 c} = 2\pi\alpha, \tag{1.40}$$

where α is the angle of attack in radians. The following result state that lift coefficient is linearly proportional to the angle of attack, for small angle approximations (*Anderson* (2011) pg.304). Its the steady state solution of the lift coefficient.

Analytical moment coefficient

The total moment about the leading edge of the vortex sheet can be found by integrating $dM = -s \cdot dL$ along the vortex sheet

$$M'_{LE} = -\int_0^c s \cdot dL = -\rho u_\infty \int_0^c s \cdot \gamma(s) ds = -\frac{1}{2} u_\infty^2 \rho c^2 \frac{\pi\alpha}{2}, \tag{1.41}$$

and the expression for the moment coefficient becomes

$$c_{m,le} = \frac{M'_{LE}}{\frac{1}{2}\rho u_\infty^2 c^2} = -\frac{\pi\alpha}{2}. \tag{1.42}$$

But from the expression of the lift coefficient, one can see that $\pi\alpha = \frac{c_l}{2}$. Thus, in terms of c_l, the moment coefficient can be rewritten as

$$c_{m,le} = -\frac{cl}{4}, \tag{1.43}$$

and the moment coefficient about the quarter chord point becomes

$$c_{m,c/4} = c_{m,le} + \frac{c_l}{4} = 0. \tag{1.44}$$

Equation (1.44) shows that the center of pressure is at the quarter chord point for the thin airfoil approximation.

Chapter 2

Aerodynamic model

In this chapter, the unsteady vortex particle method is presented. This method will be used for computing the aerodynamic loads acting on the airfoil and for convecting the wake particles behind the airfoil. The flow around the airfoil is assumed to be inviscid, irrotational and incompressible, except at the boundary surface of the body and for the wakes (*Roccia et al.* (2020)). The flow chart below shows the steps for the method.

2.1 Superposition principle

If a set of n velocity potentials $\phi_1, \phi_2, ..., \phi_n$ satisfy the Laplace's equation individually, then one can construct another velocity potential, ϕ, which is a linear combination of these other velocity potentials, where

$$\phi = \sum_{i=1}^{n} c_i \phi_i. \tag{2.1}$$

Here, the c_i are arbitrary constants and this equation must satisfy the Laplace's equation

$$\nabla^2 \phi = \sum_{i=1}^{n} c_i \nabla^2 \phi_i = 0. \tag{2.2}$$

The thin airfoil continuous vortex approximation described in chapter 2 can be simplified by replacing the continuous vortex function $\gamma(s)$ with a number of discrete vortex points on the camber line of the airfoil instead.

2.1.1 Vortex point solution

Lumped vortex element

Instead of using a continuous vortex distribution to represent the airfoil as described in chapter 1.2, one can use a set of discrete vortex point elements instead. This is done by placing the total lift $L = \rho u_\infty \Gamma$ at the center of pressure, which is located at the $\frac{c}{4}$ position. For a single vortex point placed on the airfoil surface, the non penetration boundary condition on the airfoil is given by

$$-\frac{\Gamma}{2\pi(kc - \frac{1}{4}c)} + u_\infty \alpha = -\frac{\pi c u_\infty \alpha}{2\pi(kc - \frac{1}{4}c)} + u_\infty \alpha = 0. \tag{2.3}$$

Here kc is the unknown position on the airfoil where the induced velocity is located. Solving for k in equation (2.3) leads to $k = \frac{3}{4}$. This $\frac{3}{4}c$ position is known as the collocation point. The Kutta condition is automatically satisfied when the boundary condition is evaluated at this

point (*Katz and Plotkin* (2001) pg.135). The vortex point solution is used as the singularity element to model both the airfoil and the free wake particles behind the airfoil.The airfoil will be represented as a set of discrete vortex particles placed along the camber line of the airfoil. The induced velocity at a collocation point *i* from a vortex point *j* is given by the formula (*Katz and Plotkin* (2001) pg.303)

$$\mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix} = \frac{\Gamma_j}{2\pi r_j^2} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} x_i - x_j \\ y_i - y_j. \end{bmatrix} \tag{2.4}$$

Here, Γ_j is the vortex strength and $r_j^2 = (x_i - x_j)^2 + (y_i - y_j)^2$. u and v are the horizontal and vertical velocity components.

Cutoff radius

When the distance between the vortex point and the point of interest goes to zero, the factor $\frac{\Gamma_j}{2\pi r_j^2}$ goes to infinity, $\lim_{r_j \to 0} \frac{\Gamma_j}{2\pi r_j^2} = \infty$.

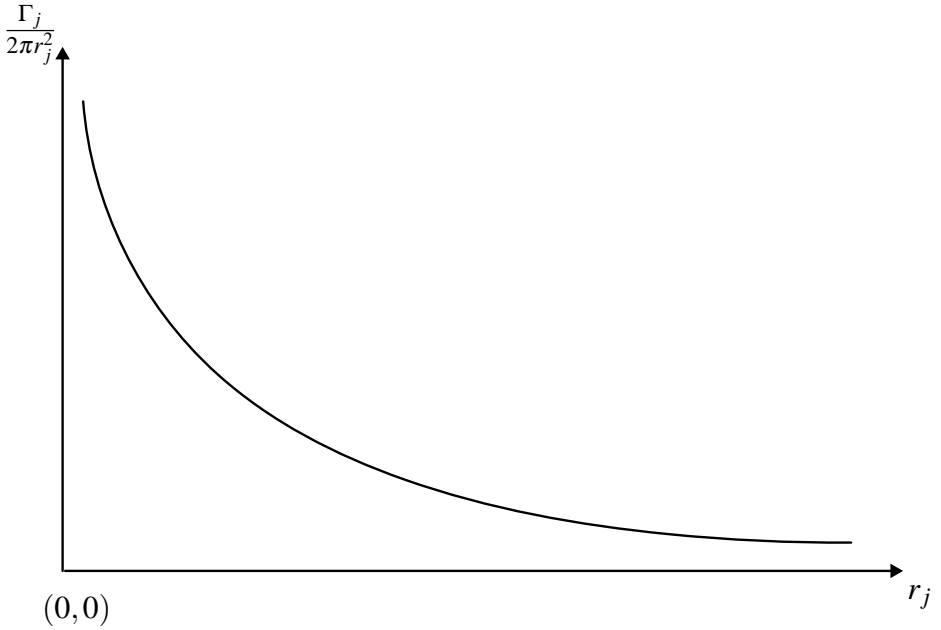

Figure 2.1: Plot of the induced velocity from a vortex singularity as a function of the distance r to the point where the velocity is induced.

In figure 2.1, the term $\frac{\Gamma_j}{2\pi r_j^2}$ for vortex singularity point is plotted as a function of the Euclidean distance r_j. As one can observe, this term goes to infinity as r_j goes to zero. Due to this problem, a cutoff radius needs to be defined such that the induced velocity does not go to infinity as the distance between the point where the velocity is induced and the position of the vortex point decreases.

$$r_j = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} >= r_{cutoff}. \tag{2.5}$$

In Algorithm 1, the induced velocity at point (x_i, y_i) is only computed when the Euclidean distance between this point and the vortex point located at (x_j, y_j) is greater than the cutoff radius. Otherwise, $\mathbf{u} = 0$. By following this approach, the large induced velocity from the vortex point is avoided as the distance gets smaller. At the boundary surface of the airfoil, this cutoff is set to zero, $r_{cutoff} = 0$. As a consequence of setting $r_{cutoff} = 0$ for the vortex points

Algorithm 1 Calculate induced vortex velocity - $VOR2D(\Gamma, x_i, y_i, x_j, y_j, r_{cutoff})$

$r_j = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$
if $r_j > r_{cutoff}$ **then**
  $u = \frac{\Gamma}{2\pi r_j^2}(y_i - y_j)$
  $v = -\frac{\Gamma}{2\pi r_j^2}(x_i - x_j)$
else
  $u = 0$
  $v = 0$
end if

at the boundary surface, the numerical discretization error will increase, depending on how many vortex points are used for modelling the airfoil surface and how close they are. When the number of vortex points on the surface goes to infinity, the model used for the airfoil should be rendered, since the numerical discretization will grow and become too large.

2.2 Airfoil panel discretization

The airfoil is discretized by dividing the chord of the airfoil into N number of panels, where each panel has an equal length of $\Delta l = \frac{c}{N}$. Here, c is the chord length $[m]$ and N is the total number of panels.

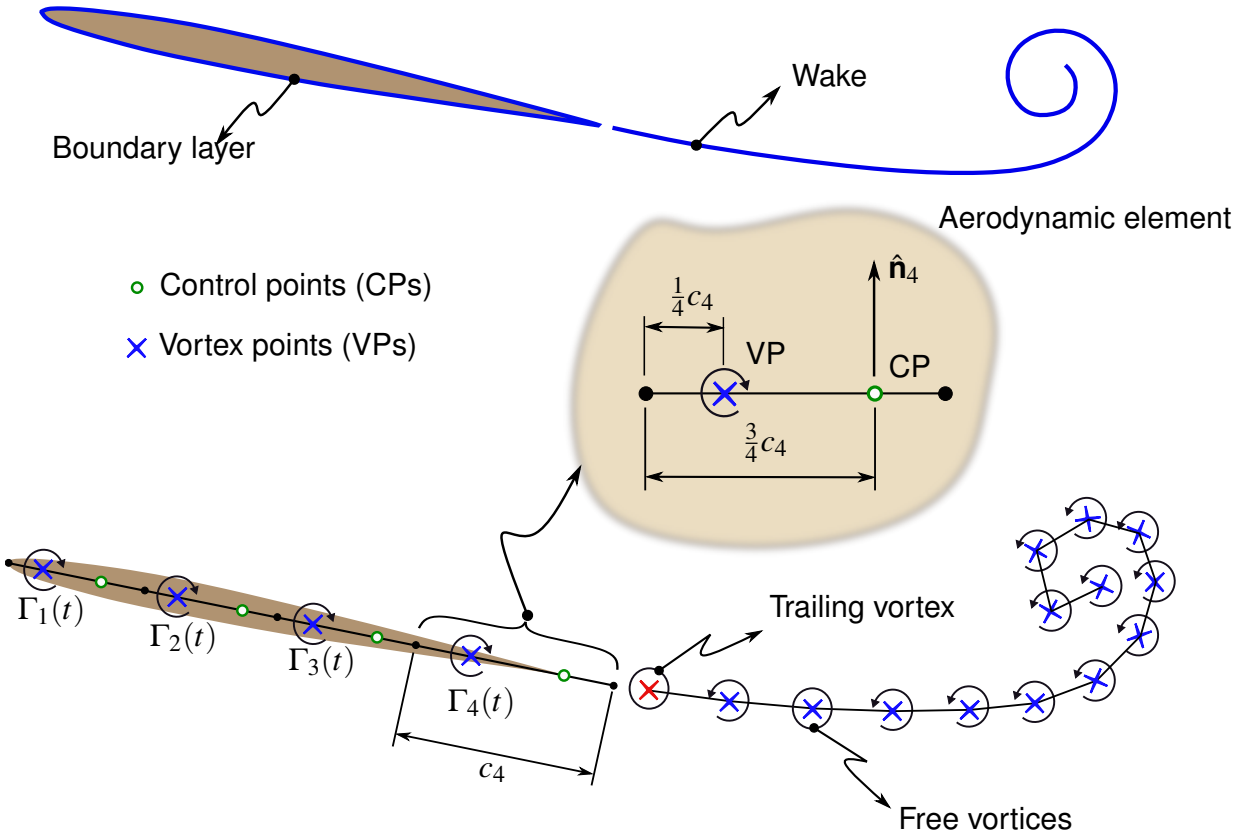

Figure 2.2: Discretization of the airfoil, using discrete vortex particles on the airfoil and the wake points.

Figure 2.2 shows how the thin airfoil and its wake is discretized using discrete vortex particles. As one can observe from figure 2.2, the singularity vortex particle elements are placed on each panel at the $\frac{1}{4}$ point of each panel along the chord line, and similarly the collocation point

is placed at the $\frac{3}{4}$ point of the panel. The collocation points are the points on the airfoil panels where the non-penetration boundary condition is computed.

Computation of vortex and collocation points

The vortex points and collocation points can be computed using a for loop and storing the values in a vector. The procedure for this is shown in Algorithm 2

Algorithm 2 Discretization of the airfoil

1: **for** $i = 1$ to $i = N$ **do**
2: $x^i_{vort} = \cos\alpha(\Delta l(i-1) + \frac{1}{4}\Delta l)$
3: $y^i_{vort} = -\sin\alpha(\Delta l(i-1) + \frac{1}{4}\Delta l)$
4: $x^i_{colloc} = \cos\alpha(\Delta l(i-1) + \frac{3}{4}\Delta l)$
5: $y^i_{colloc} = -\sin\alpha(\Delta l(i-1) + \frac{3}{4}\Delta l)$
6: **end for**

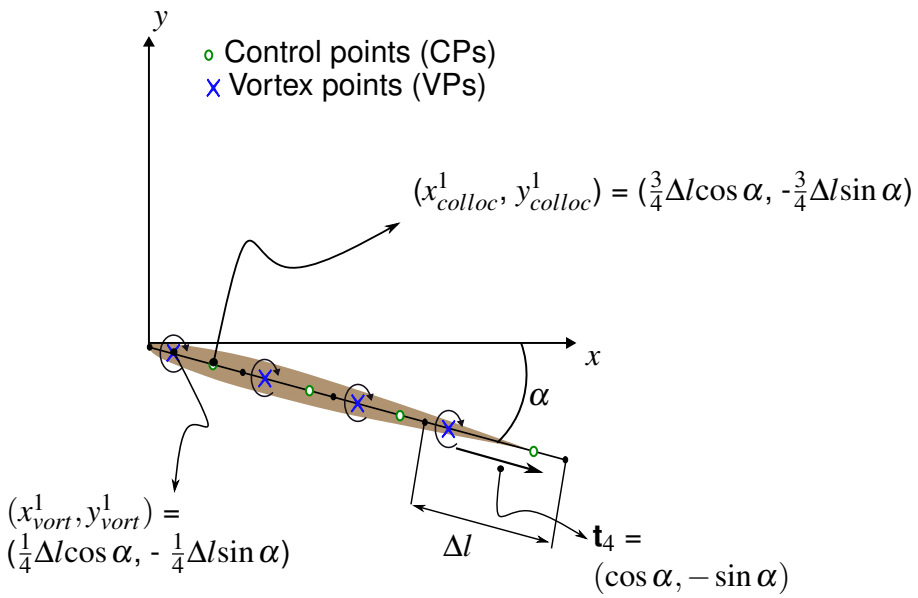

Figure 2.3: airfoil coordinates - vortex point and collocation point.

In figure 2.3, one can see the values of the corresponding collocation and vortex points for the first panel at the airfoil. In figure 2.3, the leading edge of the airfoil is assumed to be at the origin of the coordinate system. In the formula above, the latest shed wake vortex point moves a distance of

$$\Delta x_{wake} = 0.25\mathbf{u}_\infty \Delta t \tag{2.6}$$

for each time step, and leaves the trailing edge smoothly. Here, Δt is the value of the time increment $[s]$.

Normal and tangential vectors

The values of the normal and tangent vectors depends only the angle of attack of the airfoil and thus it is the same for all the panels, since the airfoil is assumed to be uncambered. In figure 2.2 and figure 2.3, the normal vector and the tangent vector for the fourth panel are displayed. The angle of attack is assumed to be positive in the clockwise direction. Thus, the formulas for the normal and tangential vectors for the airfoil becomes

$$\mathbf{n}_j = \mathbf{n} = (\sin\alpha, \cos\alpha), \tag{2.7}$$

$$\mathbf{t}_j = \mathbf{t} = (\cos\alpha, -\sin\alpha). \tag{2.8}$$

2.3 Influence matrix

The normal velocity components at each of the collocation points need to satisfy the non penetration boundary condition, which states that

$$(\mathbf{u}_W^j + \mathbf{u}_B^j + \mathbf{u}_\infty - \mathbf{u}_p) \cdot \mathbf{n} = 0. \tag{2.9}$$

In equation (2.9), $\mathbf{u}_W^j$ is the velocity at collocation point j induced by the free wake particles, $\mathbf{u}_B^j$ is the induced velocity at collocation point j from the vortex particles on the airfoil boundary surface, $\mathbf{u}_\infty$ is the free stream velocity vector and $\mathbf{u}_p^j$ is the velocity of the solid at collocation point j. By transferring all the known terms to the right hand side of equation (2.9) leads to

$$\mathbf{u}_B^j \cdot \mathbf{n} = (-\mathbf{u}_\infty - \mathbf{u}_W^j + \mathbf{u}_p) \cdot \mathbf{n}. \tag{2.10}$$

The expression for the total induced velocity due to the vortex particles placed on the boundary surface of the airfoil and the latest shed vortex point at collocation point j is given by

$$\mathbf{u}_B^j = a_{j1}\Gamma_1 + a_{j2}\Gamma_2 + ... + a_{jN}\Gamma_N + a_{jW}\Gamma_W. \tag{2.11}$$

The a_{ji} coefficients in equation (2.11) are the influence coefficients. The influence coefficient is defined as the induced normal velocity at a given collocation point i from a vortex particle j with unit circulation strength $\Gamma_j = 1$,

$$a_{ij} = (u, v)_{ij} \cdot \mathbf{n}. \tag{2.12}$$

The expression for $(u, v)_{ij}$ can be found from the formula for the induced velocity of a vortex particle with strength $\Gamma = 1$. If the airfoil is assumed to be rigid and is not going through any deformation, these values must be constant. Thus, the full no penetration boundary condition at a collocation point can be summarized as

$$a_{j1}\Gamma_1 + a_{j2}\Gamma_2 + ... + a_{jN}\Gamma_N + a_{iW}\Gamma_W = -(\mathbf{u}_\infty + \mathbf{u}_W^j - \mathbf{u}_p^j) \cdot \mathbf{n}. \tag{2.13}$$

Since theres a total of N collocation points, the total system of linear equation becomes

$$\begin{bmatrix} a_{11} & a_{12} & ... & a_{1N} & a_{1W} \\ a_{21} & a_{22} & ... & a_{2N} & a_{2W} \\ \vdots & \vdots & ... & \vdots & \\ a_{N1} & a_{N2} & ... & a_{NN} & a_{NW} \\ 1 & 1 & ... & 1 & 1 \end{bmatrix} \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \vdots \\ \Gamma_N \\ \Gamma_{W_i} \end{bmatrix} = \begin{bmatrix} RHS_1 \\ RHS_2 \\ \vdots \\ RHS_N \\ \Gamma(t - \Delta t) \end{bmatrix}. \tag{2.14}$$

The $(N+1) \times (N+1)$ matrix on the left hand side of equation (2.14) is known as the influence matrix A. The last row of matrix A comes from Kelvins condition, which states that the total circulation of the wake particles and the vortex particles on the airfoil combined must be conserved for each time step. $\Gamma(t) + \Gamma(t - \Delta t) + \Gamma_{W_i} = 0$, where $\Gamma(t) = \sum_{j=1}^{N} \Gamma_j$ is the total airfoil circulation around the airfoil at time t and Γ_{W_i} is the circulation strength of the newly shed wake vortex particle. The other components for the right hand side vector are defined as $RHS_j = -(\mathbf{u}_\infty + \mathbf{u}_W^j - \mathbf{u}_p^j) \cdot \mathbf{n}$.

Algorithm for computing A

The A matrix containing all the influence coefficients can be computed by replacing the vortex strength on each vortex point with $\Gamma = 1$. Algorithm 3 below shows how this is done, by using the subroutine $VOR2D$ defined in Algorithm 1. The a_{ij} components can be computed using two for loops, where the first loop scans each collocation points while the inner loop scans the vortex points.

Algorithm 3 Calculate influence matrix A

```
for i = 1 to i = N do
    for j = 1 to = N do
        (u,v)_ij = VOR2D(Γ = 1, x_i, y_i, x_j, y_j, 0)
        a_ij = (u,v)_ij · n_i
    end for
end for
```

Once the influence matrix is calculated, the only unknown term left is the circulation strengths vector Γ. In matrix form, the equation becomes

$$A\Gamma(t_i) = \mathbf{RHS}(t_i). \tag{2.15}$$

In the equation (2.15), A is the $(N+1) \times (N+1)$ influence matrix, Γ is a $(N+1) \times 1$ column vector with the circulation strengths for the vortex particles and **RHS** is the $(N+1) \times 1$ right hand side vector, containing all the known terms. The Γ vector solved by approximating the left hand side with the right hand side of equation (2.15), using an iterative method and not by inverting matrix A, since inverting matrix A is computationally more expensive.

2.4 Computation of the aerodynamic loads

With the values of the circulation vector $\Gamma(t_i)$, one can compute the applied jump of pressures and aerodynamic loads on each of the panels of the airfoil. The formula used to derive the expression for the pressure and loads comes from the unsteady Bernoulli equation 1.25, described in the chapter 2.

Pressure jump

From the unsteady Bernoulli equation, for two points with different pressure values at the chord line must satisfy

$$\frac{P_l}{\rho} + \frac{u_l^2}{2} + \frac{\partial \phi_l}{\partial t} = \frac{P_u}{\rho} + \frac{u_u^2}{2} + \frac{\partial \phi_u}{\partial t}. \tag{2.16}$$

The subscript l stands for lower and u for upper camber line. The gravitational potential $E = -gy$ is neglected here, since the difference in height $\Delta y \approx 0$ for the thin airfoil. The pressure difference between the upper and lower camber line then becomes

$$\Delta p = p_l - p_u = \rho[(\frac{u_t^2}{2})_u - (\frac{u_t^2}{2})_l + (\frac{\partial \phi}{\partial t})_u - (\frac{\partial \phi}{\partial t})_l]. \tag{2.17}$$

The u_t is the absolute value of the tangential velocity. The expression for the tangential velocity vector at the collocation point j on the airfoil can be found by

$$\mathbf{u}_t^j = \mathbf{u}_{tot}^j \cdot \mathbf{t}_j \pm \frac{\partial \phi}{\partial \mathbf{t}_j}. \tag{2.18}$$

The $\pm$sign stands for above and below the surface and $\mathbf{u}_{tot}^{j} = \mathbf{u}_{\infty} + \mathbf{u}_{W}^{j} + \mathbf{u}_{B}^{j} - \mathbf{u}_{p}$, is the total velocity at collocation point j. The equation for the tangential velocity at panel j is the dot product of the total velocity at the collocation point on this panel with the tangent vector. The tangential derivative of the velocity potential is approximated by

$$\pm\frac{\partial\phi}{\partial\mathbf{t}_j} \approx \pm\frac{\Gamma_j}{2\Delta l_j}. \tag{2.19}$$

The expression for the time derivative of the velocity potential is computed using the formula

$$\pm\frac{\partial\phi_j}{\partial t} \approx \pm\frac{\Delta}{\Delta t}\sum_{k=1}^{j}\frac{\Gamma_k}{2} = \pm\frac{1}{\Delta t}\sum_{k=1}^{j}\frac{\Gamma_k(t_i)-\Gamma_k(t_{i-1})}{2}. \tag{2.20}$$

In equation (2.20), the local velocity potential at panel number j is the sum of the vortices from the leading edge to the vortex point number j. Combining all the terms, the full numerical expression for computing the pressure difference between the upper and lower panel surface j of the airfoil becomes

$$\Delta p_j(t_i) = \rho\left(\mathbf{u}_{tot}^{j}\cdot\mathbf{t}_j\frac{\Gamma_j(t_i)}{\Delta c_j} + \frac{1}{\Delta t}\sum_{k=1}^{j}\frac{\Gamma_k(t_i)-\Gamma_k(t_{i-1})}{2}\right). \tag{2.21}$$

Lift force

The total lift force can be computed by approximating the integral of the pressure difference along the chord line of the airfoil, such that

$$L = F_y = \sum_{j=1}^{N}\Delta p_j\Delta c_j\cos\alpha. \tag{2.22}$$

The lift in equation (2.22) is valid when the free stream velocity $\mathbf{u}_{\infty}$ is assumed to be horizontal.

Horizontal force

The horizontal force component is computed by integrating the force which passes in the horizontal direction of the airfoil

$$F_x = \sum_{j=1}^{N}\Delta p_j\Delta c_j\sin\alpha. \tag{2.23}$$

Moment

The moment about the leading edge of the airfoil can be computed by the formula

$$M_0 = -\sum_{j=1}^{N}\Delta p_j\cos\alpha\Delta c_j x_j \tag{2.24}$$

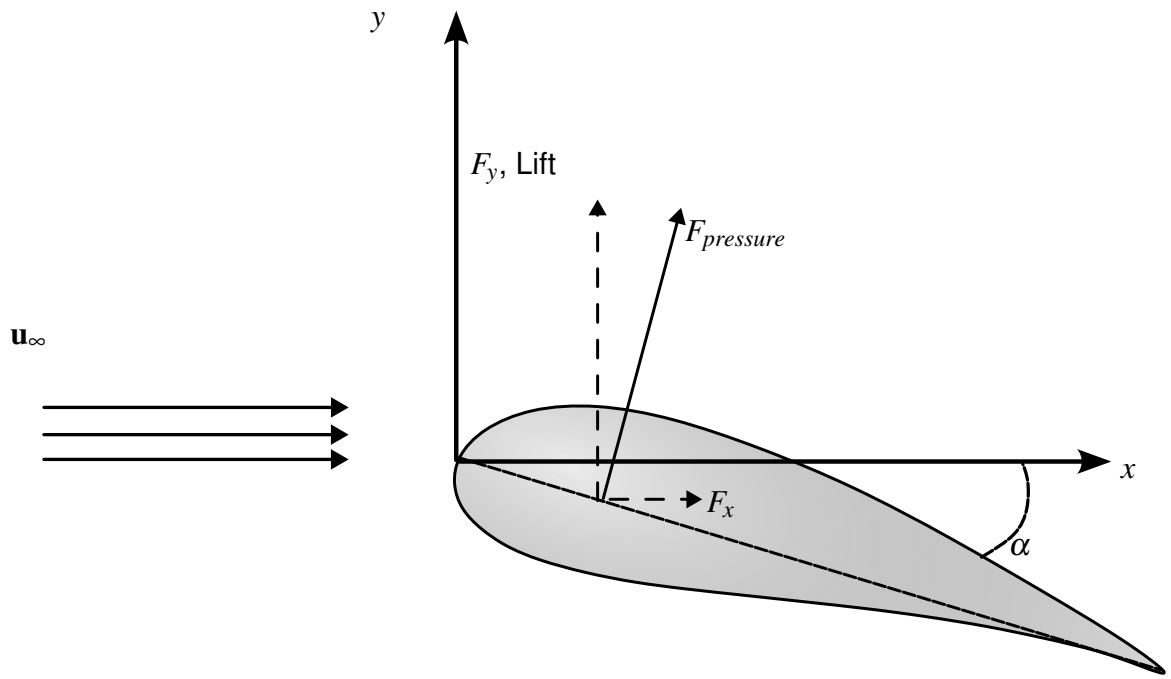

Figure 2.4: Pressure force decomposition of the airfoil.

Non dimensional lift coefficient

If the lift of the airfoil is known, then the numerical non dimensional lift coefficient can be computed as

$$C_L = \frac{L}{\frac{1}{2}\rho u_\infty^2 c}. \tag{2.25}$$

$u_\infty^2 = \mathbf{u}_\infty \cdot \mathbf{u}_\infty$ is the dot product of the free stream velocity with itself, L is the total lift of the airfoil, ρ is the air mass density and c is the length of the chord line.

Non dimensional drag coefficient

From the horizontal force, the non dimensional drag coefficient is defined as

$$C_D = \frac{F_x}{\frac{1}{2}\rho u_\infty^2 c}. \tag{2.26}$$

Jump of pressure coefficient

Similarly, the numerical non dimensional jump of pressure coefficient at panel j can be computed as

$$\Delta C_P^j = \frac{\Delta P_j}{\frac{1}{2}\rho u_\infty^2}. \tag{2.27}$$

2.5 Wake particle convection

The wake vortex particles have to move with the local stream velocity, due to the constraint that the vortex wake is force free (*Katz and Plotkin* (2001) pg.475). The local velocity for a wake vortex point is calculated as a combination of the total velocity induced by the other wake vortex particles and the vortex particles on the airfoil

$$\mathbf{u}_w^i = \sum_{j=1}^{N} \nabla\phi_j(\mathbf{x}_i) + \sum_{k=1}^{N_w} \nabla\phi_k(\mathbf{x}_i) + \nabla\phi_\infty. \tag{2.28}$$

In the equation above, $\mathbf{x}_i$ is the location of the wake vortex particle of interest, $\mathbf{u}_w^i$ is the total induced velocity at this wake vortex particle. The induced velocity is written as a superposition of the different velocity potentials, where $\sum_{j=1}^{N} \nabla\phi_j(\mathbf{x}_i)$ is the sum of the induced velocities from the bound vortex points at the boundary surface on the wake vortex point. The value of $\nabla\phi_j(\mathbf{x}_i)$ can be found using the formula for the induced velocity from a vortex point given in eq.(2.4). The second term $\sum_{k=1}^{N_w} \nabla\phi_k(\mathbf{x}_i)$ is the sum of the induced velocities from the wake vortex particles and the last term $\nabla\phi_\infty$ is the free stream velocity contribution. The vortex strengths of the wake particles Γ_k are assumed to be constant. The flow chart in figure 2.5 shows the steps for the unsteady vortex particle method.

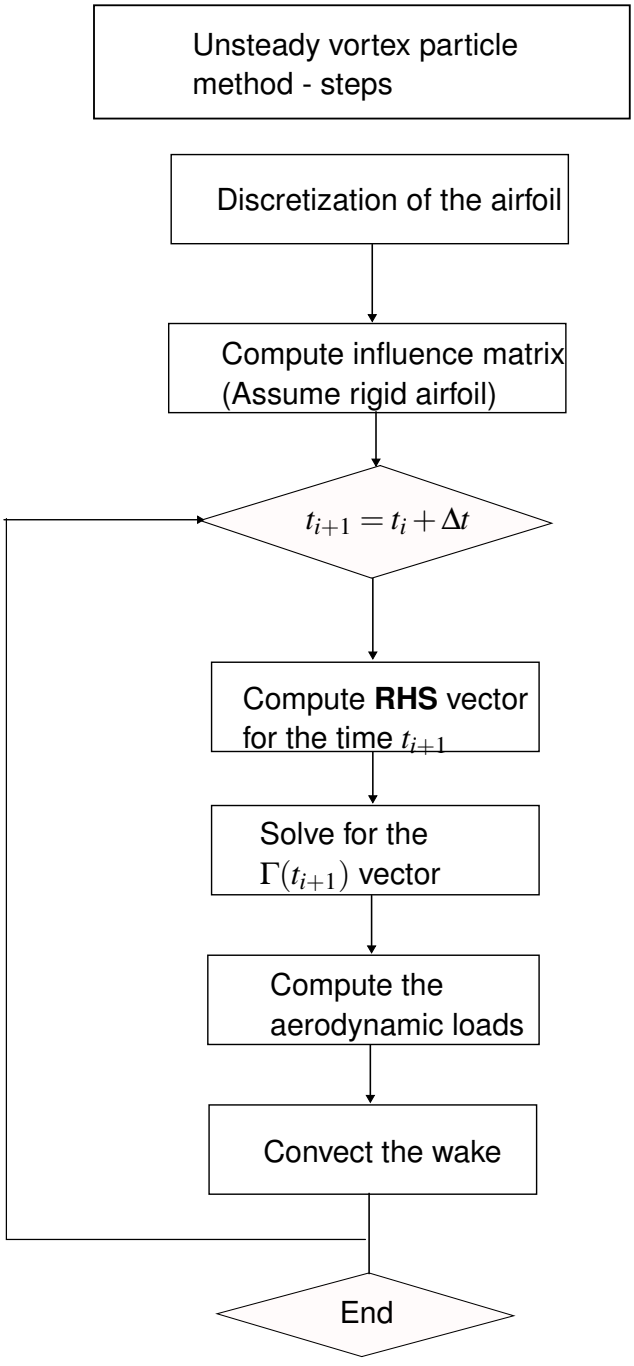

Figure 2.5: Flow chart of the unsteady vortex method.

