Towards Efficient Fluid-Structure-Control Interaction for Smart Rotors

Gillebaart, Thijs

DOI
10.4233/uuid:c078909a-a39c-47b5-8ca6-2cbc9e04486e

Publication date
2016

Document Version
Final published version

Citation (APA)
Gillebaart, T. (2016). *Towards Efficient Fluid-Structure-Control Interaction for Smart Rotors*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:c078909a-a39c-47b5-8ca6-2cbc9e04486e

Towards Efficient Fluid-Structure-Control Interaction for Smart Rotors

TOWARDS EFFICIENT FLUID-STRUCTURE-CONTROL INTERACTION FOR SMART ROTORS

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op maandag 