Chapter 3

Structural model

In this chapter, the structural model of the airfoil is presented and explained. A structural model for the airfoil is necessary in order to predict the consistent response of the mechanical system when the aerodynamic loads are applied to the system. A two degrees of freedom system is considered in this case, allowing pitching and plunging motion of the rigid airfoil. The equations of motion for the structural model are derived using the Lagrange's equations. The chapter ends with introducing the concept of a neural network generated spring term, which can be used as a replacement for the analytical spring terms in the equations of motion for the airfoil.

3.1 Physical system

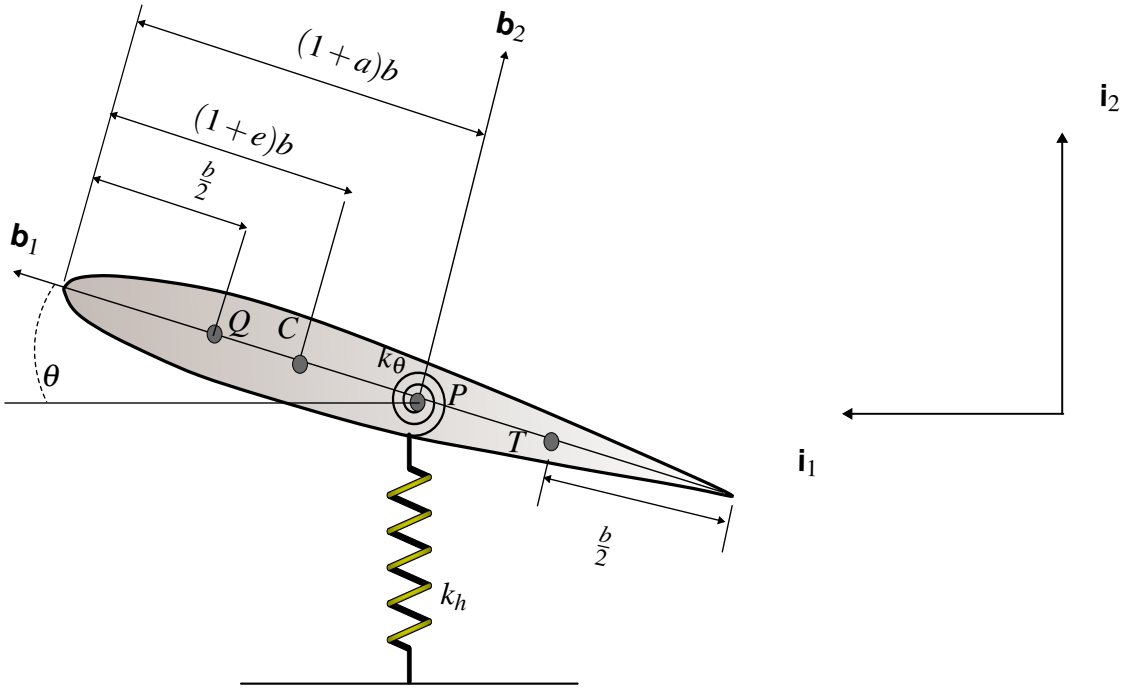

Figure 3.1: Structural model - details.

The setup for the airfoil with two degrees of freedom is illustrated in figure 3.1. Based figure 3.1 above, b is the half chord length of the airfoil, P is the reference point, C is the center of mass, Q is the aerodynamic center and T is the three quarter chord. The aerodynamic center Q is defined as the position where the aerodynamic moment remains constant, independent of

the angle of attack. For symmetric airfoils, the aerodynamic moment about the aerodynamic center is always 0 for all angles of attack (*NASA*). e and a are dimensionless numbers, where $e, a \in [-1, 1]$. k_θ and k_h are the stiffness coefficients related to the torsional spring and the vertical plunging spring. The rotation angle θ is the rotation of the body frame coordinate system with basis vectors $\mathbf{b}_1, \mathbf{b}_2$ of the airfoil, relative to the stationary inertial frame with basis vectors $\mathbf{i}_1, \mathbf{i}_2$.

3.1.1 Plunging airfoil motion

The airfoil structure can undergo plunging oscillations about some reference point. This type of motion is characterized by a motion in the vertical direction varying with time. The change of vertical position can be described using a sinusoidal function

$$h(t) = h_0 \cdot sin(\omega \cdot t). \tag{3.1}$$

In equation (3.1) ω is the angular frequency, while h_0 is the maximum amplitude. Figure 3.2 illustrates this type of motion.

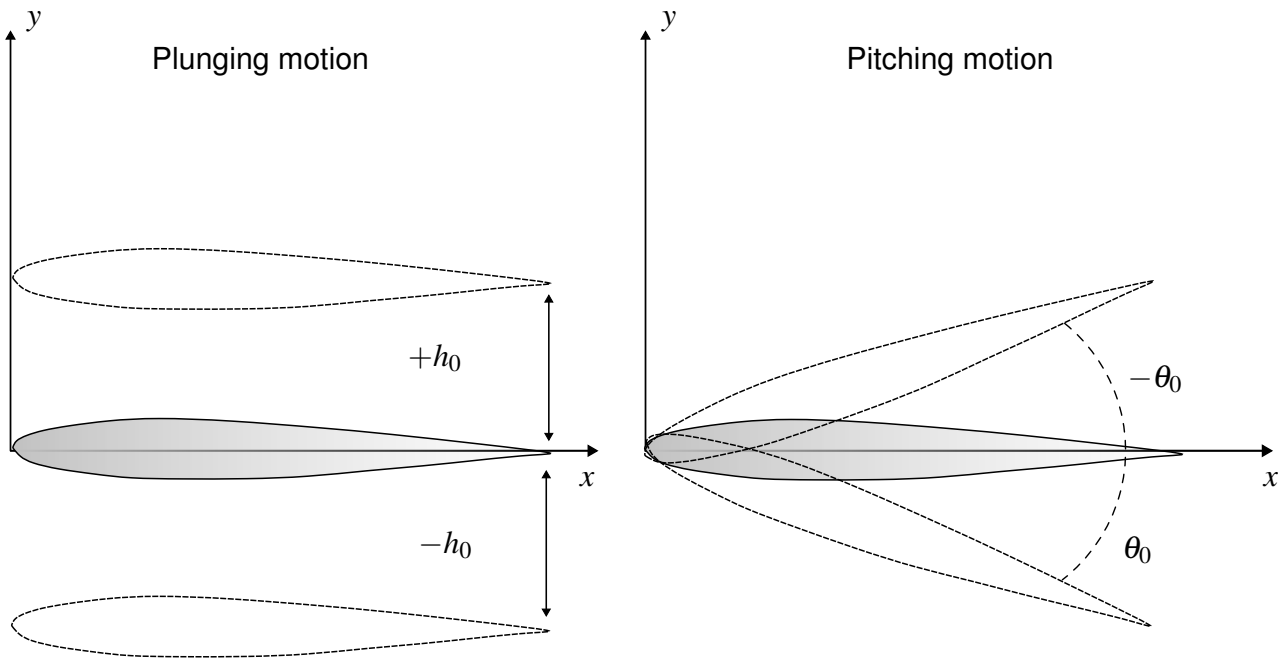

Figure 3.2: plunging and pitching airfoil illustration.

3.1.2 Pitching airfoil motion

The pitching oscillation is described as varying the angle of rotation of the structure with respect to some reference point on the airfoil edge. This can be illustrated in figure 3.2, where the airfoil is undergoing pitching motion about the leading edge of the airfoil. The pitching motion in figure 3.2 is periodic and given by the formula

$$\theta(t) = \theta_0 \cdot sin(\omega \cdot t). \tag{3.2}$$

In equation (3.2), θ_0 is the max pitching angle.

3.2 Equations of motion

For deriving the equations of motion for the airfoil structural system, the Lagrange's equations are used. In order to use Lagrange's equations, its first necessary to identify the potential and kinetic energy terms of the structural system.

Potential energy

The general formula for the potential energy of a spring with a stiffness coefficient k is given by (*Young et al.* (2006)) $E_p = \frac{1}{2}kx^2$. Here, x is the displacement from the equilibrium position of the system and k is the stiffness coefficient. Thus, the total energy potential for the airfoil structural model in figure 3.2, which consists of a vertical and torsional linear spring becomes

$$U = \frac{1}{2}k_h h^2 + \frac{1}{2}k_\theta \theta^2. \tag{3.3}$$

Here h is the vertical plunging distance and θ is the torsional twist angle from the equilibrium.

Kinetic energy

The kinetic energy is derived using the center of mass as the point for the velocity. The total velocity at the center of mass is given by

$$\mathbf{v}_C = \mathbf{v}_P + \dot{\theta}\mathbf{b}_3 \times b[(1+a)-(1+e)]\mathbf{b}_1. \tag{3.4}$$

In equation (3.4), $\mathbf{v}_P$ is the plunging velocity of the reference point P, $\mathbf{v}_P = \dot{h}\mathbf{i}_2$. The overdot used in $\dot{h}$ and $\dot{\theta}$ is the same as the time derivative of h and θ. The unit vector $\mathbf{b}_3$ is the unit vector pointing outside of the two dimensional plane, orthogonal to vector $\mathbf{b}_1$ and $\mathbf{b}_2$ in figure 3.1. By performing the cross product of the second term, the total velocity of the center of mass becomes

$$\mathbf{v}_C = \dot{h}\mathbf{i}_2 + b\dot{\theta}(a-e)\mathbf{b}_2. \tag{3.5}$$

Thus, the total kinetic energy can be written as

$$T = \frac{1}{2}m\mathbf{v}_C \cdot \mathbf{v}_C + \frac{1}{2}I_C\dot{\theta}^2. \tag{3.6}$$

The relationship between the unit vectors in the body reference system expressed in terms of the unit vectors of the stationary system is expressed as

$$\mathbf{b}_1 = \cos(\theta)\mathbf{i}_1 + \sin(\theta)\mathbf{i}_2, \tag{3.7}$$

$$\mathbf{b}_2 = -\sin(\theta)\mathbf{i}_1 + \cos(\theta)\mathbf{i}_2. \tag{3.8}$$

This relationship between unit vectors in the body reference system and the unit vectors of the stationary system is shown in figure 3.1. The I_C term is the moment of inertia about the center of mass of the airfoil. In order to get the expression of the kinetic energy T, the velocity of the center of mass can be written in terms of inertial unit vectors

$$\mathbf{v}_C = -b\dot{\theta}(a-e)\sin(\theta)\hat{\mathbf{i}}_1 + (\cos(\theta)b\dot{\theta}(a-e) + \dot{h})\hat{\mathbf{i}}_2. \tag{3.9}$$

Thus, the first part of the expression for the kinetic energy becomes

$$\frac{1}{2}m\mathbf{v}_C \cdot \mathbf{v}_C = \frac{1}{2}m(b^2\dot{\theta}^2(a-e)^2\sin(\theta)^2 + \cos(\theta)^2 b^2\dot{\theta}^2(a-e)^2 + 2\dot{h}\cos(\theta)\dot{\theta}(a-e)b + \dot{h}^2), \tag{3.10}$$

which can be further simplified to

$$T = \frac{1}{2}m(b^2\dot{\theta}^2x_\theta^2 + 2\dot{h}\cos(\theta)b\dot{\theta}x_\theta + \dot{h}^2) + \frac{1}{2}I_C\dot{\theta}^2. \tag{3.11}$$

$x_\theta = a - e$ in equation (3.11) and is defined as the chordwise offset of the center of mass from the reference point. By using the parallell axis theorem, the kinetic energy term can be written as

$$T = \frac{1}{2}m(2\dot{h}\cos(\theta)b\dot{\theta}x_\theta + \dot{h}^2) + \frac{1}{2}I_P\dot{\theta}^2, \tag{3.12}$$

where $I_P = I_C + mb^2x_\theta^2$ is the moment of inertia about the reference point P.

Generalized forces

The generalized forces related to the generalized coordinates h and θ can be found using the concept of virtual work. This is done by first computing the virtual displacement done by the aerodynamic lift about the center of mass C and similarly by computing the virtual rotation done by the pitching moment about C. The virtual displacement for the center of mass is obtained by replacing the derivatives in the expression for the velocity with the differential d instead. Thus, the virtual displacement of the point C is

$$dP_C = dh\mathbf{i}_2 + bd\theta(a + \frac{1}{2})\mathbf{b}_2. \tag{3.13}$$

Furthermore, since the angular velocity of the airfoil is $\dot{\theta}\mathbf{b}_3$, the virtual rotation simply becomes $\delta\theta\mathbf{b}_3$. The total virtual work for this point then is expressed as

$$dW = F_L(dh + b(a + \frac{1}{2})d\theta) + M_{\frac{1}{4}}d\theta. \tag{3.14}$$

Here, F_L is the lift force and $M_{\frac{1}{4}}$ is the moment about the reference point P coming from the angular velocity of the system. Thus, collecting the terms for $d\theta$ and dh, the generalized forces becomes

$$Q_h = F_L, \tag{3.15}$$

and

$$Q_\theta = F_Lb(a + \frac{1}{2}) + M_{\frac{1}{4}} = M. \tag{3.16}$$

Thus, the generalized force associated with h is the lift and the generalized force associated with θ is the pitching moment about the reference point P.

Lagrange's equations

Lagrange's equation of the system can be derived by using $q_1 = h$ and $q_2 = \theta$.The general form of Lagrange's equations are given by

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = Q_i. \tag{3.17}$$

$L = T - U$ is the difference between the kinetic and potential energy and q is the generalized coordinate and Q_i is the generalized force. By setting $q = h$, one can solve for the plunge coordinate first

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{h}} - \frac{\partial T}{\partial h} + \frac{\partial U}{\partial h} = -m\sin(\theta)\dot{\theta}^2bx_\theta + m\cos(\theta)b\ddot{\theta}x_\theta + m\ddot{h} + k_hh = Q_h. \tag{3.18}$$

Similarly setting for $q = \theta$ leads to

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{\theta}} - \frac{\partial T}{\partial \theta} + \frac{\partial U}{\partial \theta} = m\ddot{\theta}b^2x_\theta^2 + mbx_\theta\ddot{h}\cos(\theta) - mbx_\theta\dot{h}\sin(\theta)\dot{\theta} + mbx_\theta\sin\theta\dot{\theta}\dot{h} + I_C\ddot{\theta} + k_\theta\theta = Q_\theta, \quad (3.19)$$

which is further simplified to

$$m\ddot{\theta}b^2x_\theta^2 + mbx_\theta\ddot{h}\cos(\theta) + I_C\ddot{\theta} + k_\theta\theta = Q_\theta. \quad (3.20)$$

In matrix form, this system of equations can be written as

$$\begin{bmatrix} m & mbx_\theta\cos\theta \\ mbx_\theta\cos\theta & I_P \end{bmatrix}\begin{bmatrix}\ddot{h}\\ \ddot{\theta}\end{bmatrix} + \begin{bmatrix} 0 & -mbx_\theta\sin\theta\dot{\theta} \\ 0 & 0 \end{bmatrix}\begin{bmatrix}\dot{h}\\ \dot{\theta}\end{bmatrix} + \begin{bmatrix} k_h & 0 \\ 0 & k_\theta \end{bmatrix}\begin{bmatrix}h\\ \theta\end{bmatrix} = \begin{bmatrix}Q_h\\ Q_\theta\end{bmatrix} \quad (3.21)$$

Nonlinear spring

The system of equations discussed so far assumed the torsional spring and the plunging spring to be linear. In order to investigate the effects structural nonlinearities have on system, a cubic term is introduced in the spring moment and the spring force, and the equation of motions becomes

$$m\ddot{h} + mbx_\theta\cos\theta\ddot{\theta} - mbx_\theta\sin\theta\dot{\theta}^2 + k_hh + k_{hnl}h^3 = Q_h \quad (3.22)$$

for the plunge and

$$mbx_\theta\cos\theta\ddot{h} + I_P\ddot{\theta} + k_\theta\theta + k_{\theta nl}\theta^3 = Q_\theta \quad (3.23)$$

for the pitching motion. The values of the nonlinear coefficients are chosen to be $k_{hnl} = 0.01k_h$ and $k_{\theta nl} = 0.01k_\theta$. The cubic terms have a hardening effect since they are positive. The choice of using a cubic term is because this makes the spring moment $F_\theta(\theta) = k_\theta + k_{nonlin-\theta}\theta^3$ and the spring force $F_h(h) = k_hh + k_{nonlin-h}h^3$ nonlinear functions. Furthermore, the cubic spring terms must come from an elastic potential of fourth order. The elastic potential function for the cubic force and moment terms will therefore have a parabola shape, which makes it bounded below. This is an requirement to avoid stability of the system.

System of equations

The full system of equations in matrix for for the linear spring can be written as

$$\mathbf{M}\ddot{\mathbf{q}}(t) + \mathbf{D}\dot{\mathbf{q}}(t) + \mathbf{k}_{lin}\mathbf{q}(t) = \mathbf{F}(t), \quad (3.24)$$

and similarly for the nonlinear spring, the full system of equations in matrix form becomes

$$\mathbf{M}\ddot{\mathbf{q}}(t) + \mathbf{D}\dot{\mathbf{q}}(t) + \mathbf{k}_{lin}\mathbf{q}(t) + \mathbf{k}_{nonlin}\mathbf{q}^3(t) = \mathbf{F}(t). \quad (3.25)$$

Here,

$$\mathbf{M} = \begin{bmatrix} m & mbx_\theta\cos\theta \\ mbx_\theta\cos\theta & I_P \end{bmatrix} \quad (3.26)$$

is known as the mass matrix and

$$\mathbf{D} = \begin{bmatrix} d_h & -mbx_\theta\sin\theta\dot{\theta} \\ 0 & d_\theta \end{bmatrix} \quad (3.27)$$

is the matrix containing the damping terms d_h, d_θ, related to the plunging and torsional springs and additionally the nonlinear term $-mbx_\theta \sin\theta\dot{\theta}$.

$$\mathbf{k}_{lin} = \begin{bmatrix} k_h & 0 \\ 0 & k_\theta \end{bmatrix} \tag{3.28}$$

contains the linear spring coefficient terms and

$$\mathbf{k}_{nonlin} = \begin{bmatrix} k_{nonlin-h} & 0 \\ 0 & k_{nonlin-\theta} \end{bmatrix} \tag{3.29}$$

is the nonlinear spring matrix, containing the nonlinear spring terms. $\mathbf{q}(t) = \begin{bmatrix} h(t) \\ \theta(t) \end{bmatrix}$ is the state vector for the airfoil, containing the plunging height $h(t)$ and pitching rotation angle $\theta(t)$ at time t from the equilibrium position.

$$\mathbf{F}(t) = \begin{bmatrix} F_L \\ M \end{bmatrix} \tag{3.30}$$

is a 1×2 column vector, which contains the lift force F_L and the total moment M at time t.

3.2.1 Neural Network generated spring

In the equations of motion for the structural model, the analytical spring force and moment contribution from the plunging and pitching nonlinear springs is

$$F_h(h) = k_h h + k_{nonlin-h} h^3, \tag{3.31}$$

and

$$F_\theta(\theta) = k_\theta \theta + k_{nonlin-\theta} \theta^3. \tag{3.32}$$

The idea is to use a neural network as a tool for reproducing or approximating the analytical expressions for the plunging and pitching nonlinear spring terms and replace them in the equation of motions.

Short description of neural networks

The domain of artificial neural networks belongs to the bigger domain known as machine learning. The field of machine learning can be divided into three main parts which are supervised learning, unsupervised learning and reinforcement learning. Supervised learning is defined as the process of using labeled input datasets for training algorithms in classifying the output data (*IBM*). Unsupervised learning differs from supervised learning where contrary to supervised learning, it does not use labeled input data for training algorithms in order to classify the output data. Reinforcement learning is based on training an agent to maximize its reward, based on previous training experience and does not need labeled input data during the training. The field of artificial neural networks belongs to the supervised learning part of machine learning.

Multilayered perceptors

The concept of multilayered perceptrons (MLP) was originally presented by McCulloch and Pitts (*Mcculloch and Pitts* (1943)), who tried to explain the mechanisms behind which the human brain tries to solve complex tasks. A multilayered perceptor consists of one input layer, one output layer and multiple hidden layers in between the input and output layer.The input

data are fed to the input layer, then processed through the hidden layers and subsequently the final results are presented in the output layer. The training process of an MLP consists of two parts, forward propagation and backward propagation(*Hecht-Nielsen* (1992)).

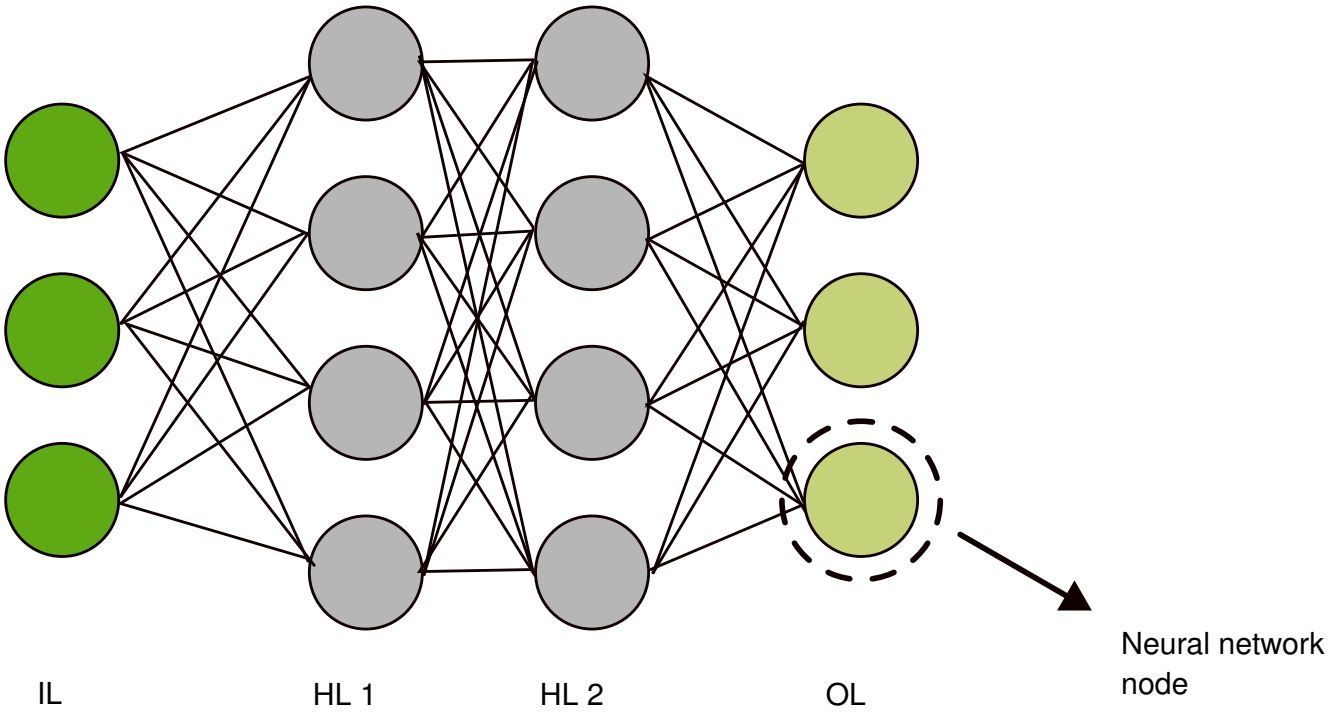

Figure 3.3: Neural network, IL = Input Layer, HL = Hidden Layer and OL = Ouput Layer.

Training process

The forward propagation process describes how the input data is propagated from the input layer to the output layer. The output value for a given neuron node from the input layer k is given by the formula:

$$y_j^k = f(\sum_{i=1}^{n} w_{ij}^k x_{zi} + b_j^k), j = 1, ...m, \tag{3.33}$$

Where y_j^k is the j-th output neural network node from the hideen layer k. In figure 3.3, the green nodes represent the input values, the grey nodes are the hidden neural network nodes and the yellow are the output neural network nodes. The lines connecting the neural nodes have each a weight associated with it, given by w_{ij}. The subcsript of w_{ij} denotes the weight between the i'th neuron in the $k-1$ layer and the j'th neuron in the layer k. The b_j^k values are known as the bias terms, where the j subscript denotes the bias of the j'th neuron in layer k. The f function is the activation function for the network. When it comes to the back propagation process, algorithms based on gradients are the most common used algorithms. For a more detailed explanation of how back propagation works, refer to *Hastie et al.* (2009).

Training of the spring moment function

The torsional spring is responsible for the spring moment in the system of equations. For the linear spring, the moment from the torsional spring due to some rotation angle $\theta(t)$ is given by

$$M_{linear-spring}(t) = k_\theta \theta(t), \tag{3.34}$$

and for the nonlinear spring it is given by

$$M_{nonlinear-spring}(t) = k_\theta \theta(t) + k_{nonlin-\theta}\theta^3(t). \tag{3.35}$$

The goal is to train an artificial neural network which approximates the functions $M_{linear-spring}(t)$ for the linear spring moment and $M_{nonlinear-spring}(t)$ for the nonlinear spring moment such that one gets $M^{NN}_{linear-spring}(t) \approx M_{linear-spring}(t)$ and $M^{NN}_{nonlinear-spring}(t) \approx M_{nonlinear-spring}(t)$, where $M^{NN}_{linear-spring}(t)$ is the trained linear neural network moment function and $M^{NN}_{nonlinear-spring}(t)$ is the trained nonlinear neural network moment function.

Training of the spring force function

Similar to the moment function of the spring, the force function of the spring is computed as a function of the plunging height $h(t)$ given by

$$F_{linear-spring}(t) = k_h h(t), \tag{3.36}$$

for the linear spring and

$$F_{nonlinear-spring}(t) = k_h h(t) + k_{nonlin-h}h^3(t), \tag{3.37}$$

for the nonlinear spring. By using an artificial neural network, one can generate a linear and nonlinear function which approximates these analytical functions, such that $F^{NN}_{linear-spring}(t) \approx F_{linear-spring}(t)$ for the linear spring and $F^{NN}_{nonlinear-spring}(t) \approx F_{nonlinear-spring}(t)$. Here, $F^{NN}_{linear-spring}(t)$ is the trained linear neural network force function and $F^{NN}_{nonlinear-spring}(t)$ is the trained nonlinear neural network force function.

Modification of equations of motion

From the equation of motions for the spring airfoil system, the force and moment contributions from the spring terms are given by

$$\mathbf{F}_{spring}(t) = \mathbf{K}_{linear}\mathbf{q}(t), \tag{3.38}$$

for the linear spring and similarly for the nonlinear spring

$$\mathbf{F}_{nonlin-spring}(t) = \mathbf{K}_{linear}\mathbf{q}(t) + \mathbf{K}_{nonlinear}\mathbf{q}^3(t). \tag{3.39}$$

These functions are only dependent on the state of the system, since the stiffness matrices $\mathbf{K}_{linear}$ and $\mathbf{K}_{nonlinear}$ are constant. The linear neural network generated spring moment and spring force can be put in vector form, such that $\mathbf{F}^{NN}_{linear-spring}(t) = \begin{bmatrix} F^{NN}_{linear-spring}(t) \\ M^{NN}_{linear-spring}(t) \end{bmatrix}$ and similarly for the nonlinear case $\mathbf{F}^{NN}_{nonlinear-spring}(t) = \begin{bmatrix} F^{NN}_{nonlinear-spring}(t) \\ M^{NN}_{nonlinear-spring}(t) \end{bmatrix}$.Thus, the equations of motion can be modified for the linear and nonlinear model to involve these terms such that

$$\mathbf{M}\ddot{\mathbf{q}}(t) + \mathbf{D}\dot{\mathbf{q}}(t) + \mathbf{F}^{NN}_{linear}(t) = \mathbf{F}(t) = \begin{bmatrix} F_L \\ M \end{bmatrix} \tag{3.40}$$

and

$$\mathbf{M}\ddot{\mathbf{q}}(t) + \mathbf{D}\dot{\mathbf{q}}(t) + \mathbf{F}^{NN}_{nonlinear}(t) = \mathbf{F}(t) = \begin{bmatrix} F_L \\ M \end{bmatrix}. \tag{3.41}$$

Chapter 4

Aeroelastic model

In this chapter, the coupling strategy used between the structural model and the aerodynamic model is defined to make an aeroelastic model for the system. The steps in the simulation scheme for transferring the information between the aerodynamic model and the structural model are presented. Furthermore, the numerical integration schemes for the wake convection part of the aerodynamic model and for the integration of the equations of motion in the structural model are then defined.

4.1 Simulation scheme

The model for solving the dynamics of the structural airfoil can be combined with the unsteady vortex particle method, which is used for solving the aerodynamic loads to create an aeroelastic model. The information of the state of the airfoil from the structural model can be used as input data when computing the corresponding aerodynamic loads and vice versa. Between the aerodynamic model and the structural model, one can transfer information bi-directionally in an iterative sequence (see figure 4.1). This iteration between the structural model and the aerodynamic model is done to continuously improve the estimation of the response of the structural model and the aerodynamic loads from the aerodynamic model. In this work, the body is assumed to be rigid which simplifies the exchange of information between the two models. The chosen time increment is dependent on the discretization of the airfoil

$$\Delta t = \frac{\Delta l}{u_\infty}, \tag{4.1}$$

where Δl is the panel length and $u_\infty = \sqrt{\mathbf{u}_\infty \cdot \mathbf{u}_\infty}$ is the Euclidean norm of the free stream velocity vector. One major advantage of partitioning the system into two models, one for the aerodynamics and the other for the structural part is that for each domain (flow and structure), well established modelling, discretization and solution methods can be applied (*Reimer et al.* (2010), *Braun* (2007)). For instance, the unsteady vortex particle method can be found in the literature (eg. *Katz and Plotkin* (2001), *Roccia et al.* (2020)). The time increment is assumed to be the same for the structural model and the aerodynamic model.

4.2 Simulation scheme - steps

- At the current time step t_i, use the aerodynamic model to convect the wake vortices, from $\mathbf{r}_{vp}(t_i)$ to $\mathbf{r}_{vp}(t_i + \Delta t)$. For the rest of the time t, the wake particles are frozen.

- From the computed aerodynamic loads, the corresponding response of the airfoil is calculated by solving the equation of motions for the airfoil. This is done by replacing the force vector with $\mathbf{F}(t_i) = \begin{bmatrix} F_L(t_i) \\ M(t_i) \end{bmatrix}$, which contains the computed lift $F_L(t_i)$ and the moment term $M(t_i)$.

- With the current state of the structural model $\mathbf{q}(t_i)$, the aerodynamic model is used again to compute the updated aerodynamic loads, without convecting the wake vortex particles. The loads computed from the aerodynamic model are then used as input for the structural model again. This is repeated until convergence is reached or the max number of iterations max_{iter} is reached. Usually 5-6 iterations are necessary in order to reduce the error to be less than 10^{10}. The error is computed as the absolute value of the difference between the two state vectors $|\mathbf{q}^{n+1}(t_i) - \mathbf{q}^n(t_i)| < tol$, where n and $n+1$ indicate the iteration step for the time step t_i.

- After reaching a converging state $\mathbf{q}(t_i)$, this state of the airfoil is used as input for the aerodynamic model for computing the flow field and the final aerodynamic loads.

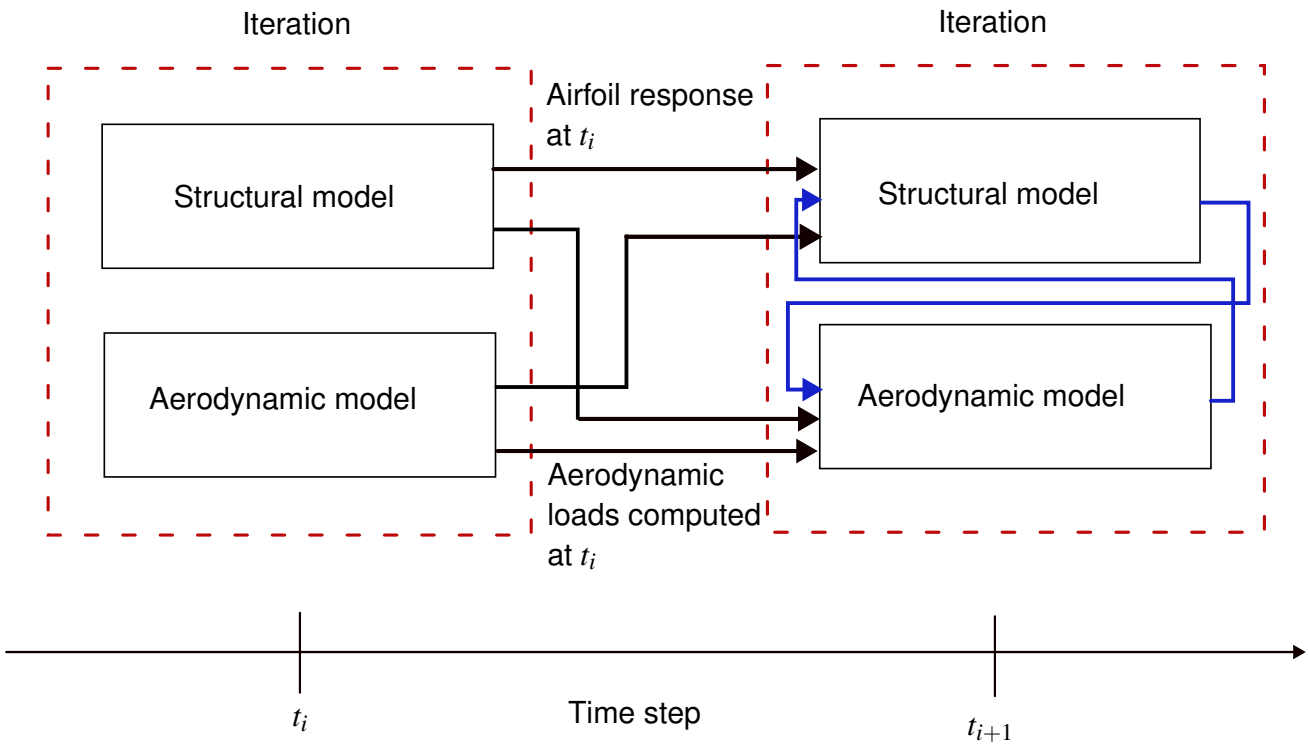

Figure 4.1: Coupling between the structural model and the unsteady vortex particle method.

Figure 4.1 shows how the two models transfer information between each other. The blue arrows is the information exchanged between the models during the iteration step, and the black arrows is the information exchanged from the previous time step to the next time step. Although the aerodynamic model and the structural model are independent computational implementations of the physical fields, the coupling procedure is indeed strong because information is bi-directionally exchanged, and the chosen step, which advances the solution in time, is unique for both simulation environments.

4.3 Numerical integration schemes

In this part, the numerical integration schemes used for convecting the wake particles and for integrating the equation of motions for the structural model are explained.

4.3.1 Wake model

From the velocity of the wake particle, one can approximate the position of the wake particle at time $t_i = t_{i-1} + \Delta t$. The accurate description of the next position of the wake particle is given by the integral

$$\mathbf{x}_w(t_i) = \int_{t_{i-1}}^{t_i} \mathbf{u}_w(\tau) d\tau + \mathbf{x}_w(t_{i-1}). \tag{4.2}$$

The integral can be approximated using a numerical convection scheme.

Explicit Euler scheme

The explicit Euler scheme is the simplest convection scheme for the wake. The formula for this method is given by

$$\mathbf{x}_w(t_i) = \mathbf{x}_w(t_{i-1}) + \Delta t \cdot \mathbf{u}_w(t_{i-1}). \tag{4.3}$$

In the equation above, $\mathbf{x}_w(t_i)$ denotes the position for the vortex wake particle at time t_i and Δt is the time increment and $\mathbf{u}_w(t_{i-1})$ is the velocity of this wake vortex particle at time t_{i-1}.

Adam Bashforth schemes

The Adam Bashforth schemes are a family of multistep methods. This means they depend not only on the velocity at the current time step t_i, but also on previous values of the velocity for the wake particles. The Adam Bashforth schemes are an alternative to the explicit Euler scheme and the first three Adam Bashforth schemes are presented below:

Wake scheme name	Formula
Adam Bashforth 2 step method	$\mathbf{x}_w(t_i) = \mathbf{x}_w(t_{i-1}) + \Delta t \cdot (\frac{3}{2}\mathbf{u}_w(t_{i-1}) - \frac{1}{2}\mathbf{u}_w(t_{i-2}))$
Adam Bashforth 3 step method	$\mathbf{x}_w(t_i) = \mathbf{x}_w(t_{i-1}) + \Delta t \cdot (\frac{23}{12}\mathbf{u}_w(t_{i-1}) - \frac{16}{22}\mathbf{u}_w(t_{i-2}) + \frac{5}{12}\mathbf{u}_w(t_{i-3}))$
Adam Bashforth 4 step method	$\mathbf{x}_w(t_i) = \mathbf{x}_w(t_{i-1}) + \Delta t \cdot (\frac{55}{24}\mathbf{u}_w(t_{i-1}) - \frac{59}{24}\mathbf{u}_w(t_{i-2}) + \frac{37}{24}\mathbf{u}_w(t_{i-3}) - \frac{9}{24}\mathbf{u}_w(t_{i-4}))$

Table 4.1: Adam Bashforth 2-4 step method formulas

As one can see from the formulas in table 2.1 of the multistep methods, they depend on the velocity vector computed at earlier time steps. They need enough points before one can use them. Thus for the starting phase, one can use the explicit Euler to compute enough points.

4.3.2 Structural model

The equations of motion for the structural model are a second order differential equation. In order to solve for the equations numerically, these equations must be converted to a set of first order differential equations. Some initial condition state vector $\mathbf{q}(t_0) = [h(t_0), \theta(t_0)]$ and initial force vector $\mathbf{F}(t_0)$ at the beginning of the simulation is necessary. Then one can define $\mathbf{z}_1(t) = \mathbf{q}(t)$ and $\mathbf{z}_2(t) = \dot{\mathbf{q}}(t)$. By taking the time derivative of these vector, one gets $\dot{\mathbf{z}}_1(t) =$

$\dot{\mathbf{q}}(t) = \mathbf{z}_2(t)$ and $\dot{\mathbf{z}}_2(t) = \ddot{\mathbf{q}}(t)$. If the initial conditions are defined, one can find $\dot{\mathbf{z}}_2(t) = \ddot{\mathbf{q}}(t)$ by solving for $\ddot{\mathbf{q}}(t_0)$ in

$$\mathbf{M}\ddot{\mathbf{q}}(t_0) + \mathbf{D}\dot{\mathbf{q}}(t_0) + \mathbf{k}_{lin}\mathbf{q}(t_0) + \mathbf{k}_{nonlin}\mathbf{q}^3(t_0) = \mathbf{F}(t_0), \tag{4.4}$$

In order to solve the equations of motion, a prediction correction method is implemented. The prediction correction method is used because the corrector method adds stability and accuracy to the solution. The disadvantage of using this method is the added computational cost during the correction step of the method. (*Sauer* (2012) pg.342). The formulas used here can be found in *Sauer* (2012).

Algorithm 4 $Euler_{pred-corr}(\mathbf{q}(t_i), \dot{\mathbf{q}}(t_i), \Delta t, tol, max_{iter})$

1: $\mathbf{q}(t_{i+1})_{pred} = \mathbf{q}(t_i) + \Delta t\dot{\mathbf{q}}(t_i)$
2: $iter = 1$
3: **while** $Error > tol$ **and** $iter <= max_{iter}$ **do**
4: $\quad \mathbf{q}(t_{i+1})_{new} = \mathbf{q}(t_i) + \frac{\Delta t}{2}(\dot{\mathbf{q}}(t_i) + \dot{\mathbf{q}}(t_{i+1})_{pred})$
5: $\quad Error = |\mathbf{q}(t_{i+1})_{new} - \mathbf{q}(t_{i+1})_{pred}|$
6: $\quad iter = iter + 1$
7: $\quad \mathbf{q}(t_{i+1})_{pred} = \mathbf{q}(t_{i+1})_{new}$
8: **end while**
9: **return** $\mathbf{q}(t_{i+1})_{new}$

In algorithm 4, the $\mathbf{q}(t_i)$ is the state vector at the current time t_i, max_{iter} is the maximum number of iterations, tol is the tolerance value, and $\dot{\mathbf{q}}(t_i)$ is the time derivative of the state vector at time t_i.

Adam Bashforth 2 Adam Moulton - prediction correction method

The Adam Bashforth 2 order predictor corrector method is one degree higher order than the Euler predictor corrector method. The algorithm of the method is given below.

Algorithm 5 $Adam2_{pred-corrector}(\mathbf{q}(t_{i-1}), \mathbf{q}(t_i), \dot{\mathbf{q}}(t_{i-1}), \dot{\mathbf{q}}(t_i), \Delta t, tol, max_{iter})$

1: $\mathbf{q}(t_{i+1})_{pred} = \mathbf{q}(t_i) + \frac{\Delta t}{2}(3\dot{\mathbf{q}}(t_i) - \dot{\mathbf{q}}(t_{i-1}))$
2: $iter = 1$
3: **while** $Error > tol$ **and** $iter <= max_{iter}$ **do**
4: $\quad \mathbf{q}(t_{i+1})_{new} = \mathbf{q}(t_i) + \frac{\Delta t}{12}(5\dot{\mathbf{q}}(t_{i+1})_{pred} + 8\dot{\mathbf{q}}(t_i) - \dot{\mathbf{q}}(t_{i-1}))$
5: $\quad Error = |\mathbf{q}(t_{i+1})_{new} - \mathbf{q}(t_{i+1})_{pred}|$
6: $\quad iter = iter + 1$
7: $\quad \mathbf{q}(t_{i+1})_{pred} = \mathbf{q}(t_{i+1})_{new}$
8: **end while**
9: **return** $\mathbf{q}(t_{i+1})_{new}$

In algorithm 5, $\mathbf{q}_{pred}(t_{i+1})$ is the prediction step using the Adam Bashforth 2 step method. The $\mathbf{q}_{new}(t_{i+1})$ is the correction step, using the Adam Moulton 2 step method.

Structural model - combined integration scheme

Since the Adam Bashforth - Adam Moulton 2 step prediction correction method depends on a previous solution from a one step method, one can combine the two methods, where for the

first step the Euler predictor - corrector method is used, and for later steps the Adam Bashforth - Adam Moulton 2 step method is used.

Algorithm 6 $Update_{structural-state}(\mathbf{q}(t_{i-1}), \mathbf{q}(t_i), \dot{\mathbf{q}}(t_{i-1}), \dot{\mathbf{q}}(t_i), \Delta t, tol, max_{iter}, time_{step})$

1: **if** $time_{step} == 1$ **then**
2: $\quad \mathbf{q}(t_{i+1})_{new} = Euler_{pred-corr}(\mathbf{q}(t_i), \dot{\mathbf{q}}(t_i), \Delta t, tol, max_{iter}))$
3: **else**
4: $\quad \mathbf{q}_{new} = Adam2_{pred-corr}(\mathbf{q}(t_{i-1}), \mathbf{q}(t_i), \dot{\mathbf{q}}(t_{i-1}), \dot{\mathbf{q}}(t_i), \Delta t, tol, max_{iter}))$
5: **end if**
6: **return** $\mathbf{q}(t_{i+1})_{new}$

In Algorithm 6, the $time_{step}$ is the current time step of the simulation.

Chapter 5

Results and Discussion

5.1 Results

In this chapter, the results regarding the use of the aerodynamic model, the structural model and the aeroelastic model are presented. The chapter is divided into two main parts, where the first part will be focused on the numerical verification of the models, in which the numerical results from the aerodynamic model will be verified using the analytical solution from the thin airfoil theory in chapter 2. The numerical integration scheme used for solving the equations of motion will be verified by solving the linear pendulum problem which has an analytical solution. For the aeroelastic model, an already solved problem known studied by *Fung* (1955) is analyzed, and the results of this problem are available in the literature. The second part of this chapter is the numerical experimentation, where the some interesting cases of study are presented, including the neural network part related to the spring terms in the structural model.

5.2 Numerical verification

The different models are verified seperately, starting with the aerodynamic model, then the structural model and finishing with the aeroelastic model at the end.

5.2.1 Aerodynamic model

Precision quotient

In order to verify the order of the numerical methods used in convecting the wake particles, the numerical precision quotient for the different wake schemes is computed. The order of the Euler method can be derived using the Taylor expansion for a given position vector $\mathbf{r}(t)$

$$\mathbf{r}(t+\Delta t) = \mathbf{r}(t) + \frac{d\mathbf{r}}{dt}(\Delta t) + O(\Delta t^2) \approx \mathbf{r}(t) + \frac{d\mathbf{r}}{dt}(\Delta t). \tag{5.1}$$

Here, the error term $O(\Delta t^2)$ is raised to the power of 2. Thus, the order of the explicit Euler scheme is: $p = 2-1 = 1$, which is a first order method. The precision quotient for a numerical method is computed by first evaluating the position vector $\mathbf{r}_w(t)$ using the time increments Δt, $\frac{\Delta t}{2}$ and $\frac{\Delta t}{4}$ for the numerical scheme. Then the following precision quotient is given by the formula (*Kreiss and Ortiz* (2014) pg.35)

$$Q(t) = \frac{(\mathbf{r}_{w1}(t) - \mathbf{r}_{w2}(t))}{(\mathbf{r}_{w2}(t) - \mathbf{r}_{w3}(t))} \approx 2^p. \tag{5.2}$$

In the equation above, $r_{w1}(t)$, $r_{w2}(t)$ and $r_{w3}(t)$ correspond to the position vector for a given wake particle, using time increment Δt for $r_{w1}(t)$, $\frac{\Delta t}{2}$ for $r_{w2}(t)$ and $\frac{\Delta t}{4}$ for $r_{w3}(t)$. Thus, for each of the methods, the simulation needs to run three times, one for Δt, $\frac{\Delta t}{2}$ and for $\frac{\Delta t}{4}$. Since all the Adam Bashforth wake schemes are using the Euler scheme for the first step, the value of the precision quotient q factors should be 2 for all the cases. The precision quotient q for the wake is calculated by following a specific wake particle and comparing its value using three simulations with different time increments.

Property name	Symbol	Value	Unit
Number of panels	N_{panels}	10	
Time increment	Δt	0.1	s
Chord	c	1	m
Angle of attack	α	$\frac{10\pi}{180}$	
Number of iterations	I	75	
Air mass density	ρ	1.255	kg/m^2
Horizontal free stream velocity	u_∞	1	m/s
Cutoff vortex radius	$cutoff_{rad}$	0.0001	m

Table 5.1: System parameters - q factor verification

In order to compute the precision quotient for the wake, a specific wake particle must be chosen. In this simulation, the wake particle number 30 is chosen as the reference wake vortex when computing the precision quotient for the different wake schemes. The wake particle number 30 corresponds to the wake particle 60 when the time increment is $\frac{\Delta t}{2}$ and 120 when the time increment is $\frac{\Delta t}{4}$. The reason for why the reference wake particle is chosen to be 30 is to avoid calculating the precision quotient near the far end of the wake, where the wake particles are clustered and will affect the value of the precision quotient.

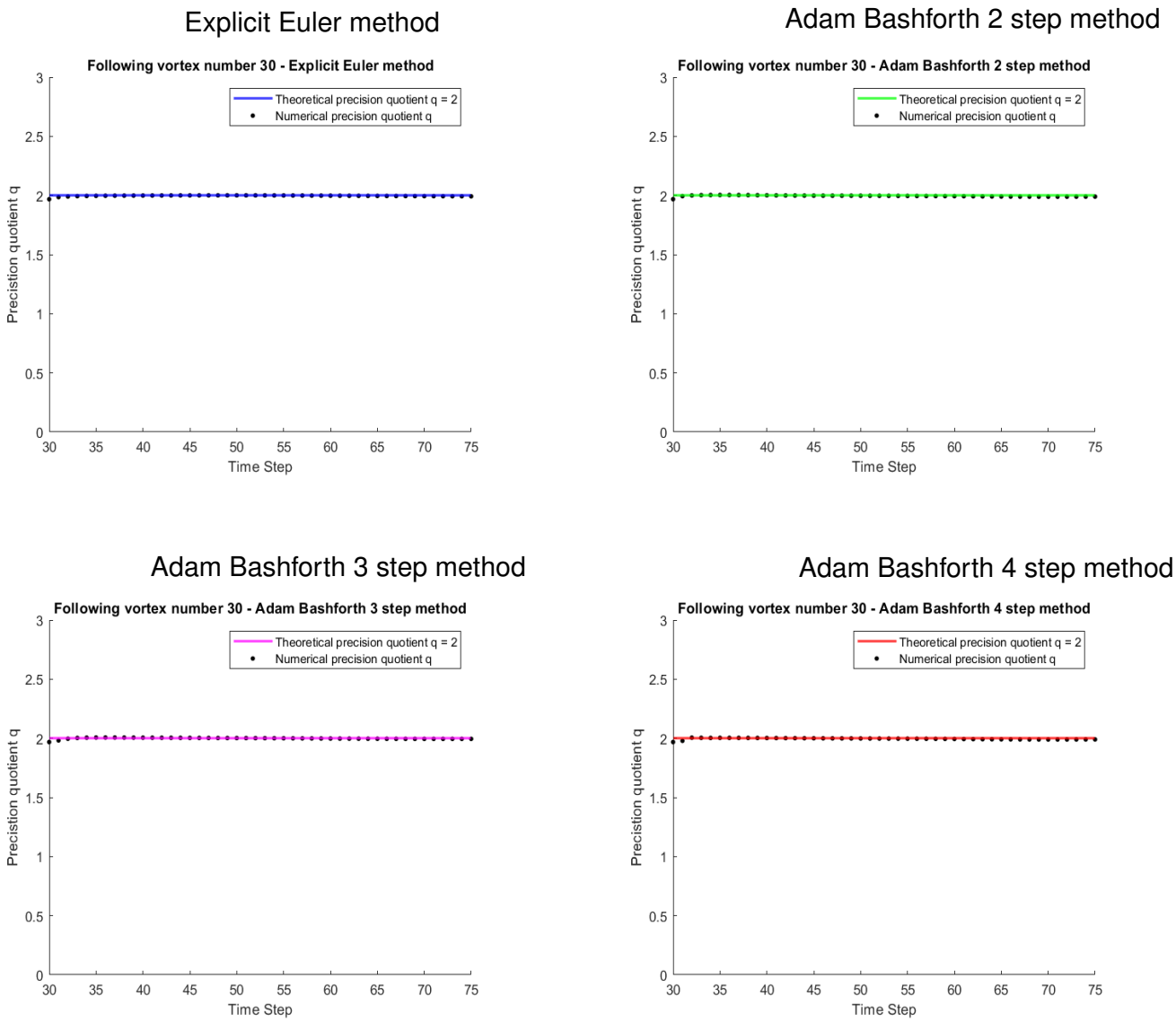

Figure 5.1: precision quotients q for the wake schemes.

Figure 5.1 shows the precision quotients for the different wake schemes. The y axis shows the precision quotient and the x axis shows the time step of the simulation which is using a time increment of Δt. As one can observe, the plots show the correct numerical precision quotients for each time step according to the theory, since the explicit Euler is used for the starting phase for the other Adam Bashforth methods.

Numerical coefficients

To verify the method of the unsteady vortex method, the lift coefficient for the airfoil is calculated. In order for the method to be correct, the lift coefficient has to approach the analytical lift coefficient, which for small angles of attack ($\alpha \in [-\frac{15\pi}{180}, \frac{15\pi}{180}]$) is given by

$$C_l = 2\pi\alpha. \tag{5.3}$$

Property name	Symbol	Value	Unit
Number of panels	N_{panels}	10	
Time increment	Δt	0.025	s
Chord	c	0.25	m
Angle of attack	α	$\frac{10\pi}{180}$	
Number of iterations	I	1000	
Air mass density	ρ	1.255	kg/m^2
Horizontal free stream velocity	u_∞	1	m/s
Cutoff vortex radius	$cutoff_{rad}$	0.0001	m

Table 5.2: System parameters - unsteady vortex particle verification

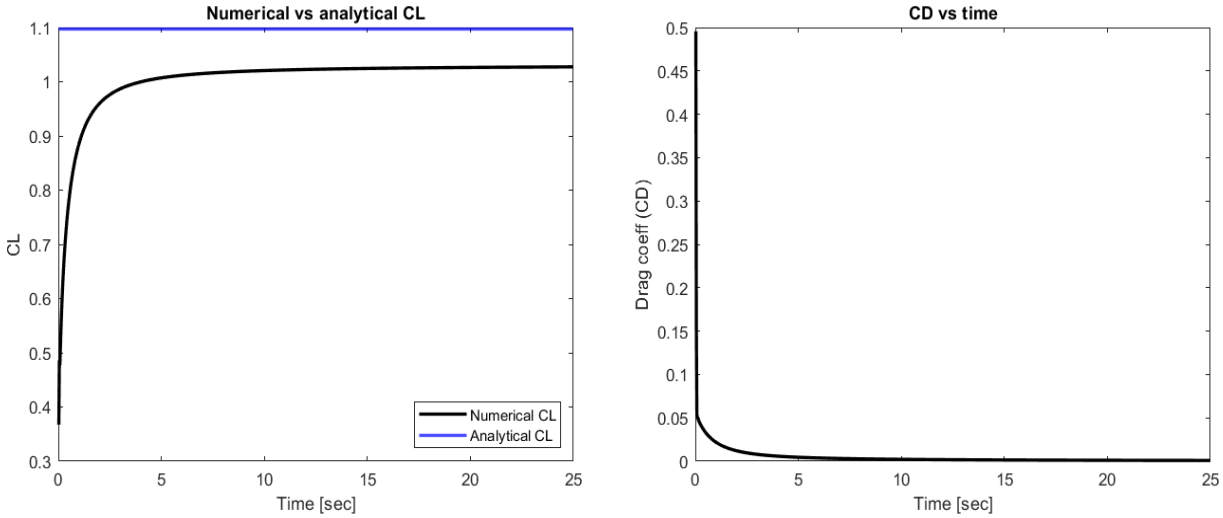

Figure 5.2: Plot of the numerical lift coefficient vs analytical lift coefficient on the left, plot of the drag coefficient on the right.

Figure 5.2 shows the numerical and analytical lift coefficient as a function of time on the left, while it shows the drag coefficient as a function of time for the airfoil on the right. As one can observe, the numerical lift coefficient is approaching the analytical solution as time increases. The rapid increase in the lift coefficient at the beginning is explained by the fact that the wind speed starts at zero and then suddenly the wind speed increases to the steady free stream velocity. This leads to a rapid increase in the $\frac{\Gamma(t+\Delta t)-\Gamma(t)}{\Delta t}$ term when computing the pressure jump across the airfoil, and thus a rapid increase in the lift coefficient at the beginning. As time progresses, the change of the total circulation around the airfoil $\Delta\Gamma(t) = \Gamma(t) - \Gamma(t-\Delta t)$ becomes smaller and thus the lift coefficient approaches the steady state analytical solution. Similarly, the horizontal force coefficient or drag force is rapidly decreasing during the transient since the change in total circulation around in the airfoil $\Delta\Gamma(t)$ goes to zero as time increases and thus the horizontal force component becomes smaller. Based on the lift coefficient plot in figure 5.2, the steady state lift coefficient is $C_L = 1.0272$, which is 0.0728 lower than the analytical lift coefficient. The numerical lift coefficient will approach the analytical lift coefficient, as the angle of attack decreases.

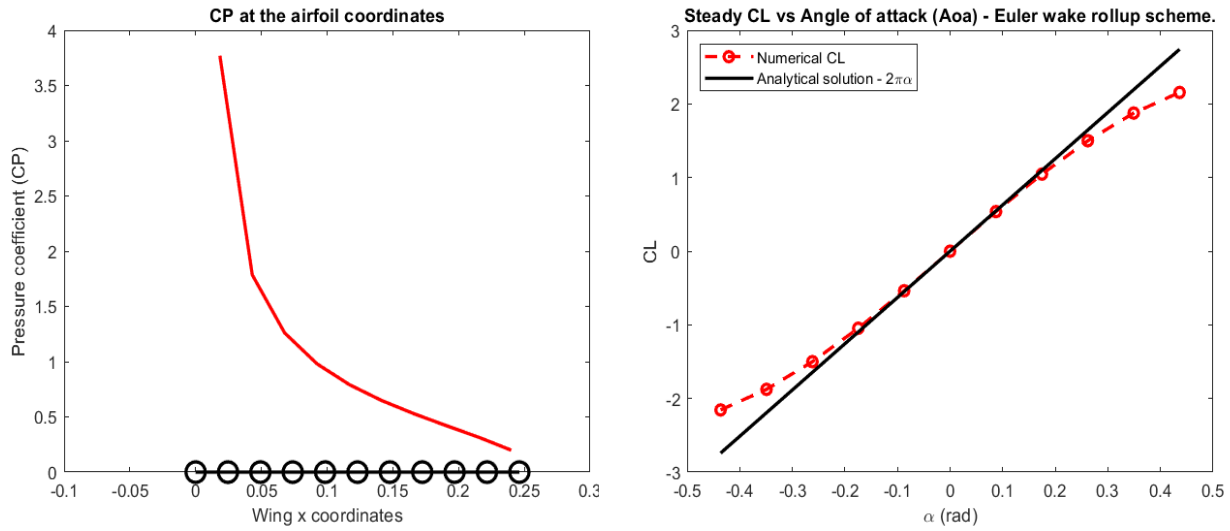

Figure 5.3: Plot of the steady pressure coefficient as a function of the airfoil coordinates in the left figure. The right figure is the steady lift coefficient plotted against the angles of attack in the small angle domain.

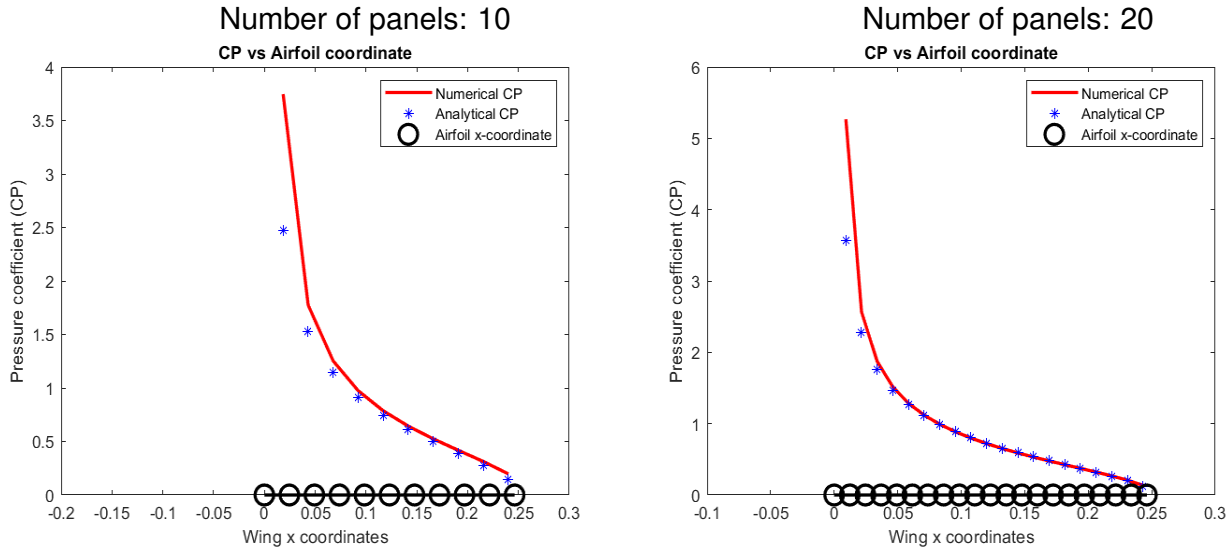

Figure 5.4: Plot of the numerical Cp and the analytical Cp, for both 10 and 20 panels used in the discretization of the airfoil.

In in the left figure of figure 5.4, the steady pressure coefficients for the airfoil are plotted against the airfoil coordinate along the chord line. The pressure coefficient has the highest value at the leading edge and decreases towards the trailing edge of the airfoil. The analytical pressure jump for a thin symmetrical airfoil is given by the formula $\Delta C_p = 4\sqrt{\frac{c-x}{x}}\alpha$, where c is the chord length and x is the x coordinate of the airfoil. (*Katz and Plotkin* (2001) pg.131). As one can see from figure 5.4, the pressure coefficient for the numerical and analytical solution fit when well except near the leading edge. This is because near the leading edge, the symmetrical thin airfoil is singular and thus the model is not correct. The figure on the right of figure 5.4 shows the steady lift coefficient plotted against the angle of attack used for the airfoil. As one can see, the analytical and numerical solutions are almost identical for small angles, but as the angles increases, the difference between the analytical and numerical coefficients increases.

Pitching and plunging motion - verification

To verify the pitching and plunging motion of the airfoil is correct, the numerical results are compared to the analytical Garrick solution (*Garrick* (1936)). For harmonic plunging motions $h(s) = -h_0 \cos(k_h s)$, Garrick's solution can be written in dimensionless form as (*Gülçat* (2010))

$$C_l(s) = -2\pi k_h \frac{k_h}{b}[(G(k_h) + 0.5k_h)\cos(k_h s) + F(k_h)\sin(k_h s)], \tag{5.4}$$

where b is the half chord, $s = \frac{u_\infty t}{b}$ is the dimensionless time, $k_h = \frac{\omega_h b}{u_\infty}$ is the reduced plunging frequency, and $C(k) = F(k) + iG(k)$ is the Theodorsen function. Theodorsen's function is used for modelling changes in amplitude and phase of the unsteady sinusoidal aerodynamic forces relative to the quasi-steady forces for different reduced frequencies. The function is a complex quantity, expressed as a function of reduced frequency

$$C(k) = F(k) + iG(k) = \frac{H_1^{(2)}(k)}{H_1^{(2)}(k) + iH_0^{(2)}(k)} = \frac{K_1(ik)}{K_0(ik) + K_1(ik)}, \tag{5.5}$$

where the $K_j(ik)$ for $j = 0, 1, ...$ are modified Bessel functions of the second kind and $H_n^{(2)}(k)$ are Hankel functions of the second kind. For the reduced frequency $k <= 0.5$, an approximation for $C(k)$ is can written as

$$C(k) = 1 - \frac{0.165}{1 - \frac{0.0455}{k}i} - \frac{0.335}{1 - \frac{0.30}{k}i}. \tag{5.6}$$

For the pitching motion, the lift coefficient is given as

$$\begin{aligned} C_l(s) =& \pi\theta_0[k_\theta \cos(k_\theta s) + ak_\theta^2 \sin(k_\theta s) + 2F(k_\theta)\left(\sin(k_\theta s) + (\frac{1}{2} - a)k_\theta \cos(k_\theta s)\right) + \\ & 2G(k_\theta)(\cos(k_\theta s) - \left(\frac{1}{2} - a\right)k_\theta \sin(k_\theta s))]. \end{aligned} \tag{5.7}$$

In equation (5.7), a is a dimensionless parameter by the half chord, which defines the center of the airfoil and is measured from the mid chord (positive in the aft shaft). In the Garrick formulas, theres no initial angle of attack, which means that for plunging motion the angle of attack is zero and for the pitching motion there's no initial angle of attack (just symmetrical oscillation around the horizontal plane). The effective angle of attack for the plunging motion is

$$\alpha_e(s) = \arctan\left(\frac{\dot{h}(s)}{u_\infty}\right), \tag{5.8}$$

and for the pitching motion the effective angle is the same as the kinematic angle of attack

$$\alpha_e(s) = \theta(s). \tag{5.9}$$

Plunging case

In this section the values of the lift coefficient for the plunging airfoil is verified by comparing the numerical values with Garrick's formula for the lift coefficient in the plunging case. The table below shows the data used for the numerical simulation of the airfoil.

Property name	Symbol	Value	Unit
Wake scheme		Eulers method	
Wind speed	u_∞	1	m/s
Air mass density	ρ	1.255	kg/m^2
Number of panels	N_{panels}	10	
Angle of attack	α	0	
Chord	c	1	m
Max plunging amplitude	h_0	0.1	m
Plunging function	$h(s)$	$-h_0\cos(k_h s)$	

Table 5.3: Parameter values for the plunging motion of the airfoil - comparison between different k_h values.

$k_h = 0.1$

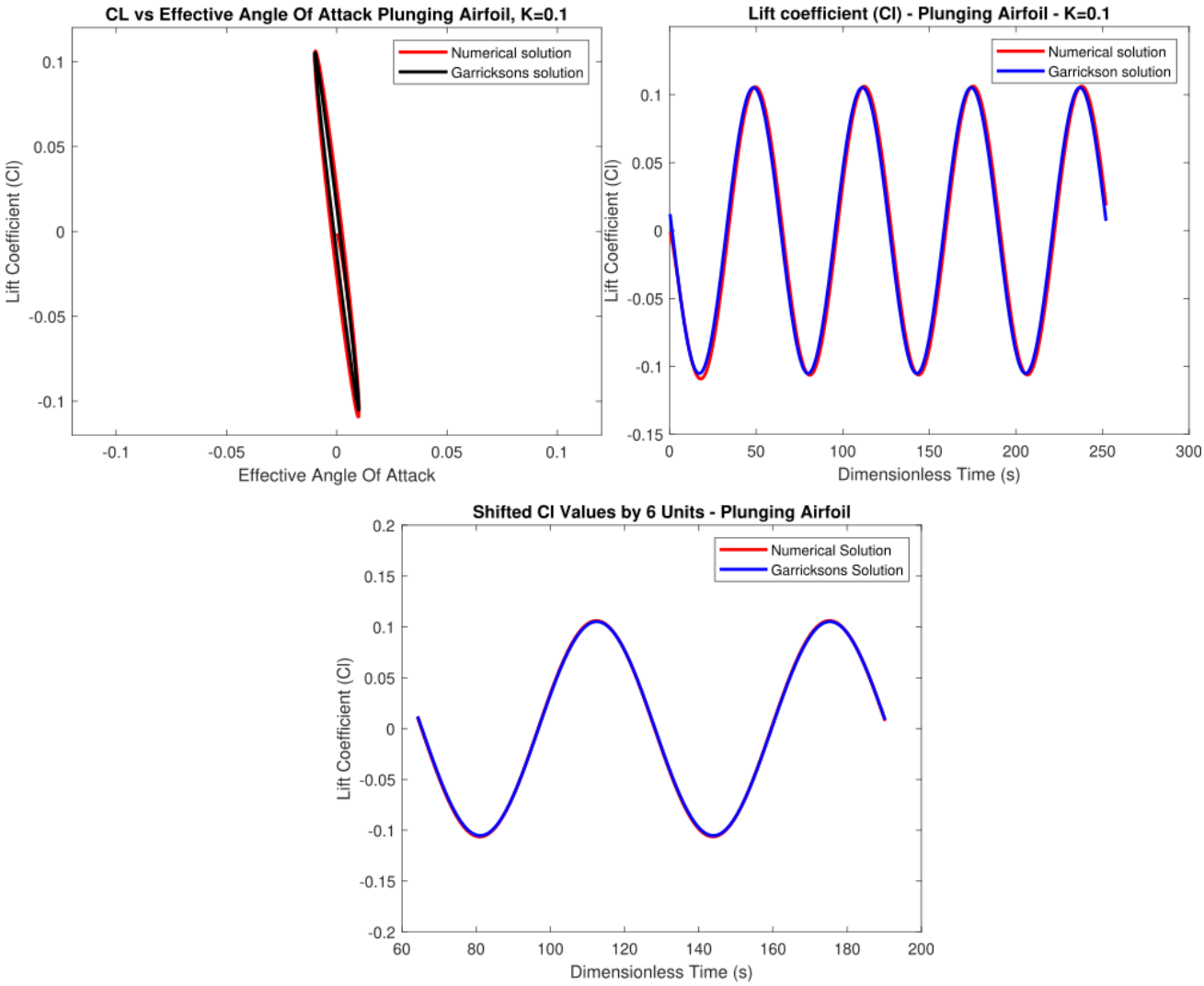

Figure 5.5: Garricks solution vs numerical solution, plunging case, $k_h = 0.1$

Figure 5.5 shows the numerical lift coefficient CL and Garrick's solution when the reduced frequency $k_h = 0.1$ for the plunging case. The plot in the left upper corner is the numerical and Garrick's solution for the lift coefficient plotted against the effective angle of attack. The figure in the right upper corner is the lift coefficient plotted against the dimensionless time for the numerical and Garrick's solution and the figure in the bottom is the shifted plot of the lift coefficient plotted against time. As one can see, the analytical Garrick solution fits well with

the numerical solution. Since Garrick's solution are the steady state solutions, the numerical lift coefficient needs to be shifted in order to overlap the analytical solution as one can see in the bottom figure of figure 5.5.

$k_h = 0.25$

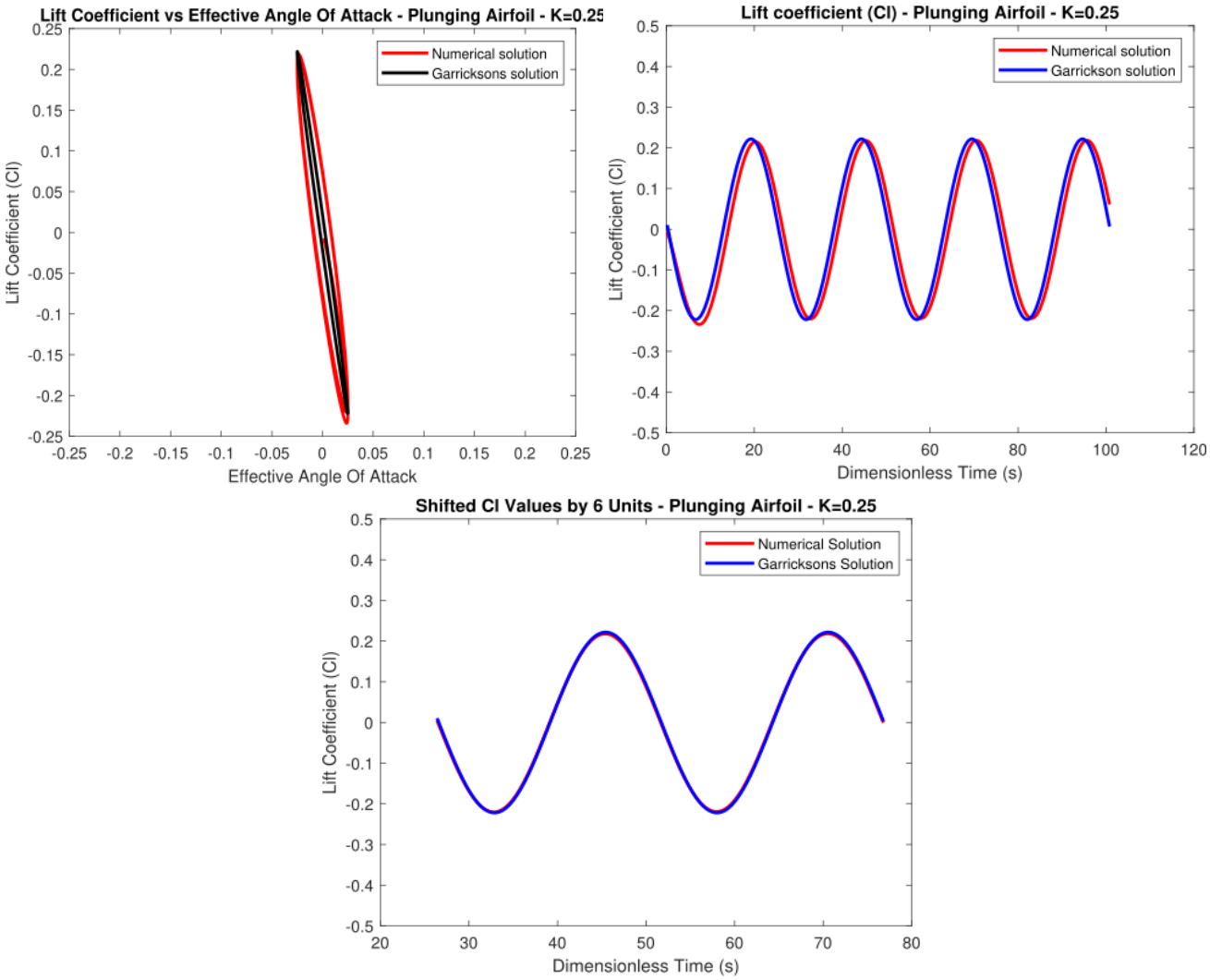

Figure 5.6: Garricks solution vs numerical solution, plunging case, $k_h = 0.25$.

Figure 5.6 shows the similar results as for figure 5.5 for the plunging case when the reduced frequency is $k_h = 0.25$. Although they are similar to the case when $k_h = 0.1$, one can see that the numerical and analytical solutions don't fit as well in the right figure of figure 5.6.

$k_h = 0.5$

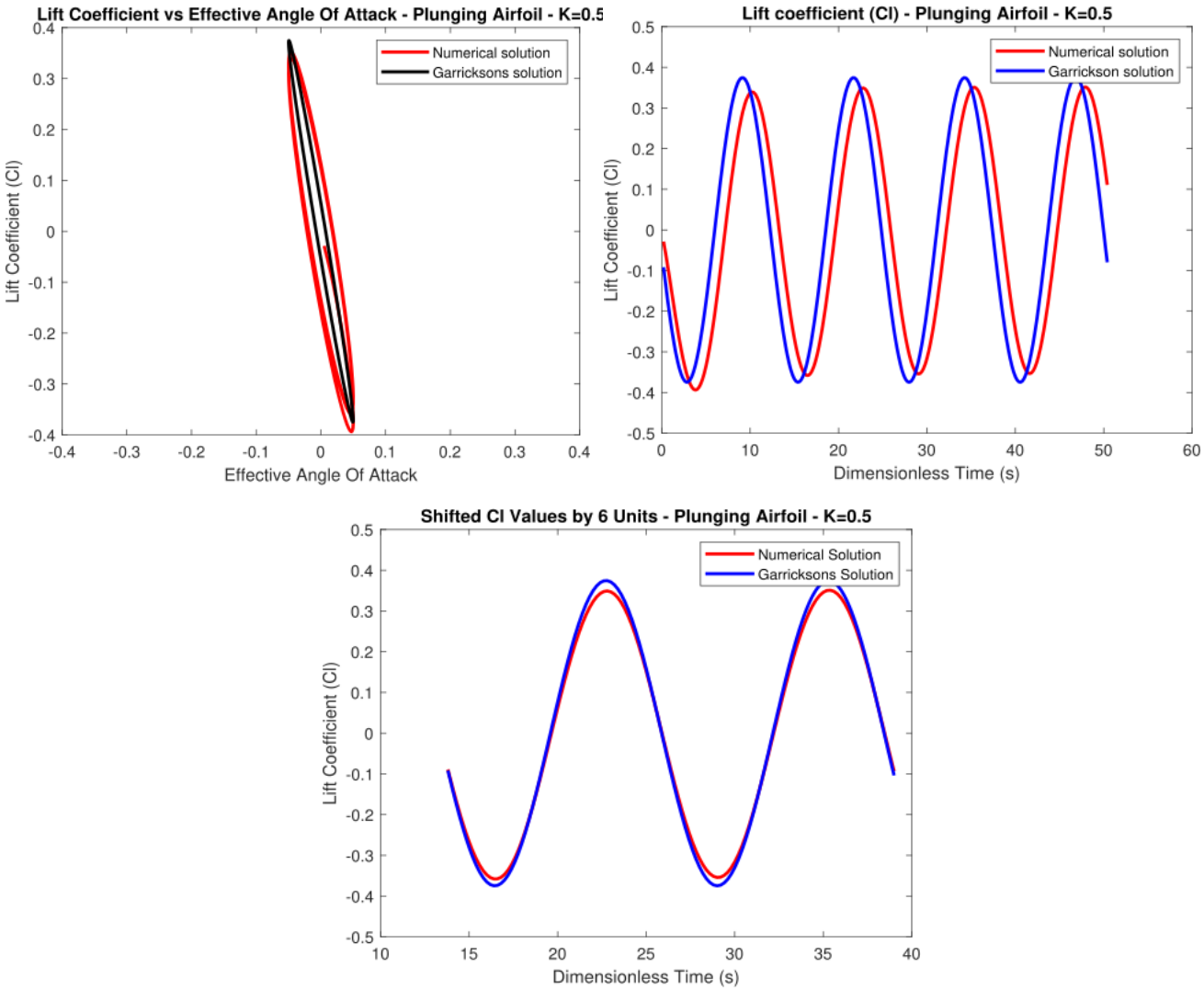

Figure 5.7: Garrick's solution vs numerical solution, plunging case, $k_h = 0.5$.

Similar to figure 5.5 and 5.6, figure 5.7 shows the comparison of Garrick's solution and the numerical solution for the case when $k_h = 0.5$ for the plunging motion. Although the numerical and analytical solutions seems to fit, one can clearly see a difference in phase of the curves from the figure on the right. As one can see on the figure on the right in figure 5.7, the numerical solution is shifted relative to the analytical solution and to a greater extent compared to the cases when $k_h = 0.1$ and $k_h = 0.25$. The reason for this increase in difference between the analytical solution and the numerical solution can be due to the approximation used for the analytical lift coefficient, which was valid for when $k_h <= 0.5$. Thus when $k_h = 0.5$, it is at the end of the range where the approximation is valid.

Number of panels - convergence

Here, figure 5.8 shows how the difference between the analytical and numerical solution for the lift coefficient decreases as the number of panels increases, using all the different wake schemes for the wake particles convection. The metric used for calculating the error is the euclidean norm. Based on figure 5.8, the difference decreases rapidly as the number of panels increases in the beginning, then it flattens out.

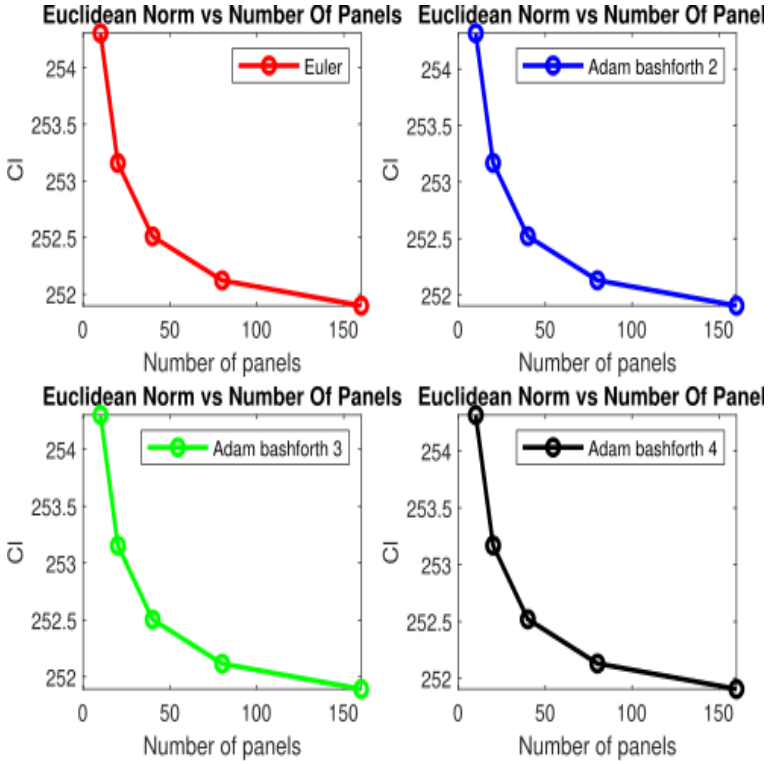

Figure 5.8: Cl error vs number of panels - different wake schemes, plunging case

Pitching case

In this section the values of the lift coefficient for the pitching airfoil is verified by comparing the numerical values with Garrick's formula for the lift coefficient in the pitching case. The table below shows the data used for the numerical simulation of the airfoil.

Property name	Symbol	Value	Unit
Wake scheme		explicit Euler method	
Horizontal free stream velocity	u_∞	1	m/s
Air mass density	ρ	1.255	kg/m^2
Number of panels	N_{panels}	10	
Angle of attack	α	0	
Chord	c	1	m
Max angle displacement	α_{max}	pi/180	
Pitching function	$\theta(s)$	$\alpha_{max}\sin(k_\theta s)$	

Table 5.4: Parameter values for the pitching motion of the airfoil - comparison between different k values.

$k_\theta = 0.1$

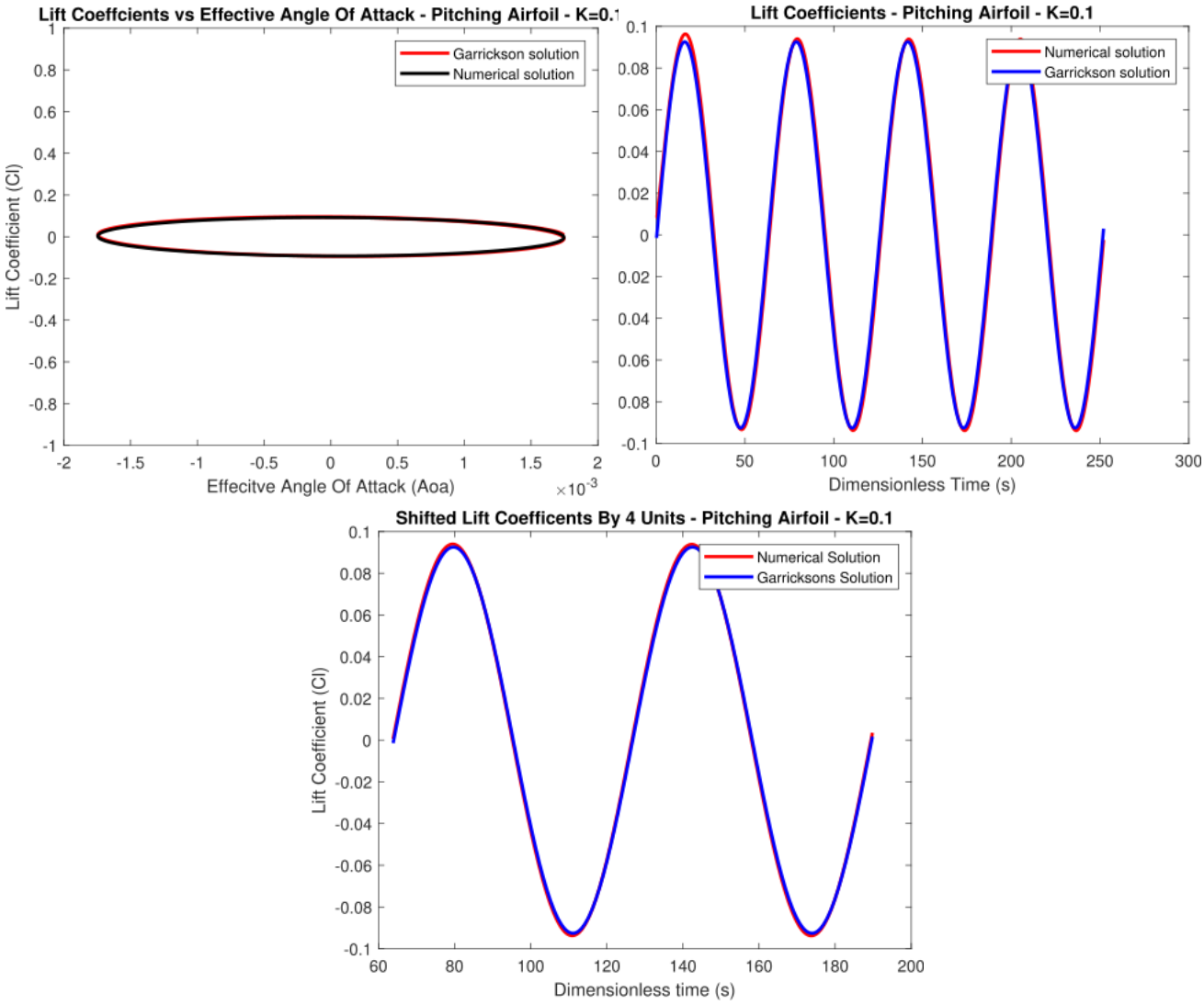

Figure 5.9: Garricks solution vs numerical solution, pitching case, k=0.1

Figure 5.9 shows the numerical lift coefficient and the analytical Garrick's lift coefficient for the pitching case when the reduced frequency is $k_\theta = 0.1$. The top left figure is the numerical and analytical lift coefficient plotted against the effective angle of attack, the top right figure is the plot of the numerical and analytical lift coefficients at the same time and the bottom figure is the shifted numerical solution at the steady state portion to fit the analytical solution. The reason behind the change in phase between the two plots is the same as for the plunging case.

$k_\theta = 0.25$

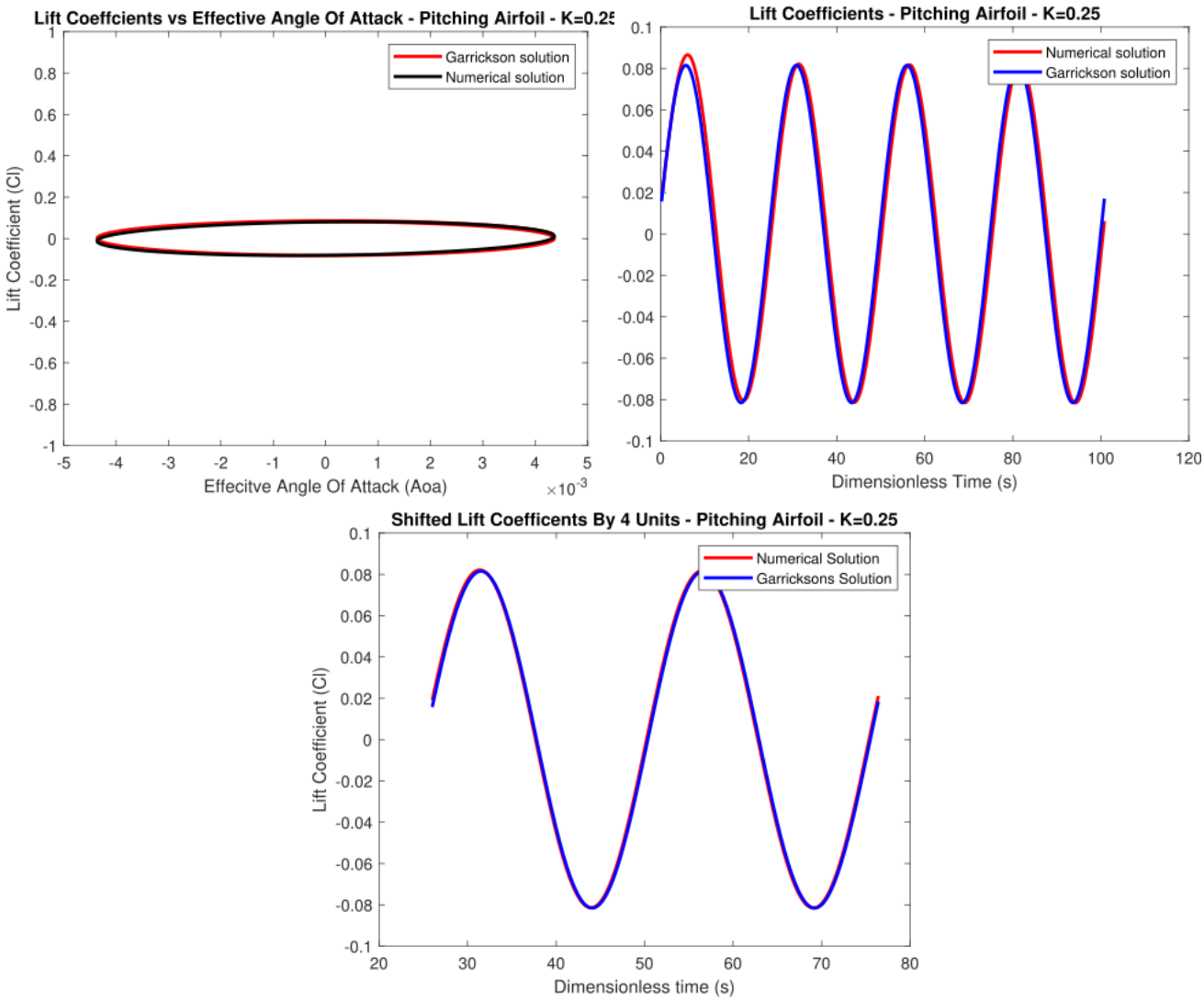

Figure 5.10: Garricks solution vs numerical solution, pitching case, k=0.25

Again, figure 5.10 shows similar plots for the pitching case as in figure 5.9, but for the reduced frequency $k_\theta = 0.25$.

$k_\theta = 0.5$

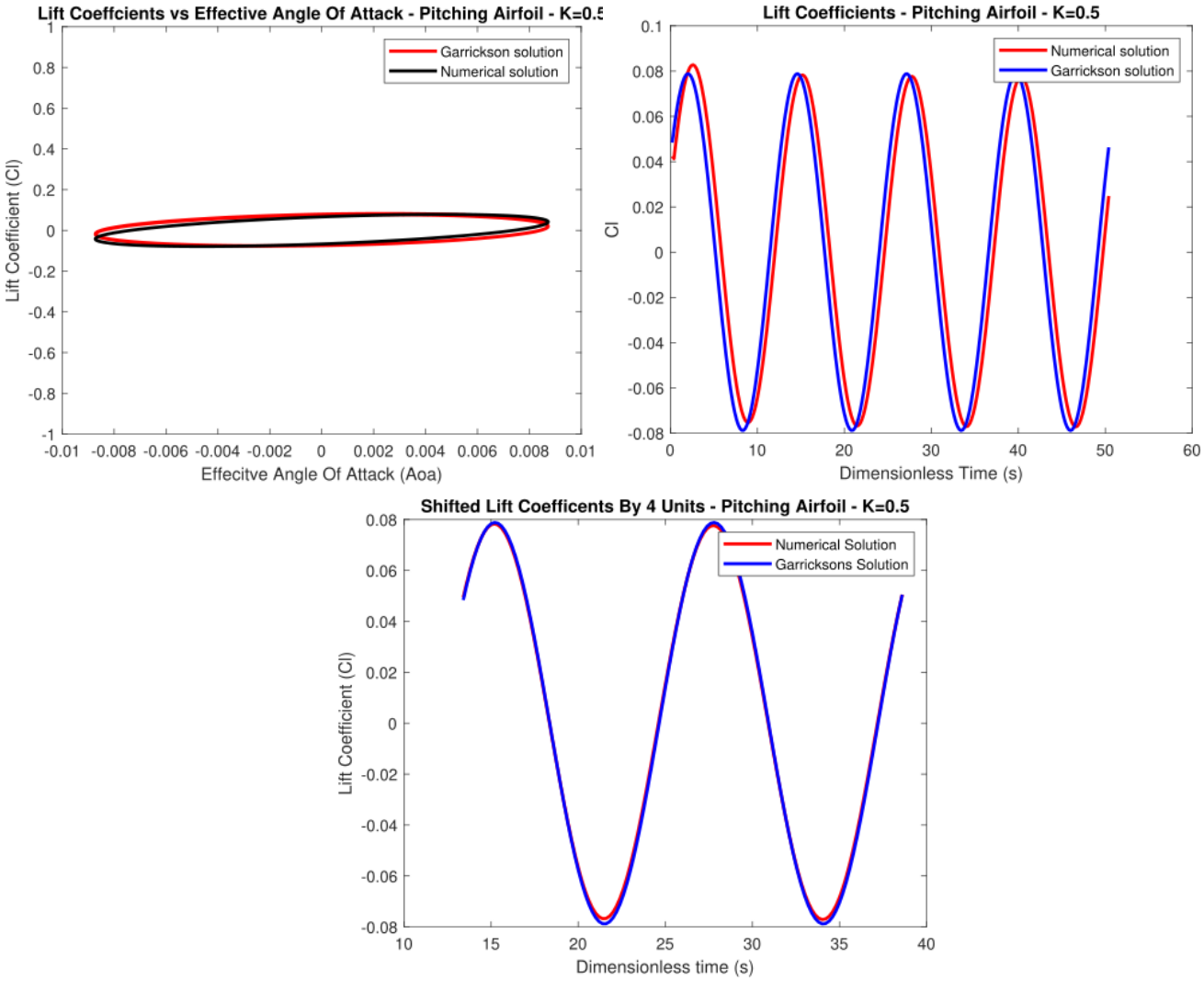

Figure 5.11: Garricks solution vs numerical solution, pitching case, k=0.5

Figure 5.11 shows the comparison of the numerical and analytical lift coefficients for the pitching airfoil when $k_\theta = 0.5$. The figure on the top right of figure 5.11 shows that the numerical solution is shifted to the right relative to the steady state analytical solution. Similarly to the plunging case, the difference between the numerical and analytical solution seems to increase as the reduced frequency increases. Since the approximation of the analytical lift is only valid in the range $k_\theta <= 0.5$, when $k_\theta = 0.5$ the analytical solution is at the end of the range where the approximation is valid and this might explain the difference in the lift coefficients as shown the figure 5.11.

Number of panels - convergence

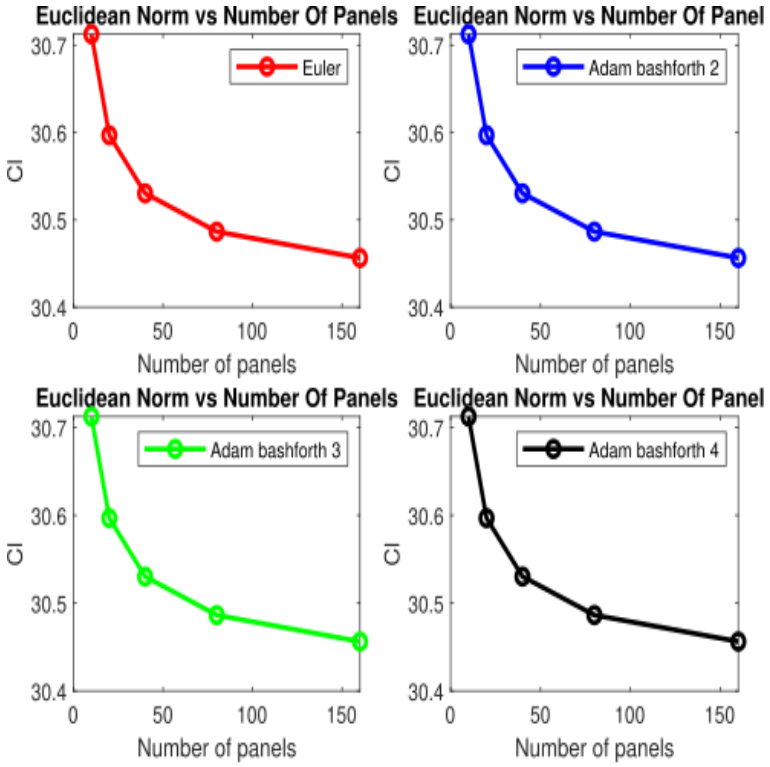

Figure 5.12: Cl error vs number of panels - different wake schemes, pitching case.

Figure 5.12 shows the difference between the lift coefficient from Garrick's solution and the numerical solution as the number of panels increases. The different subplots shows which wake scheme is used for the numerical solution. Similar to the plunging case, the difference decreases and then flattens out as the number of panels exceeds beyond $N_{panels} = 100$.

5.2.2 Structural model

In order to verify that the numerical integration scheme used for solving the equations of motion in the structural model is correct, the numerical integration scheme is used for solving linear pendulum problem. The linear pendulum problem a second order differential equation given by

$$\ddot{\theta} + \frac{g}{L}\theta = 0. \tag{5.10}$$

Since the linear pendulum has an analytical solution, one can compare the analytical solution to the numerical solution to see how close the numerical solution is to the analytical one.

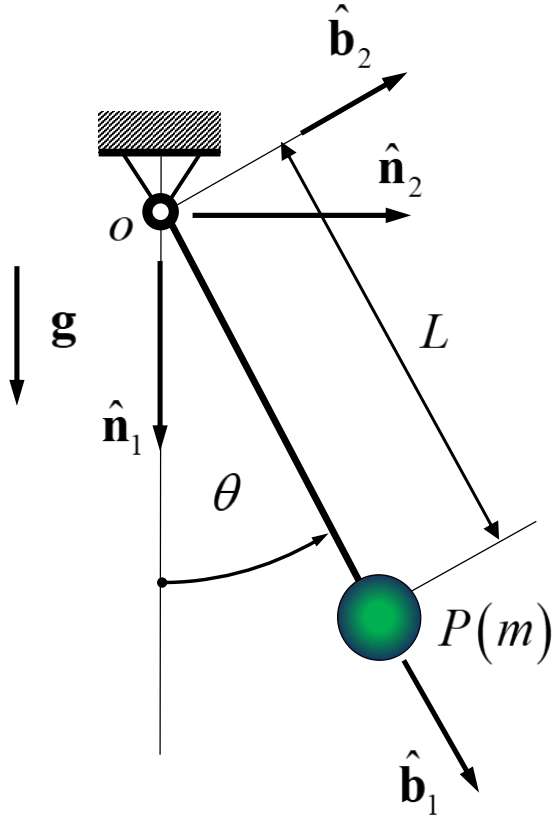

Figure 5.13: Rotating pendulum.

Figure 5.13 shows the setup of the pendulum, where θ is the angle between the pendulum line and the vertical line, $\mathbf{g} = (0, -\mathrm{g})$ is the gravitational acceleration vector and L is the length of the line of the pendulum. This system can be decomposed into a set of first order differentials, such that

$$z_1(t) = \theta(t), z_2(t) = \dot{\theta}(t). \tag{5.11}$$

Thus, the state vector of the system is summarized by $\mathbf{z}(t) = (z_1, z_2)$. The derivative of this vector is given by

$$\frac{d\mathbf{z}}{dt} = (z_2, -\frac{g}{L} z_1), \tag{5.12}$$

and the initial conditions for the simulation of the pendulum is given in the table below.

Property name	Symbol	Value	Unit
line length	L	1	m
gravitational constant	g	9.81	m/s^2
tolerance	tol	0.0001	
time increment	Δt	0.01	s
Initial angle	θ_0	$\frac{\pi}{180}$	
Initial angle velocity	$\frac{d\theta_0}{dt}$	0	s^{-1}
Number of iterations	I	1000	
Max iterations	Max_{iter}	100	

Table 5.5: Linear pendulum system parameters

In order for the integration scheme to be correct, it must approximate the analytical solution

$$\theta(t) = \theta_0 \cos \sqrt{\frac{g}{L}} t. \tag{5.13}$$

The numerical integration scheme used for this problem is the combined predictor corrector scheme described in chapter 5.

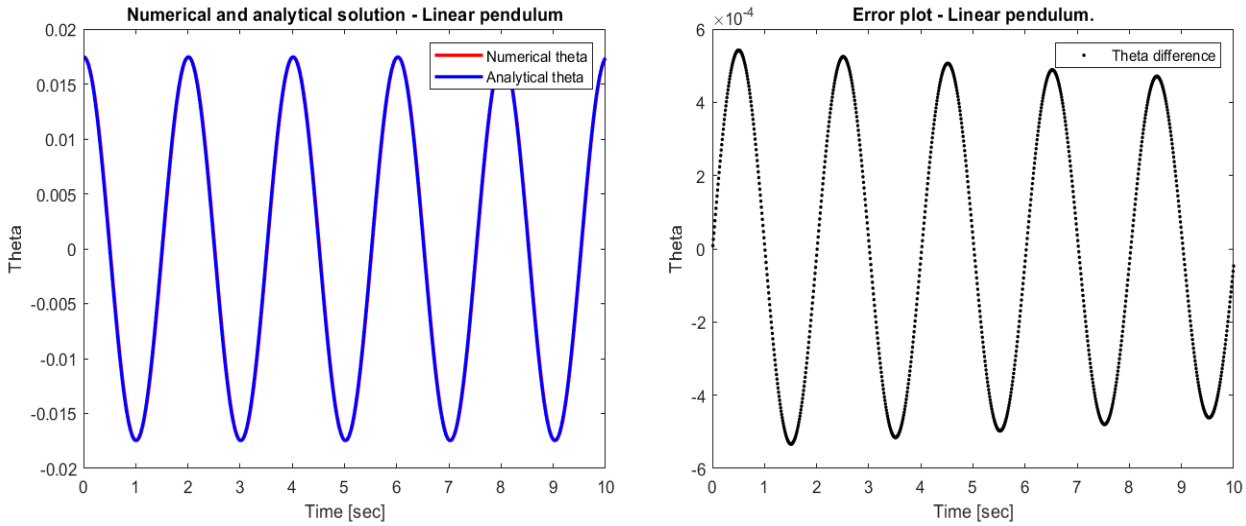

Figure 5.14: Numerical vs analytical solution to the linear pendulum and the error plot between the numerical and analytical solution.

As one can observe from figure 5.6, the figure in the left showing the numerical and analytical solutions of θ seems to fit well. Similarly, the error on the right in figure 5.6 shows that the error is within the tolerance value of 0.0001 and decreasing. Its important to note that the numerical solution is highly dependent on the time increment and thus when the time increment is high the numerical solution fits poorly relative to the analytical solution (*Sauer* (2012)).

5.2.3 Aeroelastic model

Aeroelastic flutter of a suspension bridge

In order to verify that the aeroelastic code is at least partially correct under sub-critical, critical and super-critical conditions, the well known suspension bridge problem taken from Fung *Fung* (1955) is considered. The suspension bridge is modelled as a symmetric thin airfoil with uniform mass distribution.

Definition of aeroelastic flutter

Aeroelastic flutter can be described as a dynamic instability of a flight vehicle assoiated with the interaction of aerodynamic, elastic and inertial forces (*Hodges and Pierce* (2011) pg.175). It's a self-excited oscillatory instability, in which case the aerodynamic forces acting on the body and the natural modes of the vibration of frequency of the body produce an oscillatory motion with increasing amplitude as time progresses.

Parameter	Symbol	Value	Unit
Semi chord	b	9.144	m
Mass per length	m	12879.78698	kg/m
Radius of gyration	r_α	0.788796	
Bending natural frequency	ω_h	0.88034	s^{-1}
Torsion natural frequency	ω_α	1.5524	s^{-1}

Table 5.6: Suspension bridge - parameters.

Trimmed wake

Due to the high computational cost of convecting the wake particles in the simulation, the following approach for convecting the wake particles is used. When the horizontal distance between the wake particle and the trailing edge of the airfoil is $\Delta x >= 10 \cdot c$, where c is the chord length of the airfoil, the influence from this wake particle on the airfoil and other wake particles closer to the airfoil is set to 0. This approximation will reduce the computational cost significantly and can be justified since the wake particles far away from the airfoil will have little to no influence on the airfoil. The table above shows the necessary properties of the bridge and the velocities for the sub-critical, critical and super-critical velocities and there's no damping involved. From the radius of gyration, one can find the moment of inertia bout the reference point: $r_\alpha = \sqrt{\frac{I_p}{mb^2}}$. The spring coefficients can be found using the formula $\omega_\theta = \sqrt{\frac{k_\theta}{I_P}}$ for the pitching spring coefficient and $\omega_h = \sqrt{\frac{k_h}{m}}$ for the vertical spring coefficient. Since the mass of the bridge is assumed to be distributed uniformly, $x_\theta = 0$ and the equations of motions becomes

$$\begin{bmatrix} m & 0 \\ 0 & I_P \end{bmatrix} \begin{bmatrix} \ddot{h} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} k_h & 0 \\ 0 & k_\theta \end{bmatrix} \begin{bmatrix} h \\ \theta \end{bmatrix} = \begin{bmatrix} F_L \\ M \end{bmatrix}. \tag{5.14}$$

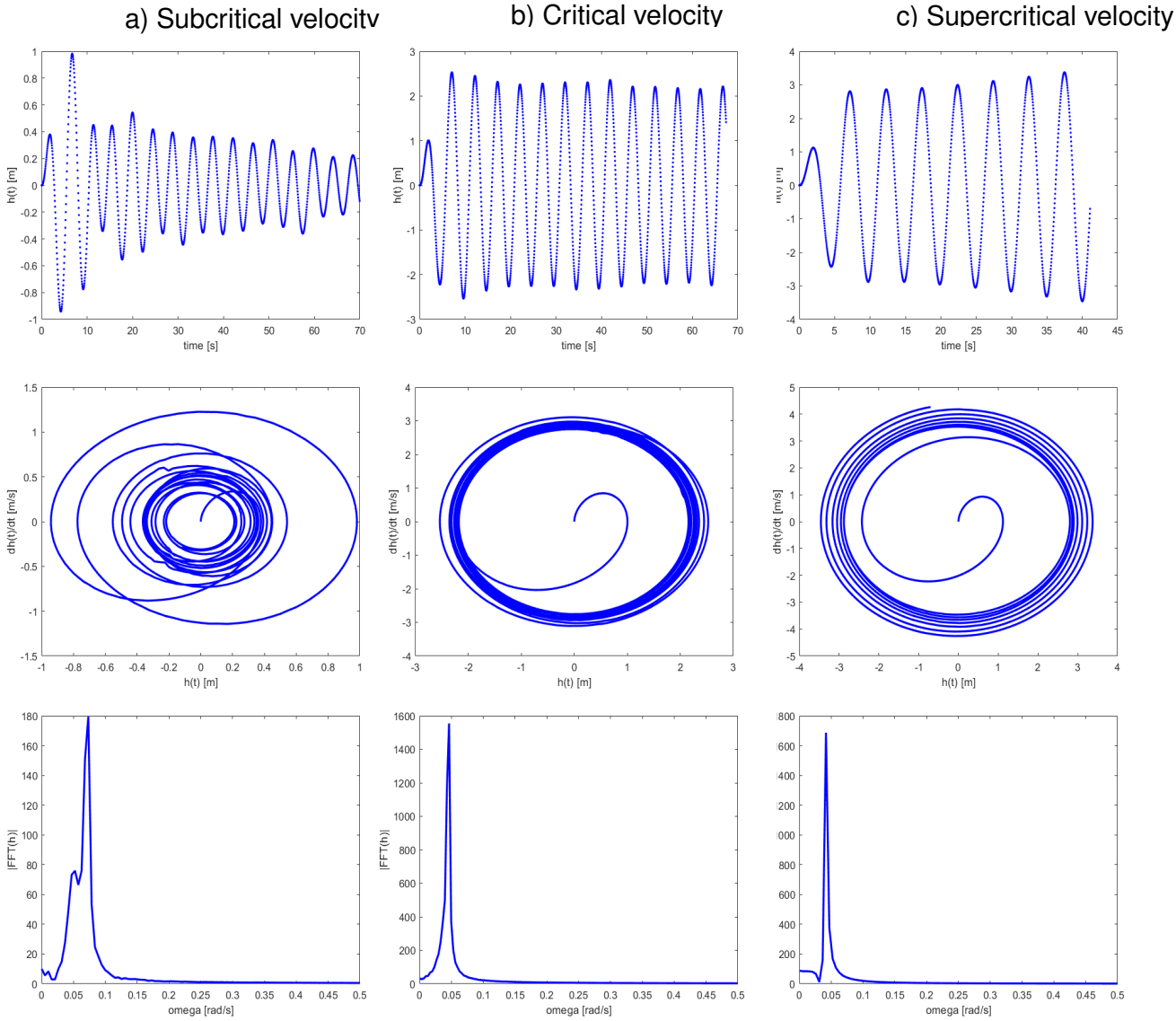

Figure 5.15: Plunging motion - subcritical, critical and supercritical velocities.

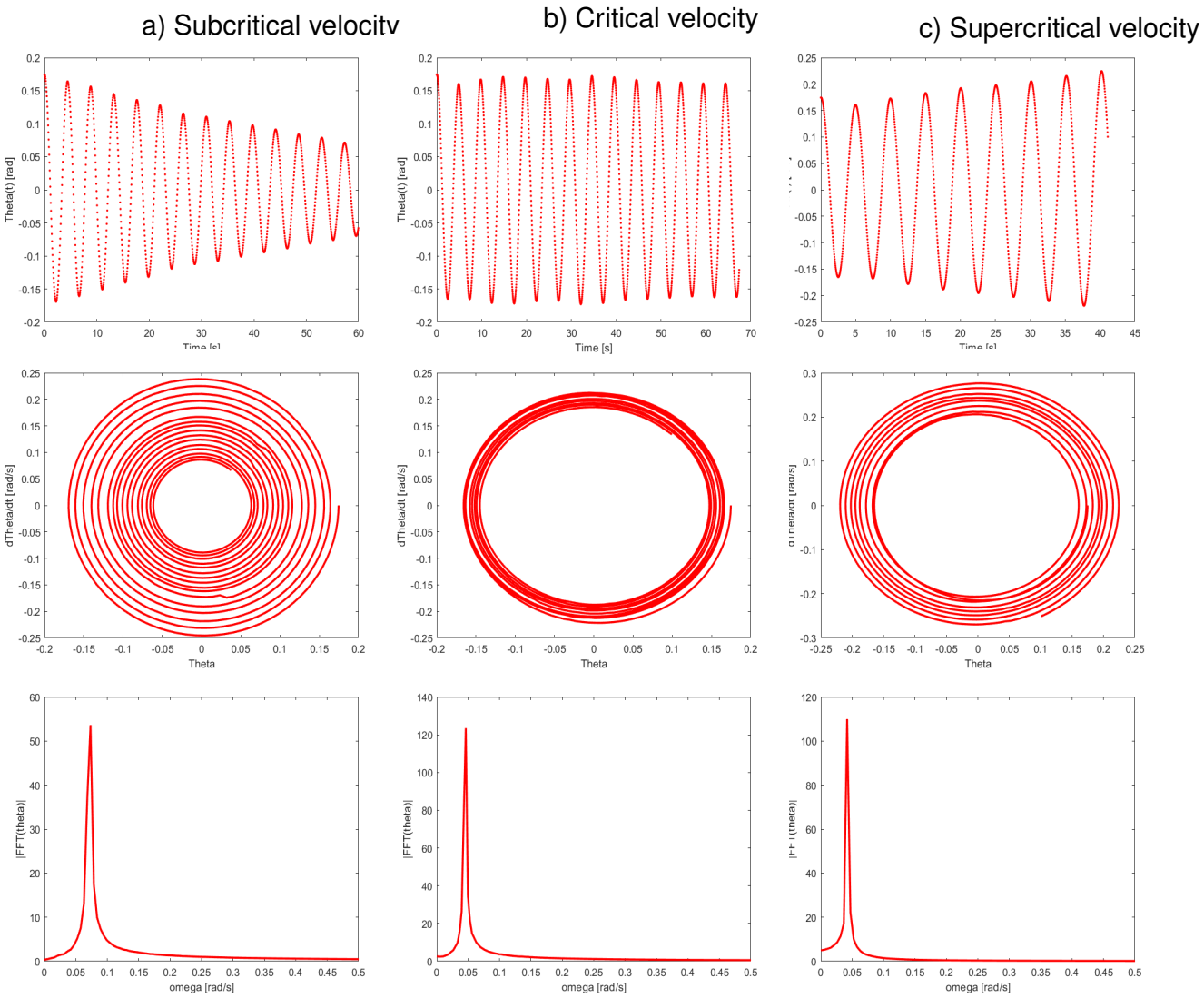

Figure 5.16: Pitching motion - subcritical, critical and supercritical velocities.

Figure 5.15 shows the plots of the plunging motion for the subcritcal, critical and supercritical cases. The first row shows the vertical displacement plotted against time, the second row shows the plunging velocity plotted against the vertical displacement and the last row is the fast Fourier transform of the vertical displacement. The damping is assumed to be purely aerodynamic, since it is considered that the structure does not have any means of dissipation. Figure 5.16 shows the plots of the pitching motion for the subcritical, critical and supercritical cases. Similarly to figure 5.15, the first row shows the pitching angle plotted against time, the second row shows the pitching velocity plotted against the pitching angle and the third row shows the fast Fourier transform of the pitching angle. One can see a limit cycle oscillation is developed in row column b) for both the pitching and plunging motion at the critical velocity. The frequencies collapses into one frequency as well in the frequency plot for the plunging and pitching motion at this velocity.

Although the aerodynamic model is inviscid and the structure is modeled without considering any type of dissipation, there is an aerodynamic damping, which is evident when the energy is transferred between the fluid and the structure. At speeds below the critical velocity, the free wind stream represents an energy sink, since energy is transferred from the structure towards the flow. On the other hand, at speeds higher than the critical velocity, the free stream wind represents a source, since the energy is transferred from the flow to the structure. As the wind speed approaches the critical velocity, the frequencies collapses into one for the plunging and pitching case. This phenomena is known by the name of flutter (*Fung* (1955)).

The critical was set to $u_{critical} = 51.1728m/s$, which is about 4.28% higher than the critical velocity for the suspension bridge given in *Fung* (1955). This error can be explained due to the discretization of the suspension bridge, where a total of 10 panels were used. A low panel discretization is used to reduce the computational cost for the simulation. To investigate this further, the suspension bridge is discretized using 20 and 30 panels to see if this affects the critical velocity speed of the system. As one can see in figure 5.17, the critical velocity for the suspension bridge when the number of panels has increased to 20 is $u_{critical} = 50.2728m/s$, which is about 2.44% higher than the experimental one given in *Fung* (1955). Similarly, the critical velocity for the suspension bridge when the number of panels is 30 is $u_{critical} = 50.1778m/s$, which is 2.28% higher than the experimental one in *Fung* (1955). Clearly the accuracy is increasing as the number of panels increases. However due to the long computation time it takes for the 20 and 30 panel discretizations, the 10 panel discretization is used for the aeroelastic system.

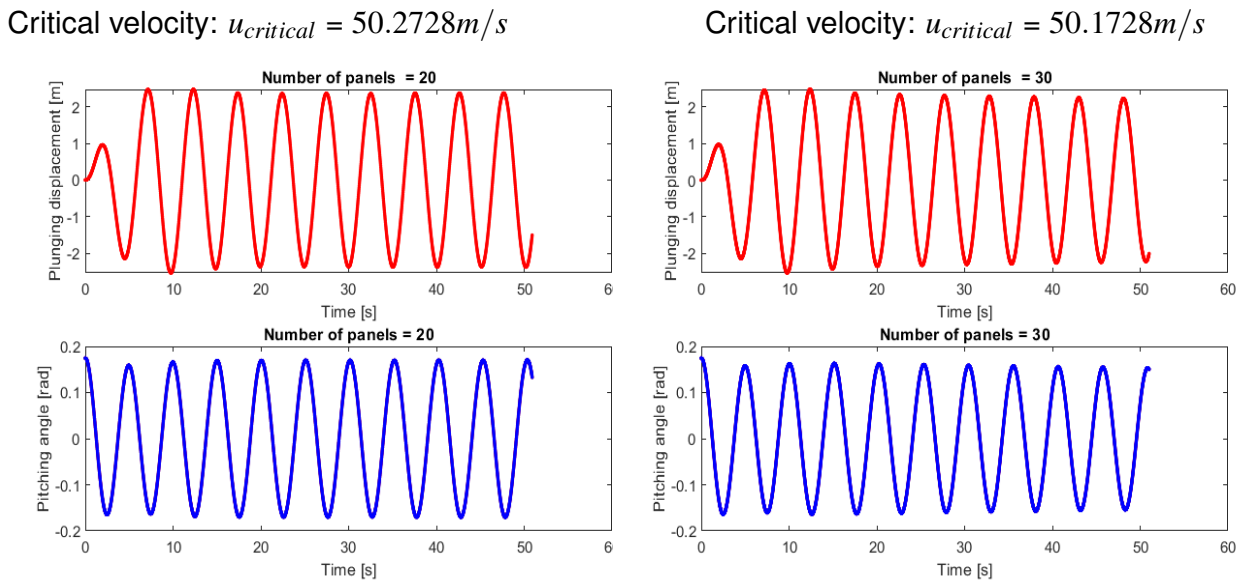

Figure 5.17: Critical velocities for the 20 and 30 number of panels discretization of the suspension bridge

The velocities have bee converted from imperial units to metric units. The subcritical veloity is given by $u_{subritical} = 119.970472ft/s = 36.576m/s$, critical velocity $u_{critical} = 161ft/s = 49.0728m/s$ and the supercritical $u_{supercritical} = 175ft/s = 53.34m/s$. The initial vertical displacement for the simulation was $h(0) = 0m$ and the initial pitch angle was set to $\theta(0) = \frac{10\pi}{180}$. The pitching velocity and plunging velocity were all set to zero at the beginning, $\frac{dh(0)}{dt} = 0$ and $\frac{d\theta(0)}{dt} = 0$.

5.3 Numerical experiments

In this part of the thesis, the results regarding the use of different wake scheme implementations for the aerodynamic model are presented. The wake shape are computed for the pitching, plunging and steady state using the different wake schemes to investigate the relationship between the methods. In addition to this, the use of a nonlinear spring instead of a linear spring in the aeroelastic model is investigated further. A trained neural network for both the linear and nonlinear spring in the structural model is used and compared against the analytical springs, to see if one can use neural networks as a means of replacing terms in the equations of motion of the structural model for the airfoil.

5.3.1 Aerodynamic model

CL vs angles of attack - for different wake schemes

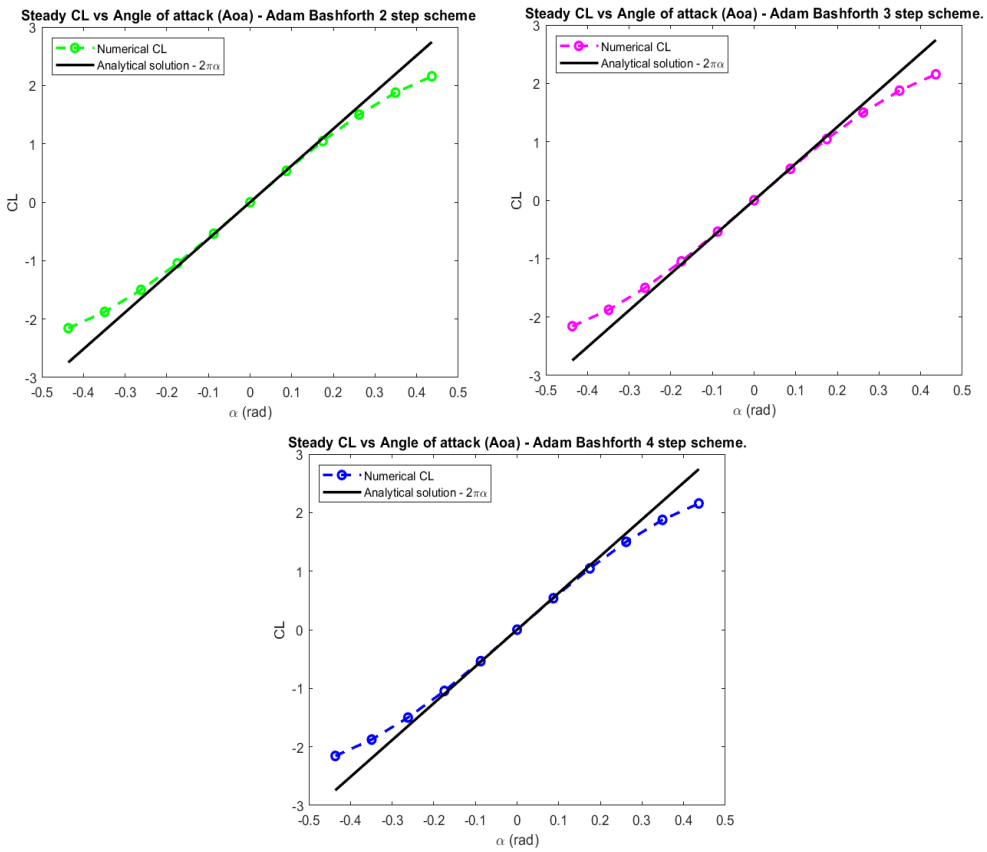

Figure 5.18: CL vs angles of attack - for the different wake schemes.

In figure 5.18, the lift coefficient for the steady state is calculated for different values of the angle of attack. The top left figure shows the lift coefficient using the Adam Bashforth 2 step method, the figure in the top right uses the Adam Bashforth 3 step method and the figure in the bottom uses the Adam Bashforth 4 step method. All the methods uses the explicit Euler method in the starting phase. The results shows that the different wake schemes seems to fit the analytical solution well when the angle of attack is small. As the angle of attack increases, the difference between the numerical lift coefficient and the analytical one increases. The parameters used for the simulation using the different wake schemes are the same given in table 5.7.

Wake shape comparison

Parameter	Symbol	Value	Unit
chord	c	0.25	m
Air mass density	ρ	1.255	kg/m^3
Cutoff vortex radius	$Cutoff_{rad}$	0.0001	m
Horizontal free stream velocity	u_∞	1	m/s
Angle of attack	α	$\frac{10\pi}{180}$	
Number of iterations	I	500	
Number of panels	N_{panels}	10	
Time increment	Δt	0.025	s

Table 5.7: Simulation parameters - wake.

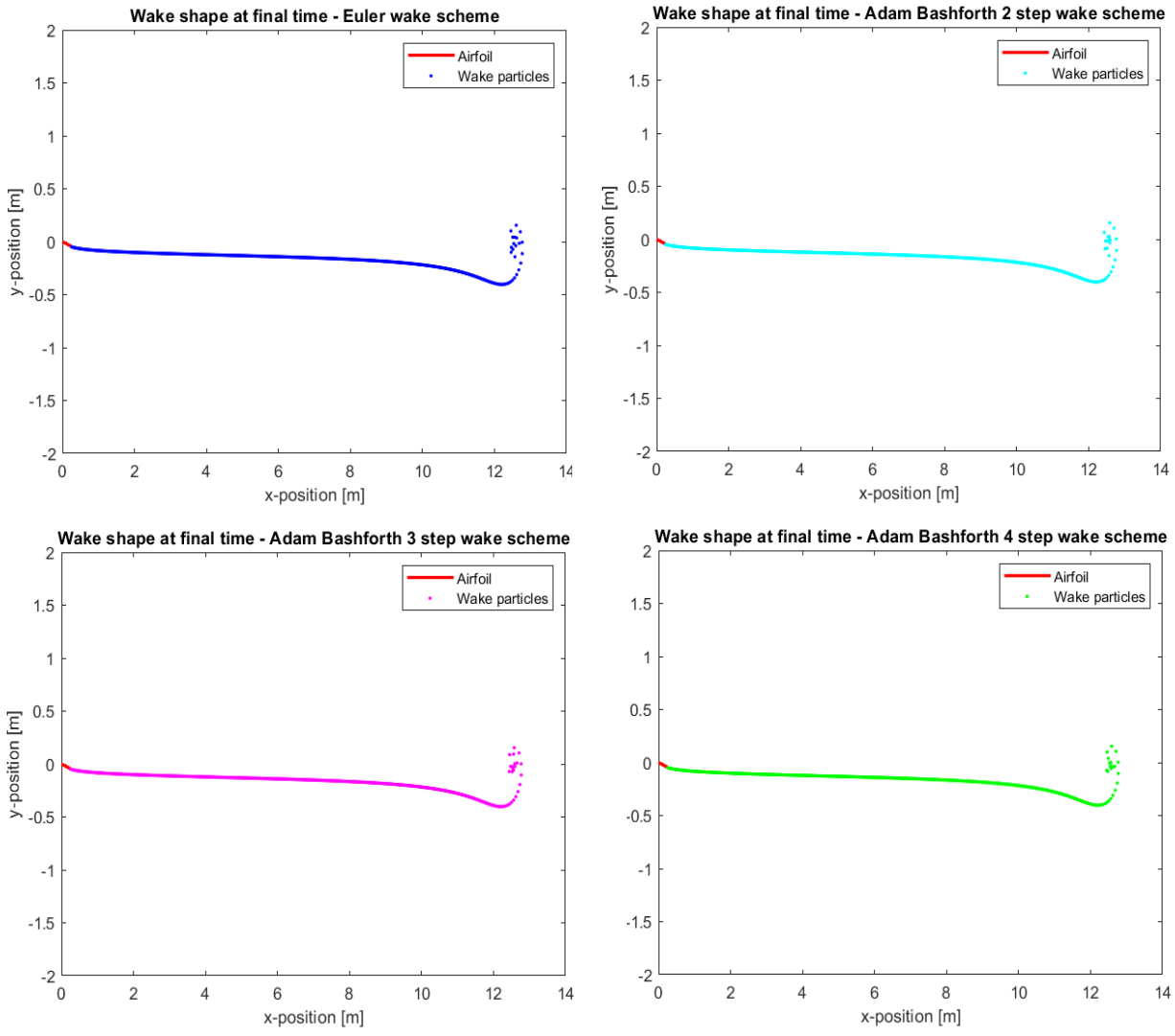

Figure 5.19: Comparison between the wake shapes for the different schemes.

Figure 5.18 shows the wake shape at the final time step for the simulation, using the different wake schemes for convecting the wake particles. The parameters used for this simulation are given in table 5.7. As one can observe from the figure, the shape of the wake is similar for all the cases, except at the near end of the wake. The motion is assumed to be a steady horizontal wind speed with constant velocity, where $u_\infty = 1m/s$. At the end of the airfoil, the shape of the

wake particles differ between the different methods. Since the wake particles start to cluster near the end, the total induced velocity for each wake particle by the other wake particles increases and thus the wake shape near the end varies between the methods.

Hausdorff distance

In order to further investigate the wake shape using different wake schemes, the wake shapes for the different wake schemes are compared using the Hausdorff distance. The Hausdorff distance metric is a method commonly used in computer vision and image processing tasks for computing the distance between two sets of points, and is regarded as an ideal choice to compute relatively more accurate distances between objects (*Maiseli* (2020)). Compared to other metric methods, the Hausdorff distance takes into account critical details of the objects position and shape to output reasonable distances between pairs of point sets. Still, one disadvantage of using this method is the highly sensitivity to outliers and noisy data.

Definition of Hausdorff distance

Given two sets of points A and B, where $a \in A$ and $b \in B$, the directed forward and backward Hausdorff distances between the two sets are defined as

$$h(A,B) = max_{a\in A}(min_{b\in B}(f(a,b))), \tag{5.15}$$

$$h(B,A) = max_{b\in B}(min_{a\in A}(f(b,a))), \tag{5.16}$$

then, the Hausdorff distance is the maximum of these two values

$$H(A,B) = max(h(A,B),h(B,A)). \tag{5.17}$$

The function $f(x,y)$ is an arbitrary distance function, such as the second norm or the infinity norm. In this case, the infinity norm and the second norm are used.

Property name	Symbol	Value	Unit
Horizontal free stream velocity	u_{inf}	1	m/s
Air mass density	ρ	1.255	kg/m^2
Number of panels	N_{panels}	10	
Angle of attack	α	$\frac{10\pi}{180}$	
Chord	c	0.25	m
Time increment	Δt	0.025	s
Time steps	I	200	

Table 5.8: Setup for the HDD comparison for the airfoil

Three cases are considered when comparing the wake shape using the different wake convection schemes, which are the steady case, plunging case and the pitching case. For each of the figures shown below, the title above the figure denotes which wake scheme that is used for comparing with the rest. This is why for some of the subplots, the Hausdorff distance is zero for all the time step, which is the case when the method is compared with itself.

Steady motion - second norm

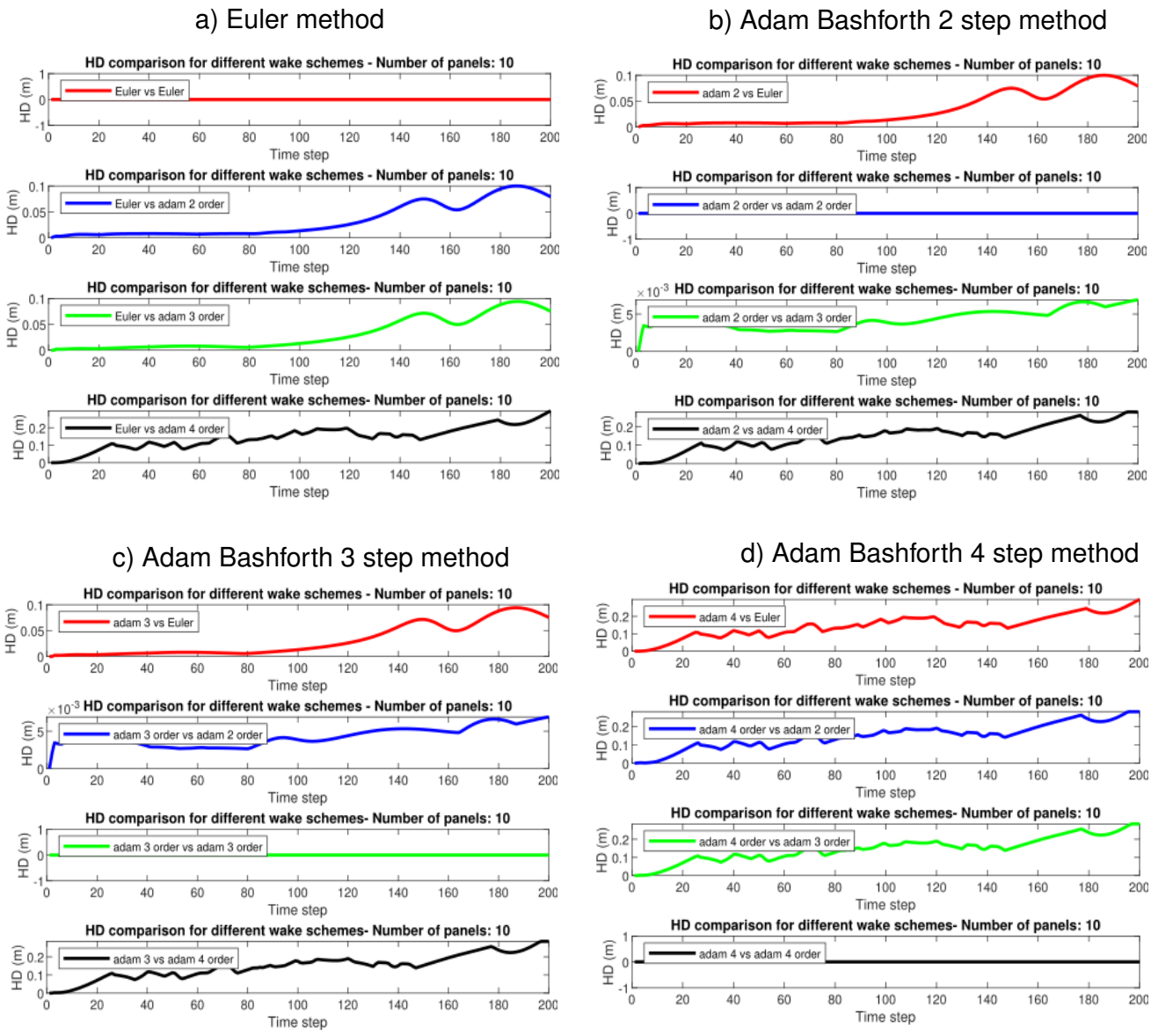

Figure 5.20: Evolution of the HDD for the wake vortices steady flow - comparison between the wake schemes, with the euclidean norm as the distance function.

Steady motion - infinity norm

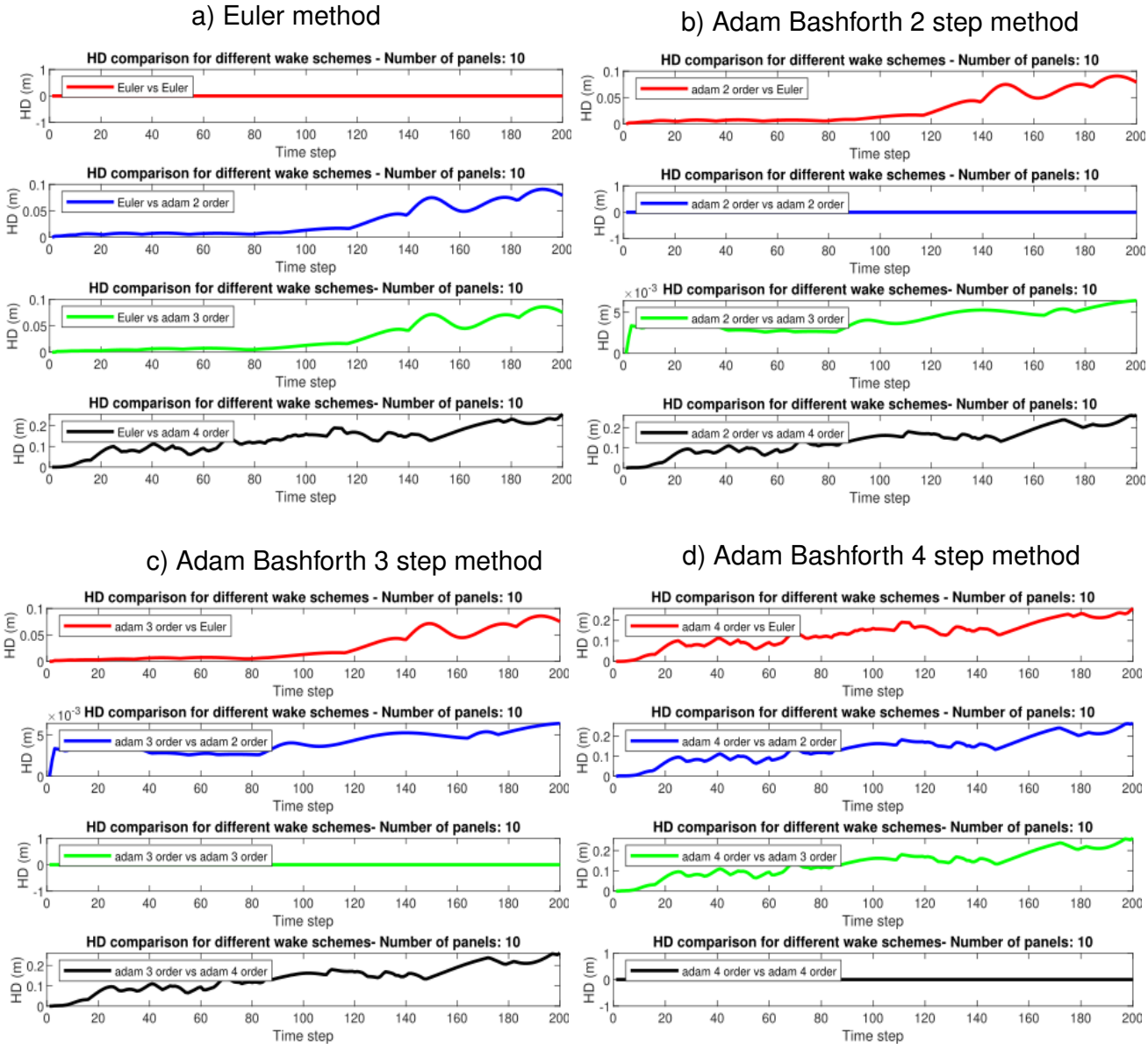

Figure 5.21: Evolution of the HDD for the wake vortices steady flow - comparison between the wake schemes, with the infinity norm as distance function.

From the observation of figure 5.20 and 5.21, one can see that for all the cases the HDD increases as the simulation time increases. The largest value in the HDD is found when between the Adam Bashforth 4 step method and the other methods, where one can observe a HDD of $0.2m$ near the end of the simulation. The smallest difference in the HDD was found between the Adam Bashforth 3 step method and the Adam Bashforth 2 step method, where the maximum HDD was at $0.005m$ at the end of the simulation.

Pitching motion - second norm

The pitching motion added to the arifoil is given as: $\alpha(t) = \alpha_{max}\sin(\omega t)$. Here, $\omega = 0.2s^{-1}$ and $\alpha_{max} = \frac{\pi}{180}$.

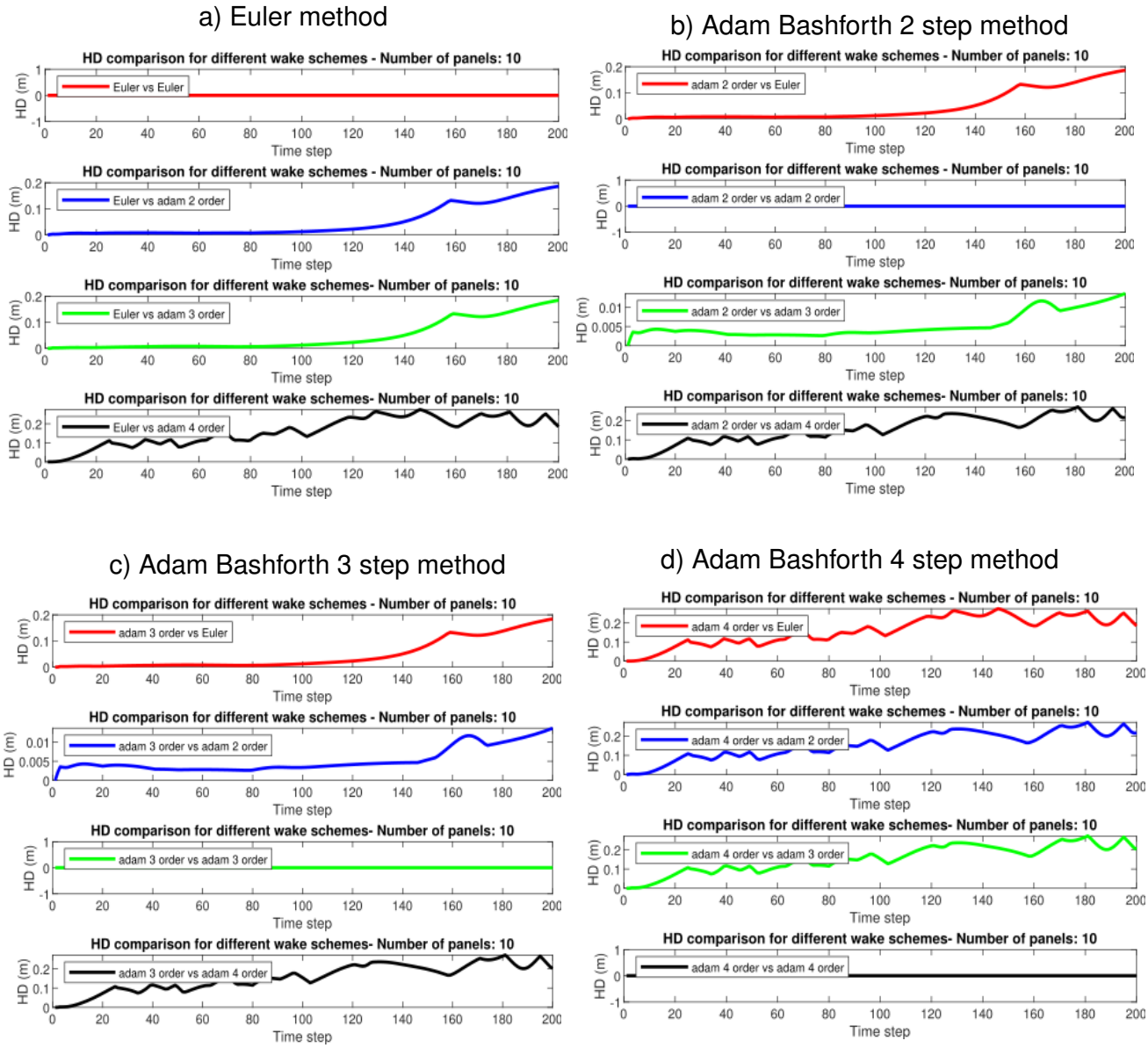

Figure 5.22: Evolution of the HDD for the wake vortices, added pitching motion - comparison between the wake schemes, with the euclidean norm as distance function.

Pitching motion - infinity norm

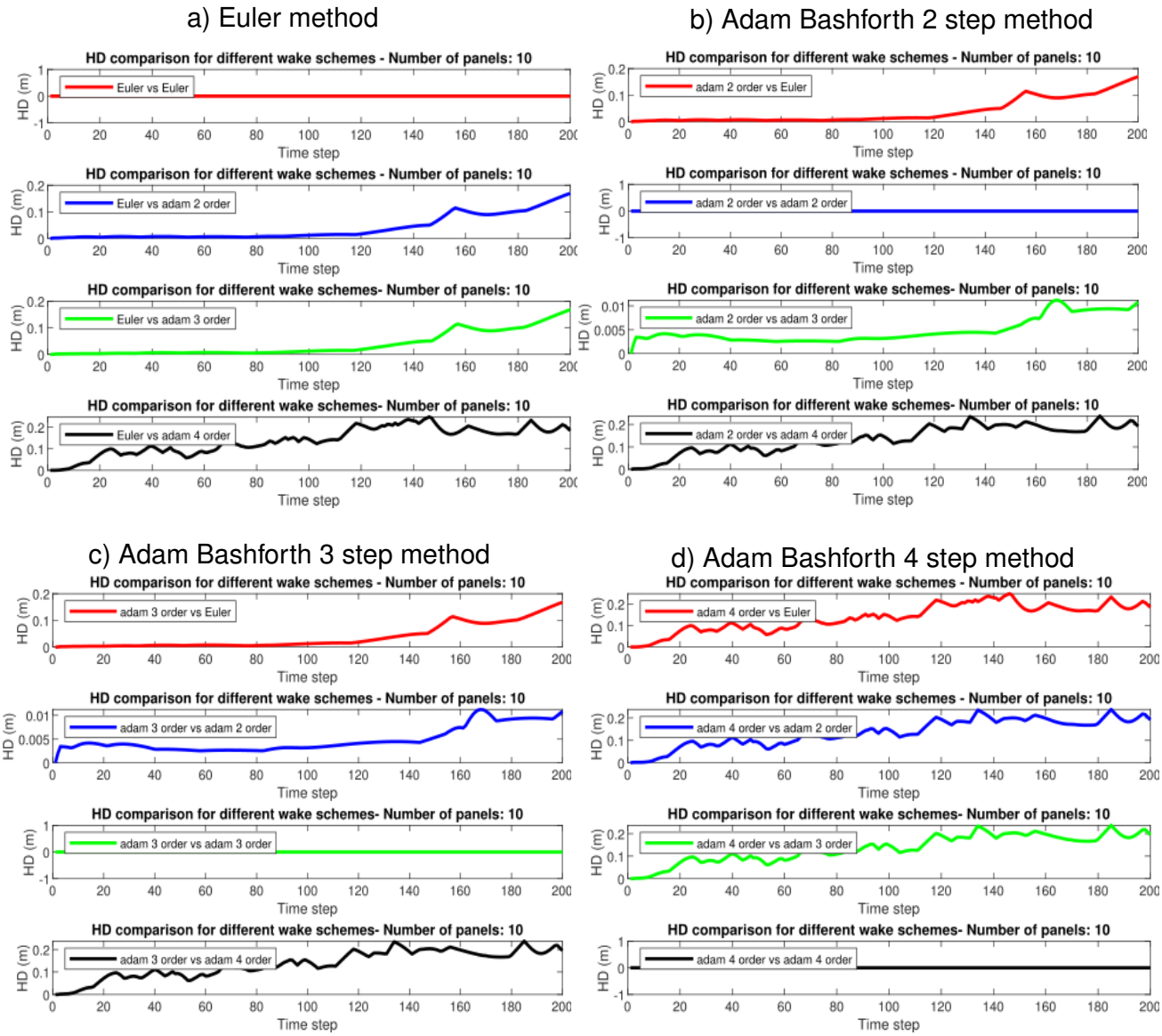

Figure 5.23: Evolution of the HDD for the wake vortices, added pitching motion - comparison between the wake schemes, with the infinity norm as distance function.

Figure 5.23 and 5.22 shows the HDD for the wake at each time step using the different wake schemes for the pitching airfoil. The HDD for both the infinity and second norm increases as the number of time steps increases. The biggest value of the HDD appears between the Adam Bashforth 4 step method and the other methods, where one can observe a HDD of about $0.2m$, when both using the infinity norm and the second norm. The smallest difference in the HDD is between the Adam Bashforth 3 step method and the Adam Bashforth 2 step method, where the HDD is $0.01m$ at the end. The HDD between the explicit Euler and the Adam Bashforth 2 and 3 step methods seem to be increase near the end, where the HDD reaches a value of $0.2m$.

Plunging motion - second norm

The added plunging motion of the airfoil is given as the function: $h(t) = -h_0 \cos(\omega t)$, where $\omega = 0.2s^{-1}$ and $h_0 = 0.1m$.

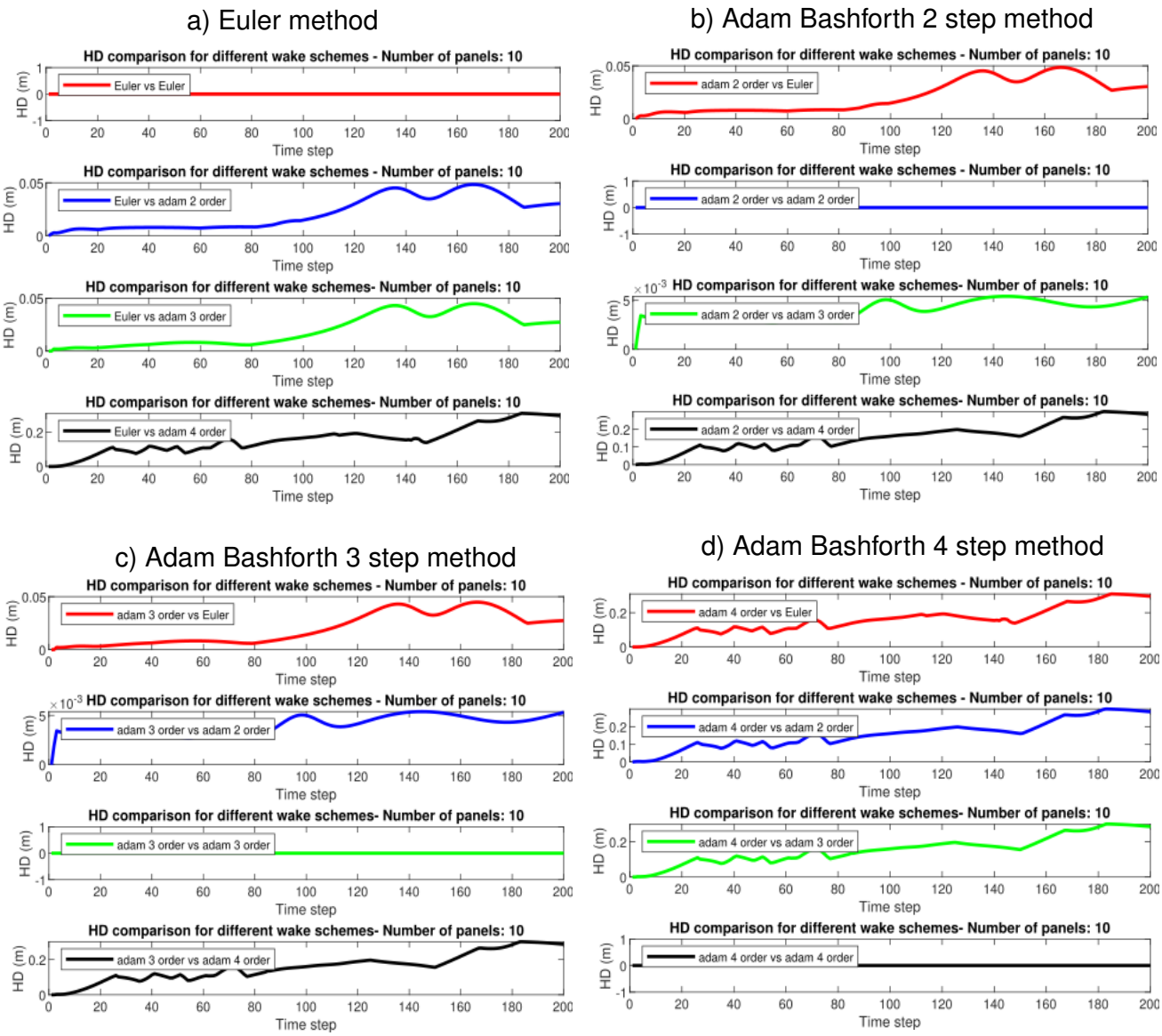

Figure 5.24: Evolution of the HDD for the wake vortices, added plunging motion - comparison between the wake schemes, with the euclidean norm as distance function.

Plunging motion - infinity norm

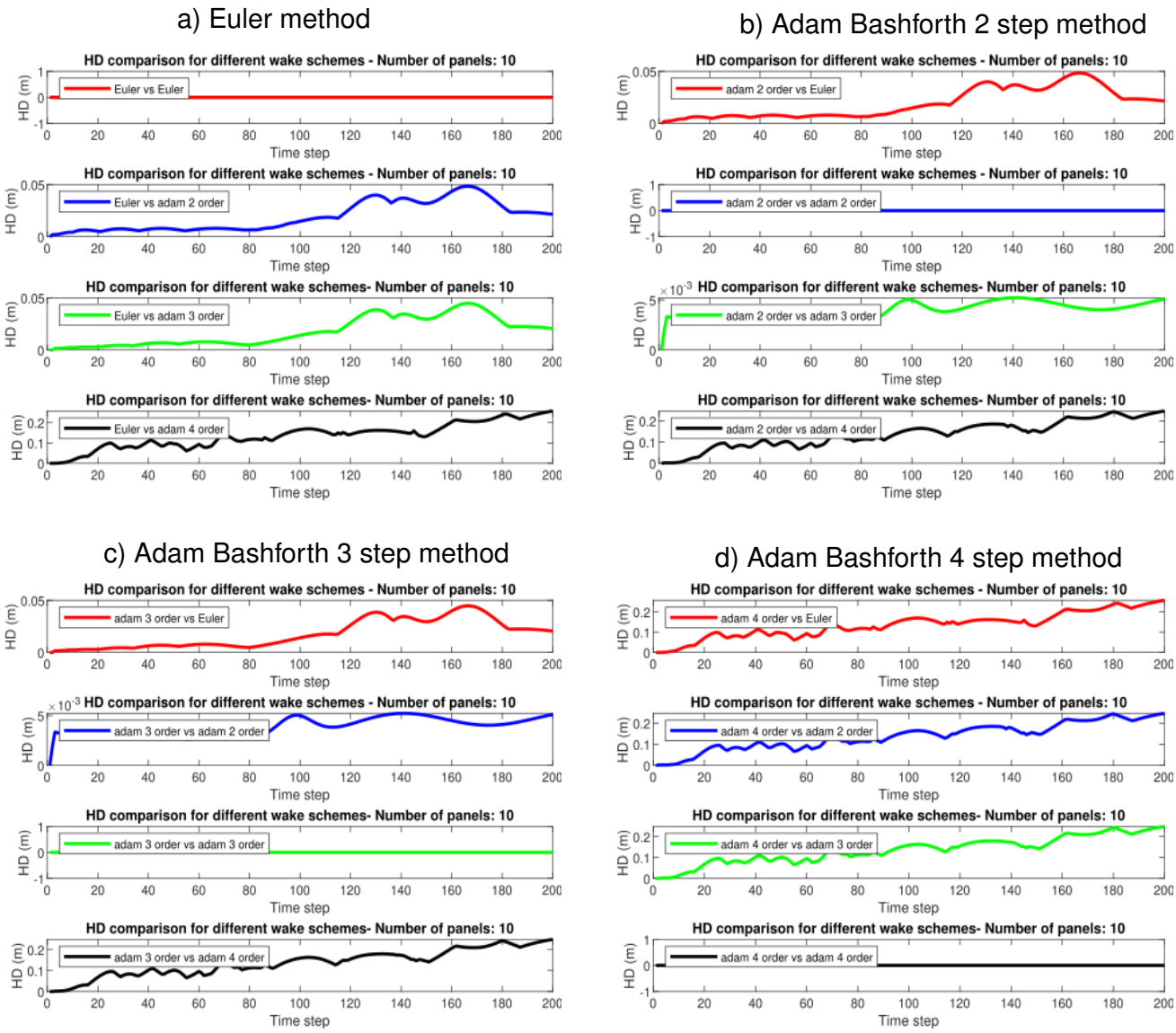

Figure 5.25: Evolution of the HDD for the wake vortices, added plunging motion - comparison between the wake schemes, with the infinity norm as distance function.

Figure 5.24 and 5.25 shows the HDD for the wake at each time step using the different wake schemes for the plunging airfoil. Again, the HDD increases as the number of time steps increases, except for the Euler method, where the HDD seems to decrease near the end when the infinity norm is used. The HDD is biggest when comparing the Adam Bashforth 4 step method to the other methods, where the HDD reaches a value of $0.2m$, and the lowest HDD is between the Adam Bashforth 3 step method and the Adam Bashforth 2 step method, where the HDD reaches a maximum of $0.005m$.

For all the results regarding the HDD in figures 5.20, 5.21, 5.22, 5.23, 5.24 and 5.25, the HDD increases near the end of the simulation for all the cases. One explanation of this may be due to the formation of a wake rollup near the end of the wake. In the beginning, wake particles are convected with the free stream velocity and the number of wake particles are low. As time increases, the number of wake particles will increase and the wake particles start clustering near the end. Near the end of the wake, the wake particles will be relatively close to each other and thus their influence on each other will be greater.

5.3.2 Aeroelastic model

In this part, the result regarding the use of a nonlinear spring is presented. The data used for the airfoil are the same as for the data in the numerical verification of the suspension bridge case with the linear spring. Furthermore, a neural network is used for generating the linear and nonlinear spring terms for the structural model. This is then replaced with the linear and nonlinear analytical spring terms for the aeroelastic model to see if one can use a neural network trained spring term as a replacement in the aeroelastic model of the airfoil.

Nonlinear spring

Here the nonlinear spring for the structural model is used in the aeroelastic model. There's no additional damping, thus $d_h = 0$ for the plunge and $d_\theta = 0$ for the pitch. Since the mass distribution of the airfoil is uniform, the chordwise offset $x_\theta = 0$. the nonlinear spring constants are set to $k_{nonlin-h} = 0.01k_h$ and $k_{nonlin-\theta} = 0.01k_\theta$ and the equations of motion becomes

$$\begin{bmatrix} m & 0 \\ 0 & I_P \end{bmatrix} \begin{bmatrix} \ddot{h} \\ \ddot{\theta} \end{bmatrix} + \begin{bmatrix} k_h & 0 \\ 0 & k_\theta \end{bmatrix} \begin{bmatrix} h \\ \theta \end{bmatrix} + \begin{bmatrix} k_{nonlin-h} & 0 \\ 0 & k_{nonlin-\theta} \end{bmatrix} \begin{bmatrix} h^3 \\ \theta^3 \end{bmatrix} = \begin{bmatrix} F_L \\ M \end{bmatrix}. \tag{5.18}$$

The data used for this simulation is the same as for the modelling of the suspension bridge in the numerical verification part of the aeroelastic model, given table 5.9 below.

Property name	Symbol	Value	Unit
Number of panels	N_{panels}	10	
Air mass density	ρ	1.255	kg/m^2
Chord	c	18.288	m
Initial angle of attack	α	$\frac{10\pi}{180}$	
Mass per unit span	m	12879.78698	kg/m
Moment of inertia	I_p	6700056.033	$kg \cdot m$
Half chord	b	9.144	m
Chordwise offset	x_θ	0	
Linear torsional stiffness coefficient	k_θ	1614835.039	N/m^2
Nonlinear torsional stiffness coefficient	$k_{nonlin-\theta}$	16148.35039	N/m
Linear plunging stiffness coefficient	k_h	9981.834	N/m
Nonlinear plunging stiffness coefficient	$k_{nonlin-h}$	99.81834	N/m

Table 5.9: Suspension bridge parameters

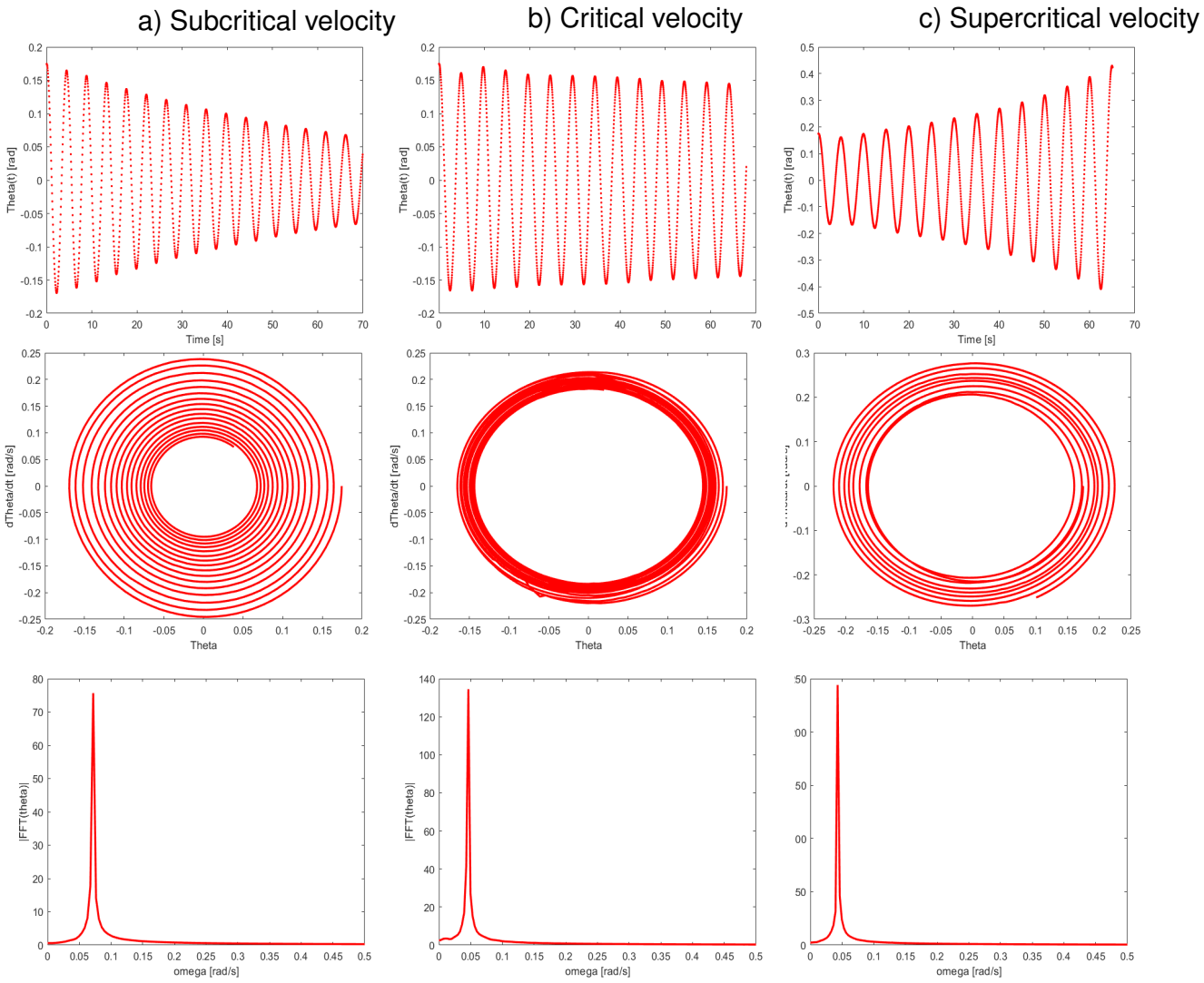

Figure 5.26: Plot of the subcritical, critical and pitching motion for the nonlinear aeroelastic model

Figure 5.26 shows the pitching results of the aeroelastic model using the nonlinear spring. As one can see, the approximation for the subcritical, critical and supercritical velocity of the model is similar to the linear spring model. The subcritical, critical and supercritical velocities for this simulation were the same as for the linear model. Figure 5.27 shows the aeroelastic results for the plunging case, using the nonlinear spring. The flutter speed for the nonlinear model was a bit lower compared to the linear model, where $u_{crit} = 50.6728 m/s$, and the increase and decrease in the plunging height at the supercritical and subcritical velocity differs from the linear aeroelastic model.

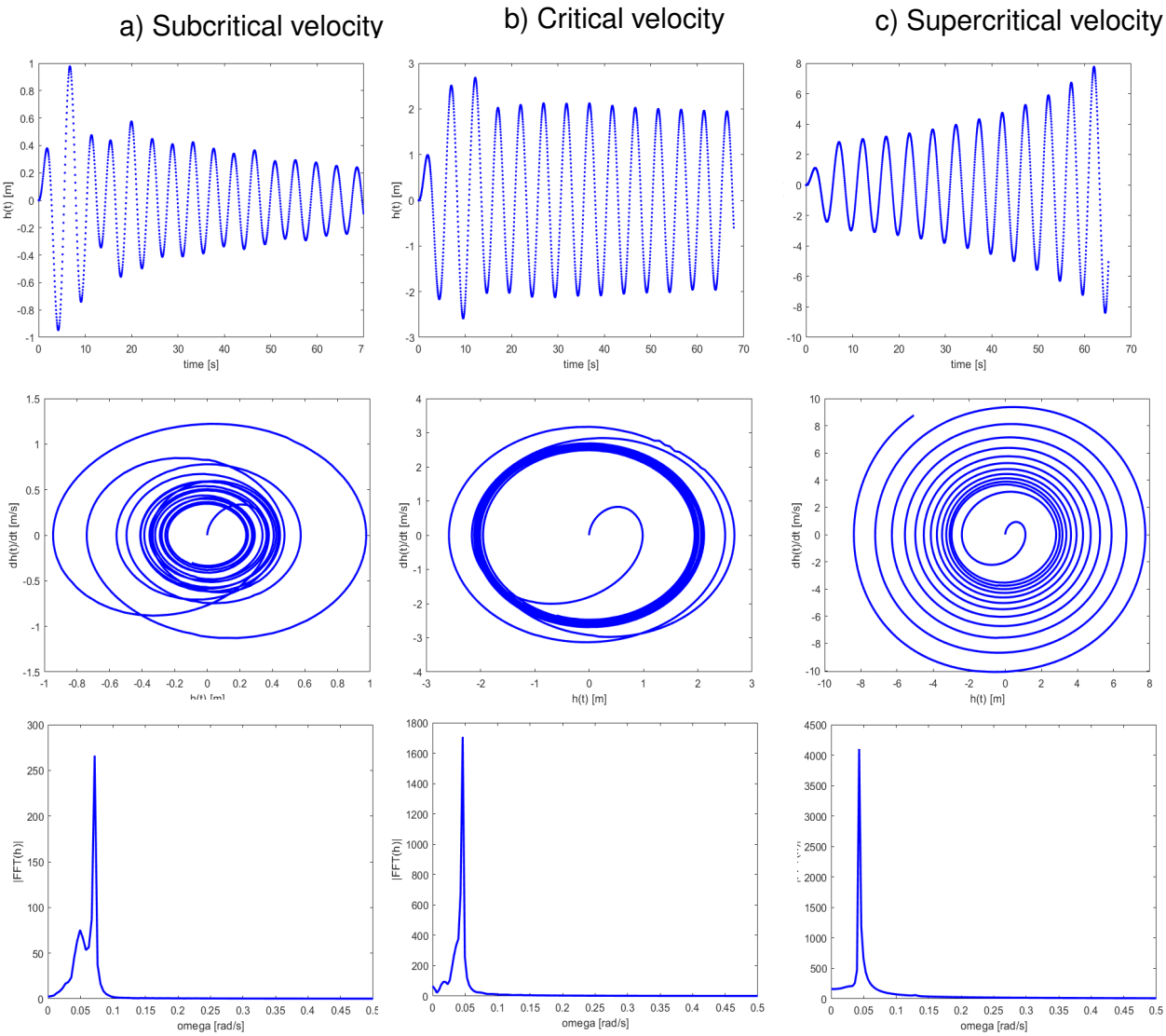

Figure 5.27: Plot of subcritical, critical and supercritical plunging motion for the nonlinear aeroelastic model.

Neural network generated spring

Here, the neural network generated spring terms for the linear and nonlinear model is presented. In this case, only one hidden layer was for the neural network during the training process. The trained neural network spring terms are then replaced with the analytical spring terms in the aeroelastic model, to see if this new function predicts the correct aeroelastic response as the analytical one.

Training parameters

In table 5.10, the data used for training the neural network is shown.

Name	Symbol	Value	Unit
Plunging height range	I_h	$(-30, 30)$	m
Pitching angle range	I_θ	$(-0.75, 0.75)$	
Number of input data points	N_{data}	10000	
Number of neurons in the hidden layer	$N_{neurons}$	10	

Table 5.10: Data - neural network

For all the cases, the data is split into 50 % training, 35 % validation and 15 % testing.

The training input data are presented to the network during the training process. The validation data are used to measure network generalization, and to halt training when generalization stops improving. Finally, the testing data is used to measure the network performance during and after training. These have no effect on training process of the network. The values for the plunging height h and the pitching angle θ are picked randomly from the I_h and I_θ vectors. The algorithm used for the back propagation is the Levenberg-Marquardt. A two-layer feed forward network with sigmoid hidden neurons and linear output neurons is used.

Neural network training - Linear spring

The linear spring terms in the structural model are linear functions given by $F_{linear}(t) = k_h h(t)$ for the plunging spring force and $M_{linear}(t) = k_\theta \theta(t)$ for the pitching moment. The values of the linear constants are $k_h = 1614835.039 Nm^{-1}$ and $k_\theta = 16148.35039 Nm$, which is the same used for the verification of the aeroelastic model.

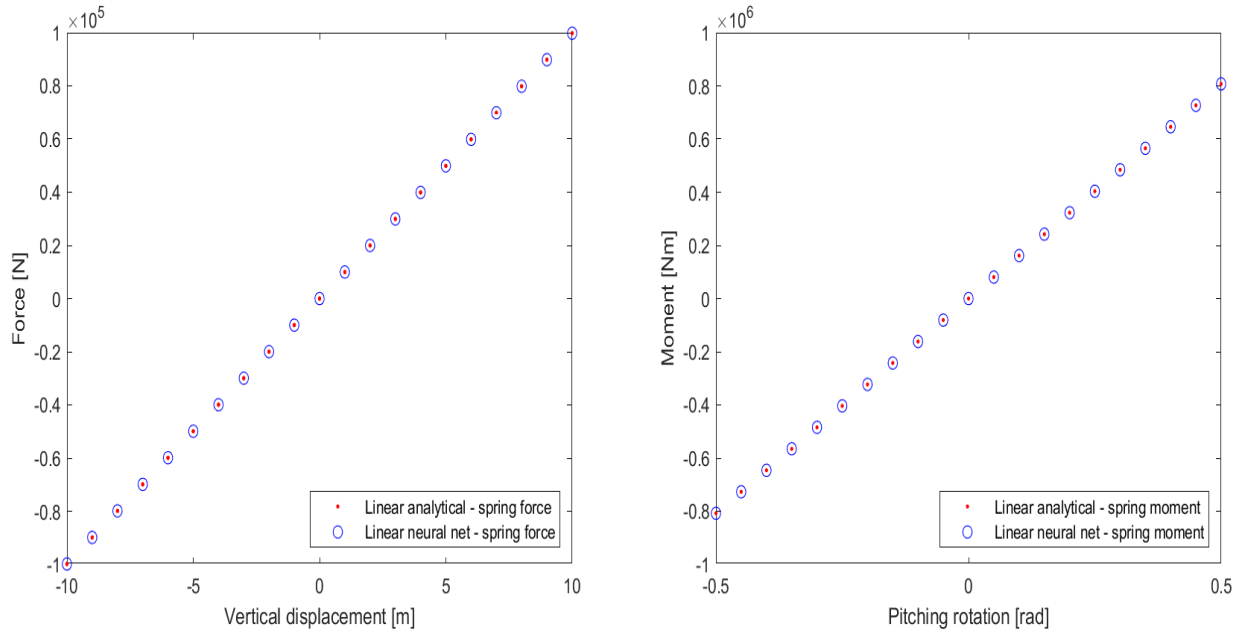

Figure 5.28: Plot showing the linear neural network spring term for pitching and plunging vs the analytical spring term, given some random input data for testing.

In figure 5.29, the trained linear neural network generated spring terms for the plunging and pitching is tested, using some testing data. A total of $N = 21$ data points in the interval $I_\theta = (-0.5, 0.5)$ for the pitching angle and $I_h = (-10m, 10m)$ for the plunging height was chosen in figure 5.29 and 5.28.As one can see, the linear neural network spring for the plunging and pitching fit well to the analytical solution for the pitching and plunging.

Neural network training - nonlinear spring

The expression for the nonlinear spring force and nonlinear spring moment for the plunge and pitching is given by $F_{nonlin}(t) = k_h h(t) + k_{nonlin-h} h(t)^3$ and $M_{nonlin}(t) = k_\theta \theta(t) + k_{nonlin-\theta} \theta(t)^3$. The linear terms have the same value as before and the nonlinear terms are given by $k_{nonlin-h} = 0.01 k_h$ and $k_{nonlin-\theta} = 0.01 k_\theta$.

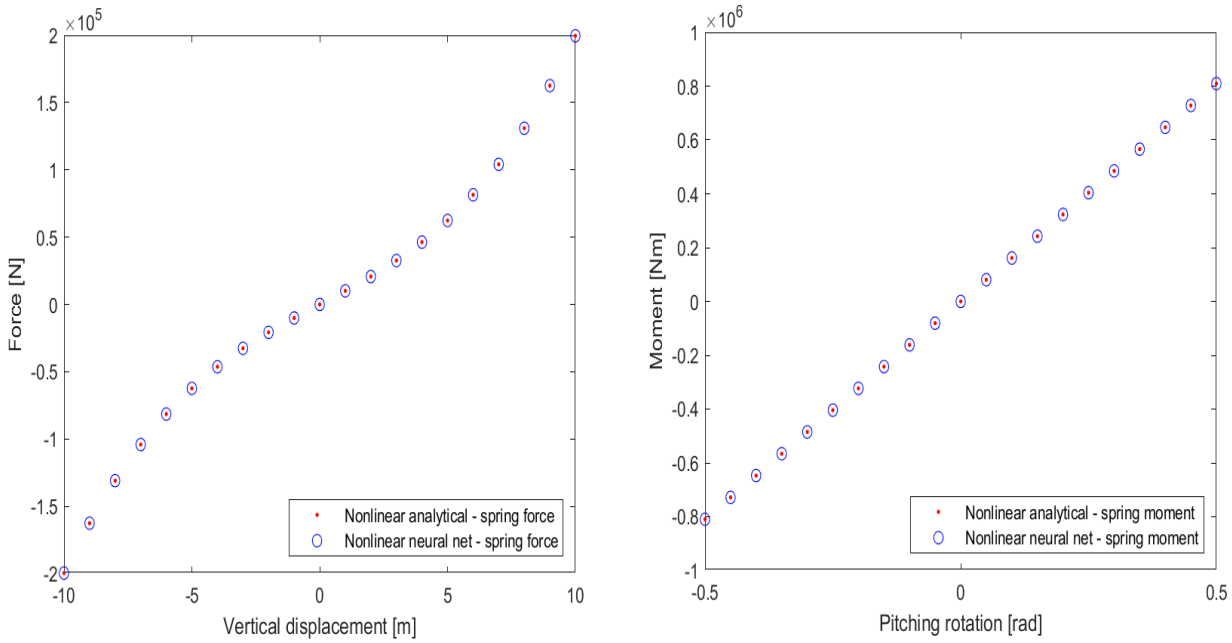

Figure 5.29: Plot showing the nonlinear neural network spring term for pitching and plunging vs the analytical nonlinear spring term, given some random input data for testing.

Figure 5.29 shows the plot of the nonlinear analytical spring force and the nonlinear neural network spring force on the left and the nonlinear analytical spring moment and the nonlinear neural network spring moment on the right. Similar to the linear spring, a total of $N = 21$ data points in the interval $I_\theta = (-0.5, 0.5)$ for the pitching angle and $I_h = (-10m, 10m)$ for the plunging height was chosen in figure 5.29 and 5.28. The neural network seems to fit the analytical solution well when tested against the random input data.

Neural network spring term - comparison

In this part, the neural network generated spring terms are replaced in the aeroelastic model for the linear and nonlinear cases. As one can already observe from figure 5.87 and 5.29, the neural network generated spring terms seems to fit well when given random input data relative to the analytical spring.

Linear spring

Here, the linear neural network spring is used and compared against the analytical spring for the subcritical, critical and supercritical velocities. To see if the initial angle of attack has any effect on the result, three different angles are chosen: $\alpha = \frac{\pi}{180}$, $\alpha = \frac{5\pi}{180}$ and $\alpha = \frac{10\pi}{180}$.

Subcritical velocity - $\alpha = \frac{\pi}{180}$

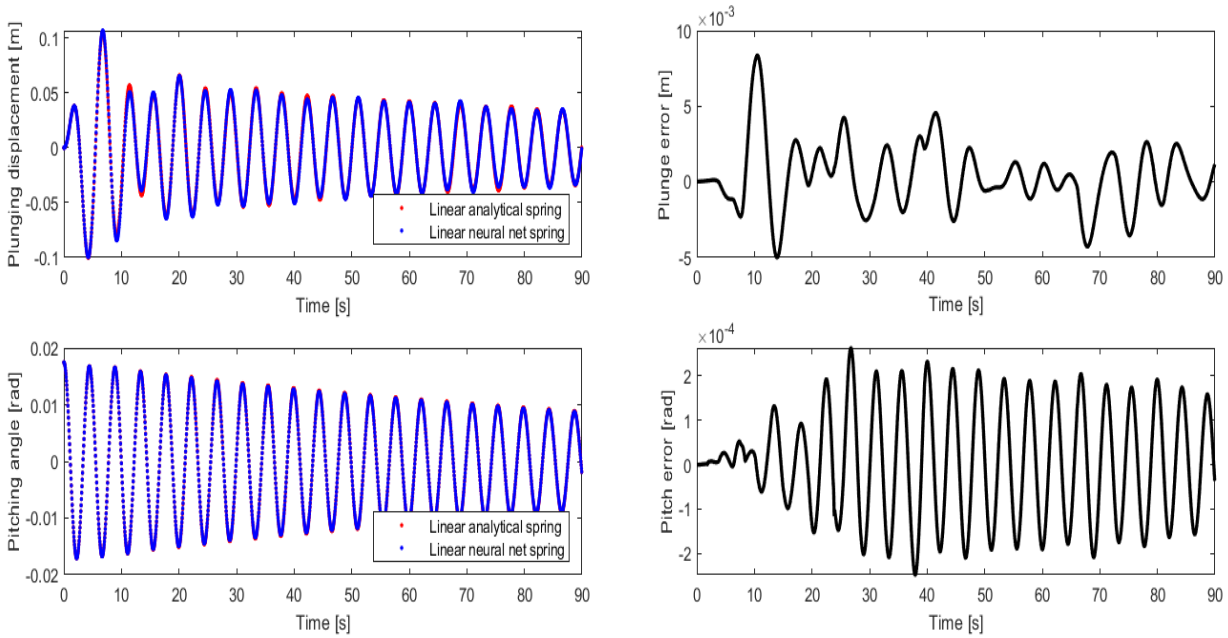

Figure 5.30: Plot of the linear analytical spring vs the linear neural network sspring, pitching and pluniging, $\alpha = \frac{\pi}{180}$

Subcritical velocity - $\alpha = \frac{5\pi}{180}$

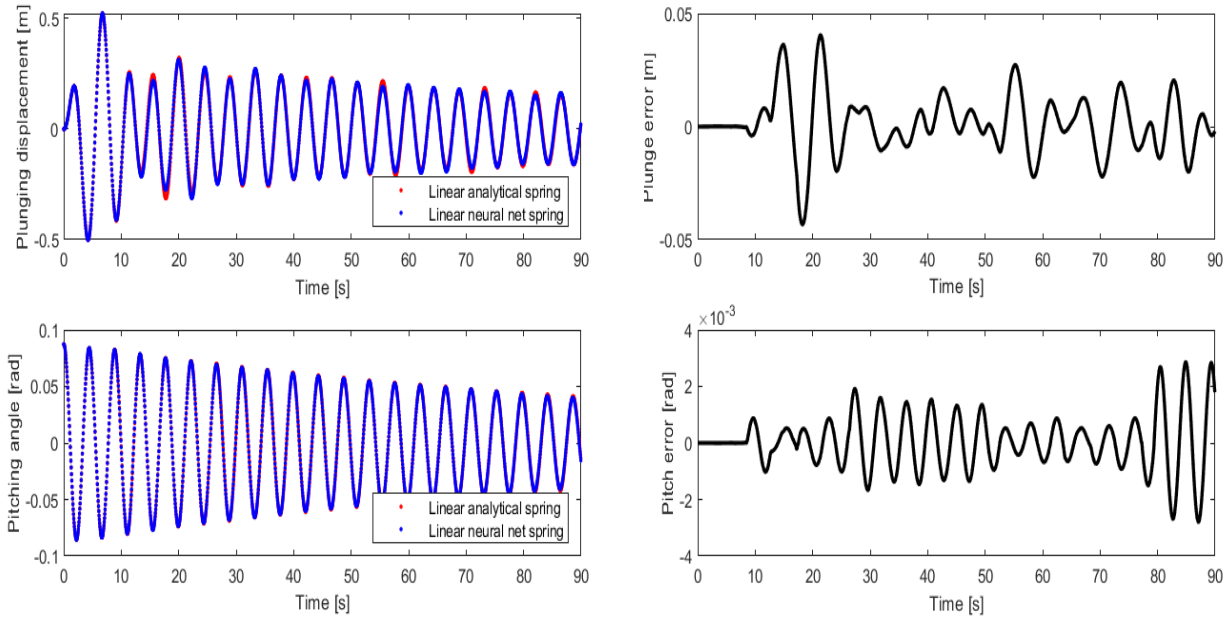

Figure 5.31: Plot of the linear analytical spring vs the linear neural network sspring, pitching and pluniging, $\alpha = \frac{5\pi}{180}$

Subcritical velocity - $\alpha = \frac{10\pi}{180}$

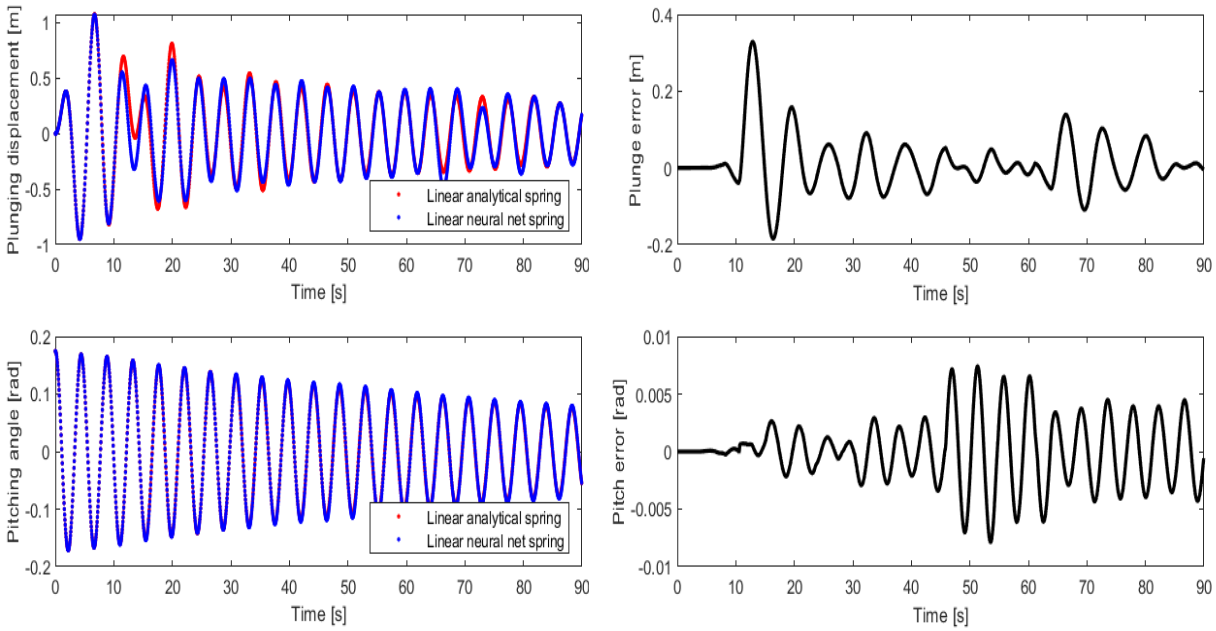

Figure 5.32: Plot of the linear analytical spring vs the linear neural network spring, pitching and plunging, $\alpha = \frac{10\pi}{180}$

Figure 5.30, 5.31 and 5.32 show the plunging height and pitching angle plotted against time in the left column of the figures and the difference between the computed height and rotation angle from the analytical spring and the neural network generated spring. The figure in the top right shows the difference between the plunging height between the analytical and neural network spring as function of time, the figure on the bottom right shows the difference between the pitching angle between the analytical and neural network spring as a function of time. As one can see, the difference between the two solutions seems to increase, as the initial angle of attack increases, for both the pitching and plunging. Still, the neural network seems to fit well according to the aeroelastic model for the subcritical velocity in the linear model.

Critical velocity - $\alpha = \frac{\pi}{180}$

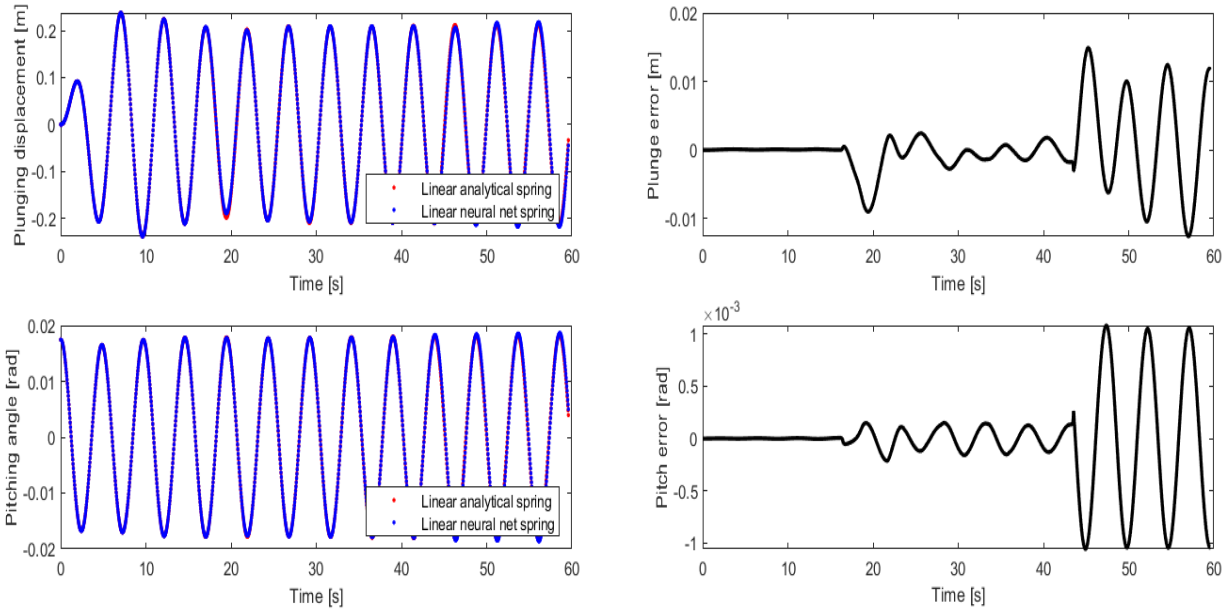

Figure 5.33: Plot of the linear analytical spring vs the linear neural network spring, pitching and plunging, $\alpha = \frac{\pi}{180}$

Critical velocity - $\alpha = \frac{5\pi}{180}$

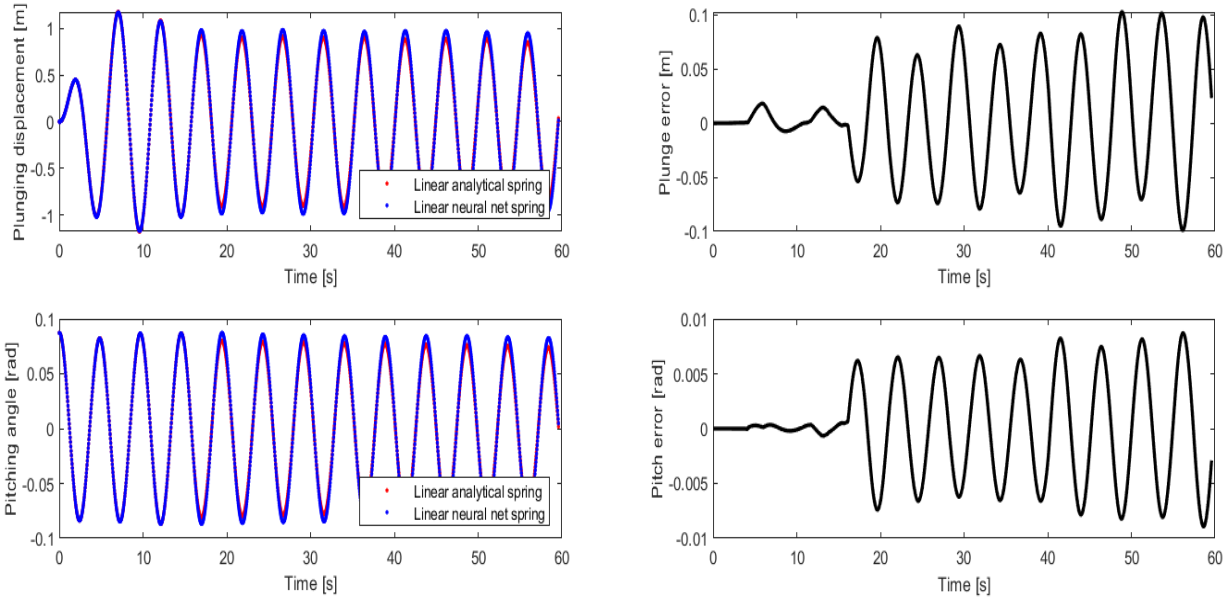

Figure 5.34: Plot of the linear analytical spring vs the linear neural network spring, pitching and plunging, $\alpha = \frac{5\pi}{180}$

Critical velocity - $\alpha = \frac{10\pi}{180}$

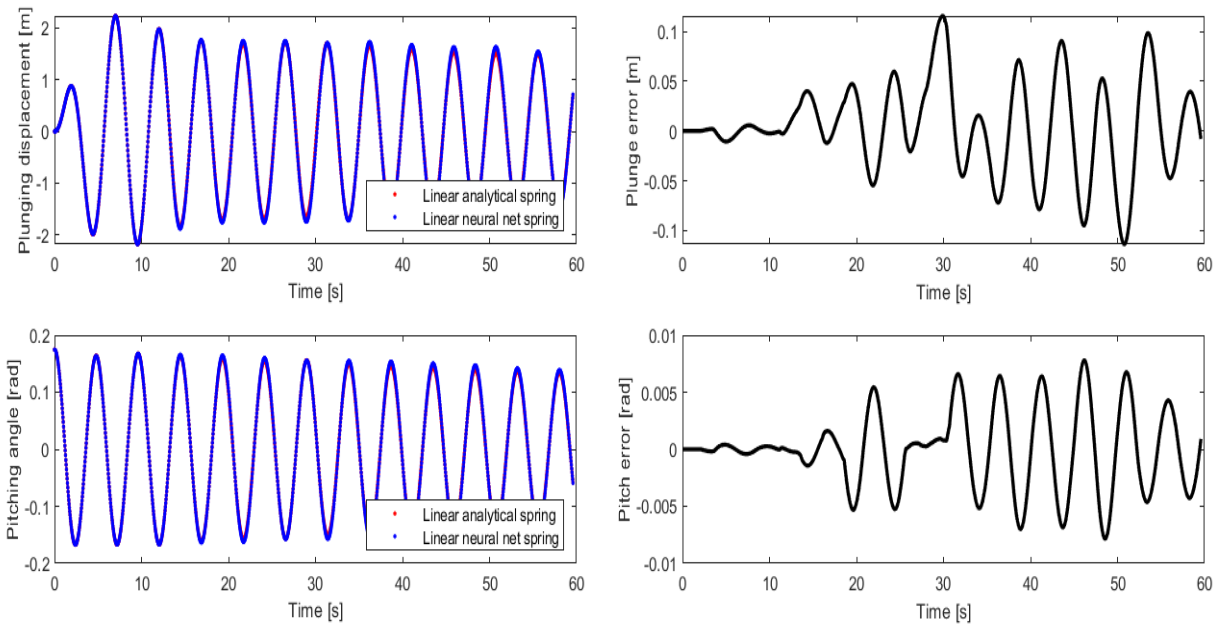

Figure 5.35: Plot of the linear analytical spring vs the linear neural network spring, pitching and plunging, $\alpha = \frac{10\pi}{180}$

In figures 5.33, 5.34 and 5.35 one can see the plunging height and pitching angle plotted against time in the left column of the figures and the difference between the computed height and rotation angle from the analytical spring and the neural network generated spring in the right column at the critical velocity for the different angles of attack. As one can see, the difference between the two solutions seems to increase, as the initial angle of attack increases, for both the pitching and plunging.

Supercritical velocity - $\alpha = \frac{\pi}{180}$

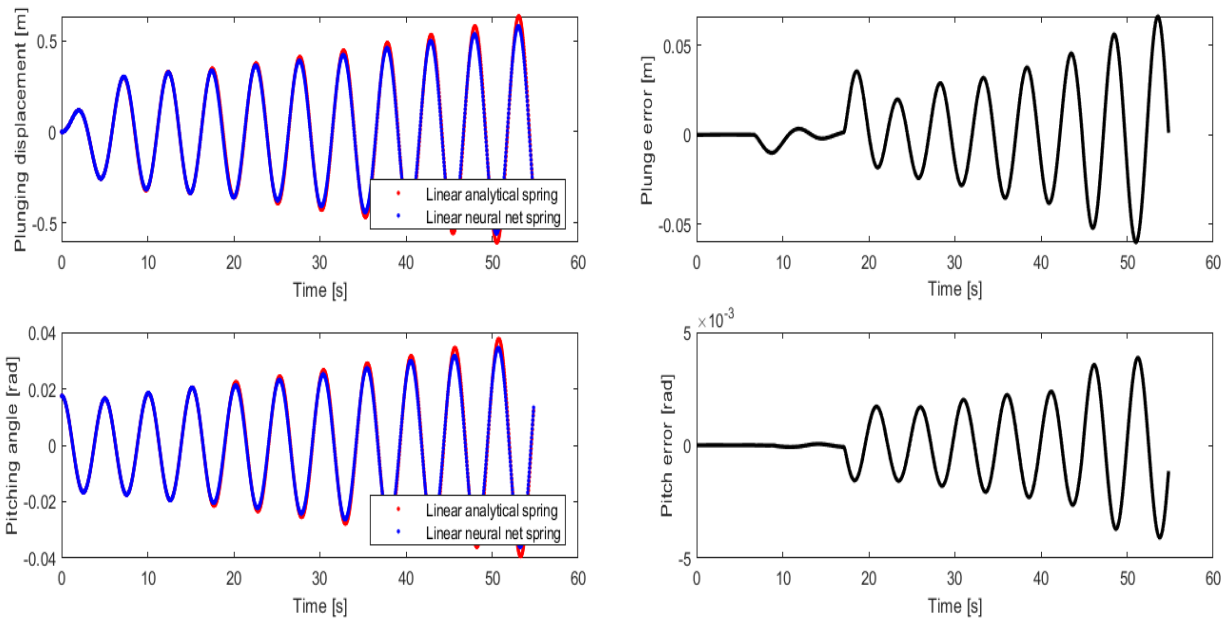

Figure 5.36: Plot of the linear analytical spring vs the linear neural network spring, pitching and plunging, $\alpha = \frac{\pi}{180}$

Supercritical velocity - $\alpha = \frac{5\pi}{180}$

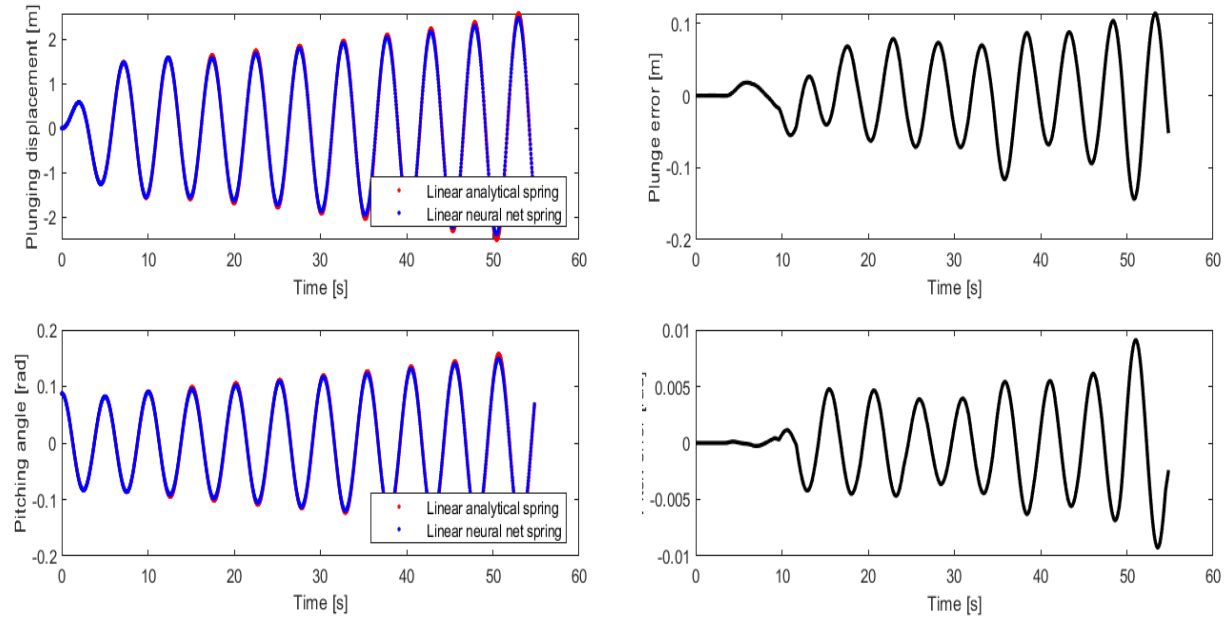

Figure 5.37: Plot of the linear analytical spring vs the linear neural network spring, pitching and plunging, $\alpha = \frac{5\pi}{180}$

Supercritical velocity - $\alpha = \frac{10\pi}{180}$

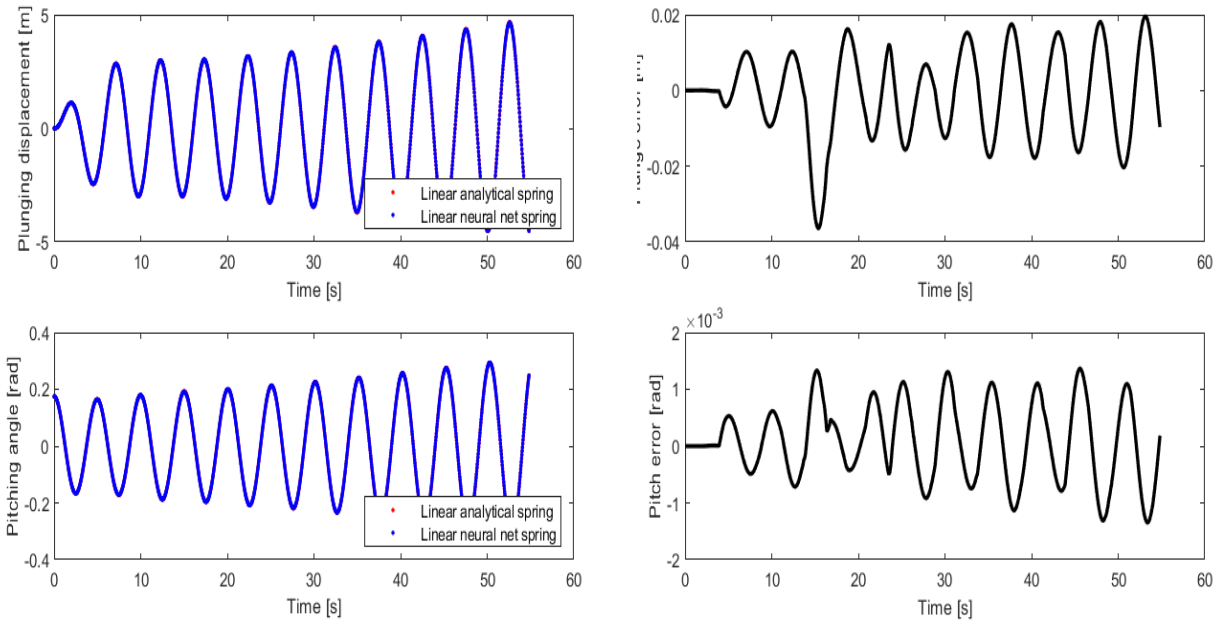

Figure 5.38: Plot of the linear analytical spring vs the linear neural network spring, pitching and plunging, $\alpha = \frac{10\pi}{180}$

In figures 5.36, 5.37 and 5.38 one can see the plunging height and pitching angle plotted against time in the left column of the figures and the difference between the computed height and rotation angle from the analytical spring and the neural network generated spring in the right column at the supercritical velocity for the different angles of attack. The error plot seems to increase for the supercritical velocity in the linear model for all initial angles.

Nonlinear spring

Here the analytical nonlinear spring for the plunge and pitch is compared against the neural network generated nonlinear spring for the aeroelastic model in the subcritical, critical and supercritical velocities. Here, the angle of attack is set to $\alpha = 1$ for all the simulations in the initial conditions.

Subcritical velocity - $\alpha = \frac{\pi}{180}$

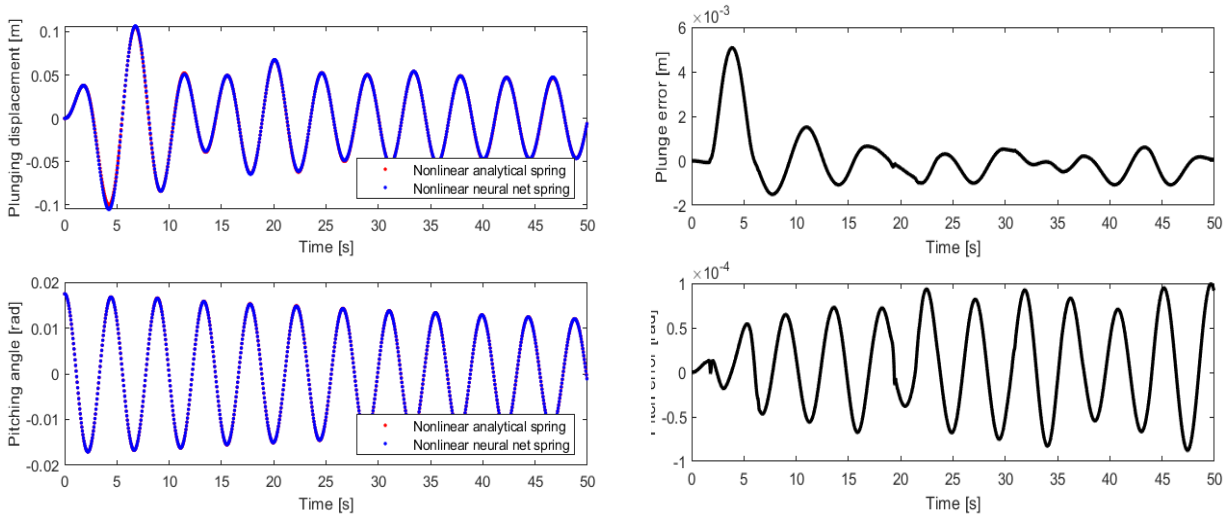

Figure 5.39: Plot of the nonlinear analytical spring vs the nonlinear neural network spring, pitching and plunging, $\alpha = \frac{\pi}{180}$

Critical velocity - $\alpha = \frac{\pi}{180}$

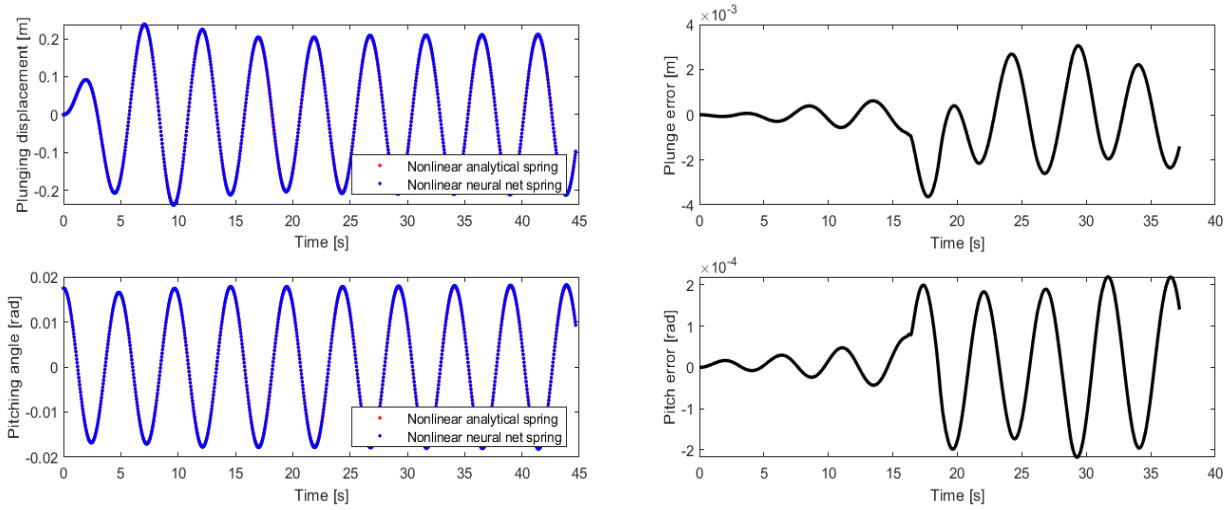

Figure 5.40: Plot of the nonlinear analytical spring vs the nonlinear neural network spring, pitching and plunging, $\alpha = \frac{\pi}{180}$

Supercritical velocity - $\alpha = \frac{\pi}{180}$

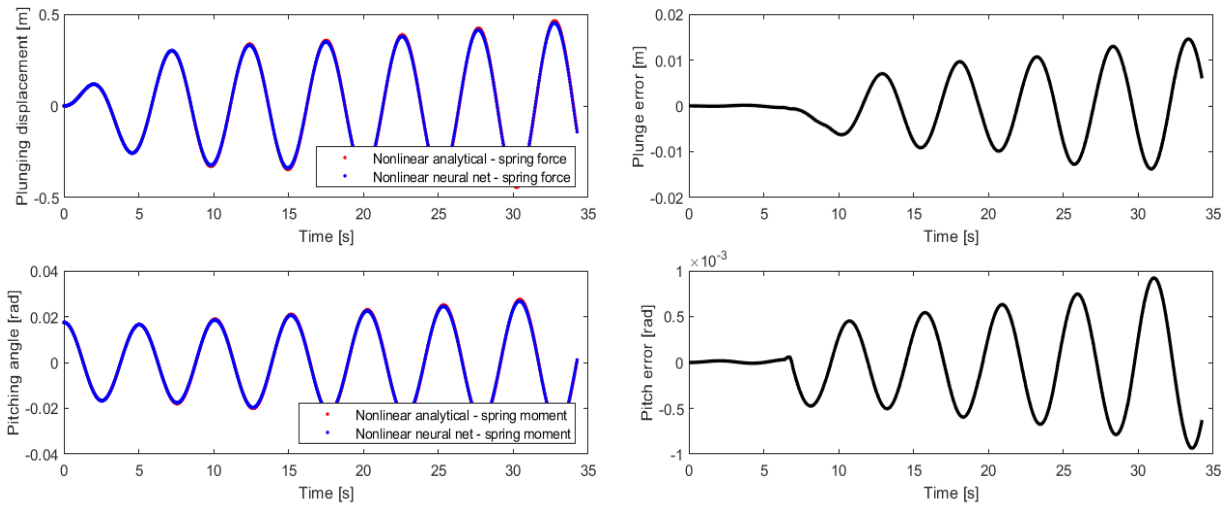

Figure 5.41: Plot of the nonlinear analytical spring vs the nonlinear neural network spring, pitching and plunging, $\alpha = \frac{\pi}{180}$

Figures 5.39, 5.40 and 5.41 show the plots of the aeroelastic results using the nonlinear spring. As one can see, the aeroelastic model using the nonlinear analytical spring seems to fit well with the aeroelastic model using the nonlinear neural network generated spring. For the subcritical velocity in figure 5.39, the error seems to decrease as time increases for the plunging motion. Similarly to the linear spring, the error increases in the supercritical velocity case. One reason why this error increases may be due to the input values for the height h and pitching angle θ are increasing. If theses values exceed the training range I_h for the plunge and I_θ for the pitch, the model using the neural network generated spring will not predict the correct response.

Wake shape comparison

Here, the wake shape for the two aeroelastic models using the neural network generated spring and the regular analytical spring are compared at the final time step. The wake scheme used for both are the explicit Euler method. Both the linear and nonlinear spring cases are displayed below. The angle of attack chosen for the wake shape comparison was set to $\alpha = 1$, for both the linear and nonlinear case.

Linear spring - wake shape comparison

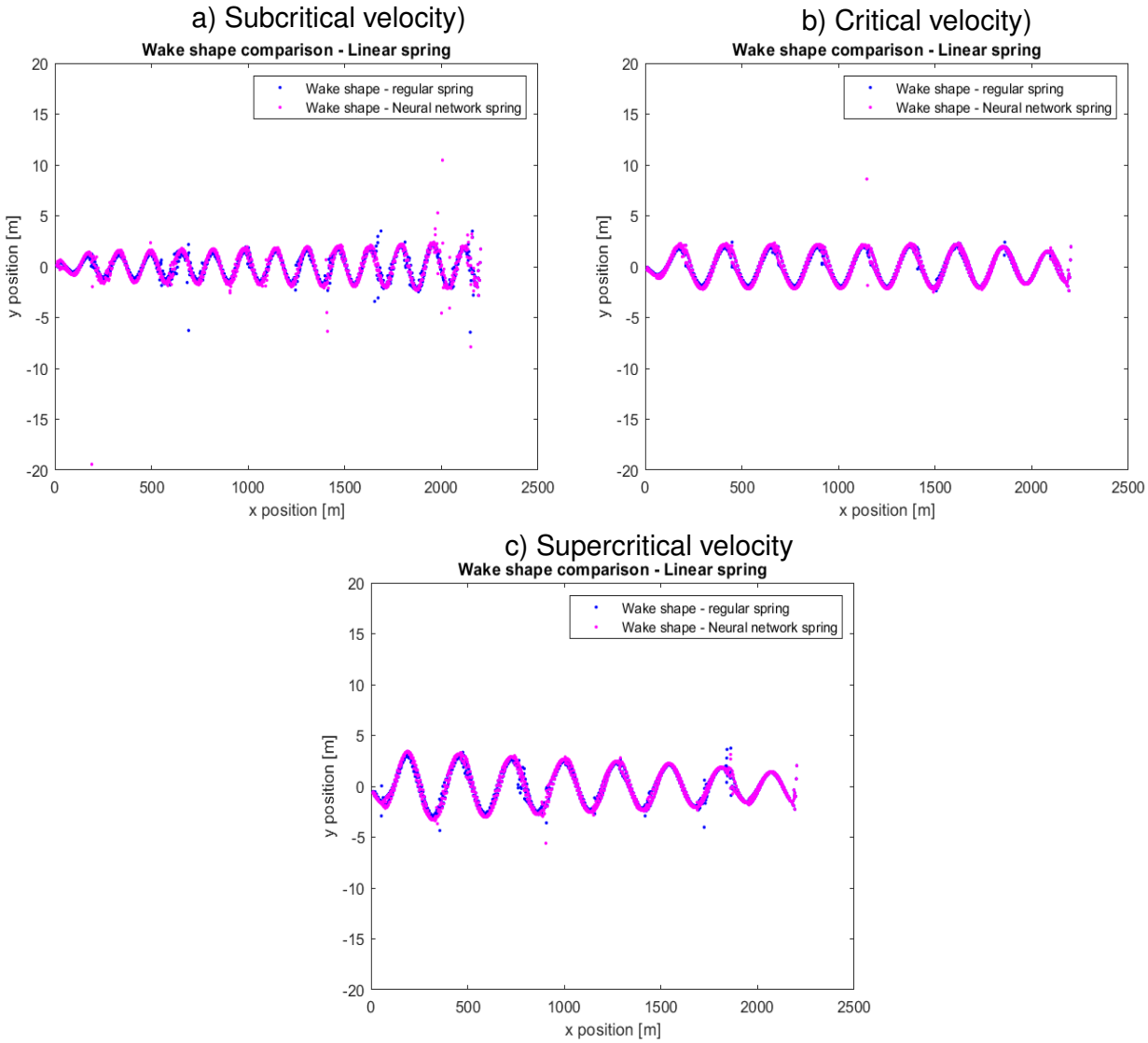

Figure 5.42: Comparison of the wake shapes for the analytical linear spring and neural network generated linear spring.

From figure 5.44, one can see that the wake shape pattern for the linear model using the trained neural network spring and for the regular analytical spring are more or less the same for all the three cases.

Nonlinear spring - wake shape comparison

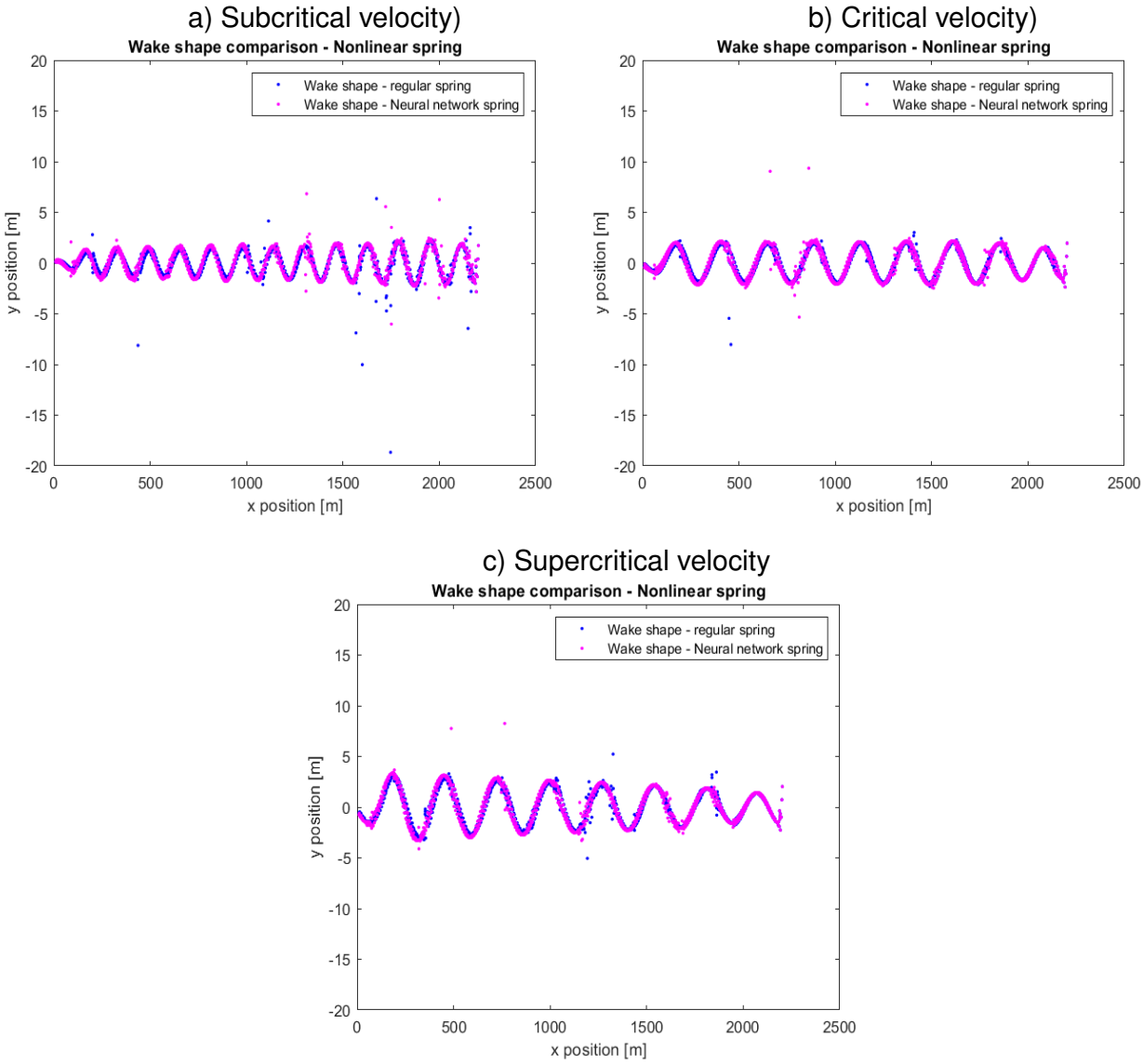

Figure 5.43: Comparison of the wake shapes for the analytical nonlinear spring and neural network generated nonlinear spring.

Similarly to figure 5.45, the wake shape for the nonlinear neural network spring and the analytical spring share the same pattern, but with some wake particles that are deviating.

Chapter 6

Conclusions and Future Work

In this study, an aeroelastic model was made, using the unsteady vortex particle method as the aerodynamic model and a predictor corrector numerical scheme for the structural model. In the beginning, the aerodynamic model was verified by comparing the numerical lift coefficient of the model with the analytical lift coefficient from the thin airfoil theory. The structural model for solving the equations of motion was verified by solving the linear pendulum problem and the aeroelastic model was verified, using the experimental results from the *Fung* (1955) case.

Furtermore, the Adam Bashforth methods were used and compared against the explicit Euler method for convecting the wake particles in the aerodynamic model. The results shows that using the Adam Bashforth methods in the wake scheme for the aerodynamical model has little influence in terms of the computed numerical lift coefficient. The Hausdorff distance shows that the wake shape for the different wake schemes are quite similar to each other, especially at the beginning of the simulation. This was consistent for both the steady flow state and for the periodical pitching and plunging flow conditions. From this observation, one can conclude that using linear multistep methods such as the Adam Bashforth methods will make little to no difference compared to the explicit Euler method.

As of right now, the trained neural network spring has shown promising results for the aeroelastic model. The aeroelastic model using the neural network trained spring terms have been able to accurately predict the plunging and pitching motion for the subcritical, critical and the supercritical velocity compared to the model using the analytical spring terms. This shows that even using the simple one hidden layer forward neural network for training the nonlinear and linear spring terms can accurately predict the correct fluttter onset for the aeroelastic model. Although these results are promising, its important to note that the cubic structural nonlinearity used is relatively simple, and other nonlinearities arising in the aeroelastic system have not been considered. In the future, it would be interesting to train a neural network for the other terms in the equation of motions, such as the martix for the damping of the airfoil.

Bibliography

Anderson, J. (2011), *EBOOK: Fundamentals of Aerodynamics (SI units)*, McGraw Hill. 1.1.10, 1.1.10, 1.2.2, 1.2.3, 1.2.4, 1.2.4, 1.2.5

Billah, K., and R. Scanlan (1990), Resonance, tacoma narrows bridge failure, and undergraduate physics textbooks, *American Journal of Physics*, *59*, 119–124. 0.2.3

Braun, C. (2007), Ein modulares verfahren für die numerische aeroelastische analyse von luftfahrzeugen. 4.1

Bryant, M., and E. Garcia (2009), Development of an aeroelastic vibration power harvester, in *Active and Passive Smart Structures and Integrated Systems 2009*, vol. 7288, pp. 409–418, SPIE. 0.2.3

de Souza, C., R. da Silva, and C. Cesnik (2012), Nonlinear aeroelastic framework based on vortex-lattice method and corotational shell finite element, in *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA*, p. 1976. 0.2.4

Dias, J., C. De Marqui Jr, and A. Erturk (2015), Three-degree-of-freedom hybrid piezoelectric-inductive aeroelastic energy harvester exploiting a control surface, *AIAA Journal*, *53*(2), 394–404. 0.2.3

Erturk, A., W. Vieira, C. De Marqui Jr, and D. Inman (2010), On the energy harvesting potential of piezoaeroelastic systems, *Applied physics letters*, *96*(18), 184,103. 0.2.3

Fung, Y. (1955), *An introduction to the theory of aeroelasticity*, Dover Publications. 5.1, 5.2.3, 5.2.3, 6

Garrick, I. (1936), *Propulsion of a flapping and oscillating airfoil*. 5.2.1

Gebhardt, C., and B. Roccia (2014), Non-linear aeroelasticity: an approach to compute the response of three-blade large-scale horizontal-axis wind turbines, *Renewable Energy*, *66*, 495–514. 0.2.4

Gebhardt, C., S. Preidikman, and J. Massa (2010), Numerical simulations of the aerodynamic behavior of large horizontal-axis wind turbines, *International Journal of Hydrogen Energy*, *35*(11), 6005–6011. 0.2.4

Genc, S. (2017), Parametric system identification using deep convolutional neural networks, in *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2112–2119, IEEE. 0.2.3

GWEC (), Global wind report 2022, https://gwec.net/wp-content/uploads/2022/03/GWEC-GLOBAL-WIND-REPORT-2022.pdf, accessed:2022-12-08. 0.2.2

Gülçat, (2010), *Fundamentals of modern unsteady aerodynamics*, springer. 5.2.1

Hansen, M. (2008), *Aerodynamics of Wind Turbines*, Routledge. 0.2.2

Hastie, T., et al. (2009), *The elements of statistical learning: data mining, inference, and prediction*, Springer Science & Business Media. 3.2.1

Hecht-Nielsen, R. (1992), *Theory of the backpropagation neural network*, 65-93 pp., Elsevier. 3.2.1

Hodges, D., and G. Pierce (2011), *Introduction to Structural Dynamics And Aeroelasticity, second edition.*, 243 pp., Cambridge University Press. 0.2.3, 5.2.3

IBM (), Supervised learning, https://www.ibm.com/cloud/learn/supervised-learning, accessed:2022-12-20. 3.2.1

IEA (), Wind electricity, https://www.iea.org/reports/wind-electricity, accessed:2022-12-10. (document), 0.2.2, 2

IPCC (), Mitigation of climate change, https://www.ipcc.ch/report/ar6/wg3/downloads/report/IPCC_AR6_WGIII_SPM.pdf, accessed:2022-12-08. 0.1

IRENA (), Wind energy, https://www.irena.org/Energy-Transition/Technology/Wind-energy, accessed:2022-12-23. 0.1

IRENA (2019), Future of wind: Deployment, investment, technology, grid integration and socio-economic aspects (a global energy transformation paper). 0.1

Joseph, D. (2006), Helmholtz decomposition coupling rotational to irrotational flow of a viscous fluid. 1.1.3

Katz, J., and A. Plotkin (2001), *Low-Speed Aerodynamics*, Cambridge Aerospace Series, Cambridge University Press. 1.1, 1.1.2, 1.1.7, 1.1.8, 1.1.9, 1.1.11, 2.1.1, 2.5, 4.1, 5.2.1

Kreiss, H.-O., and O. Ortiz (2014), *Introduction to numerical methods for time dependent differential equations*, John Wiley and Sons. 5.2.1

Krizhevsky, A., I. Sutskever, and G. Hinton (2017), Imagenet classification with deep convolutional neural networks, *Communications of the ACM*, *60*(6), 84–90. 0.2.3

Kumar, A., O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher (2016), Ask me anything: Dynamic memory networks for natural language processing, in *International conference on machine learning*, pp. 1378–1387, PMLR. 0.2.3

Kundu, P., and I. Cohen (2003), *Fluid Mechanics 2nd Edition*, Elsevier Exclusive. 1.1, 1.1.1, 1.1.3, 1.1.3

Lee, H., and D.-J. Lee (2019), Numerical investigation of the aerodynamics and wake structures of horizontal axis wind turbines by using nonlinear vortex lattice method, *Renewable Energy*, pp. 1121–1133. 0.2.4

Leishman, J. (2002), Challenges in modeling the unsteady aerodynamics of wind turbines, *Wind Energy*, *5*, 82–135. 0.2.2

Maiseli, B. (2020), Hausdorff distance with outliers and noise resilience capabilities. 5.3.1

Mcculloch, W., and W. Pitts (1943), *A logical calculus of the ideas immanent in nervous activity.*, 115-133 pp., The bulletin of mathematical biophysics. 3.2.1

NASA (), Aerodynamic center, https://www.grc.nasa.gov/www/k-12/VirtualAero/BottleRocket/airplane/ac.html, accessed:2022-12-20. 3.1

Pasqualetti, M. (2014), History of wind energy. 0.2.1

Rados, K., G. Larsen, R. Barthelmie, W. Schlez, B. Lange, G. Schepers, T. Hegberg, and M. Magnisson (2001), Comparison of wake models with data for offshore windfarms, *Wind Engineering*, *25*(5), 271–280. 0.2.4

Reimer, L., C. Braun, G. Wellmer, M. Behr, and J. Ballmann (2010), *Development of a Modular Method for Computational Aero-structural Analysis of Aircraft*, 205-238 pp., Springer Berlin Heidelberg. 4.1

Ritchie, H., et al. (2017), Annual co2 emissions by world region, https://ourworldindata.org/co2-and-other-greenhouse-gas-emissions. (document), 1

Roccia, B., S. preidikman, J. Massa, and D. Mook (2013), A modified unsteady vortex-lattice method to study the aerodynamics of flapping wings in hover flight, *AIAA Journal*, *51*, 2628–2642. 0.2.4

Roccia, B., M. Verstraete, L. Ceballos, B. Balachandran, and S. Preidikman (2020), Computational study on aerodynamically coupled piezoelectric harvesters, *Intelligent Material Systems and Structures*, *31*, 15–16. 0.2.3, 0.2.4, 0.2.5, 2, 4.1

Sauer, T. (2012), *Numerical Analysis*, Always learning, Pearson. 4.3.2, 5.2.2

Strganac, T., and D. Mook (1990), Numerical model of unsteady subsonic aeroelastic behavior, *AIAA journal*, *28*(5), 903–909. 0.2.4

Wang, Z., P. Chen, D. Liu, and D. Mook (2010), Nonlinear-aerodynamics/nonlinear-structure interaction methodology for a high-altitude long-endurance wing, *Journal of aircraft*, *47*(2), 556–566. 0.2.4

Young, H., R. Freedman, and A. Ford (2006), *Sears and Zemansky's University Physics*, no. v. 1 in Addison-Wesley series in physics, Pearson Addison-Wesley. 3.2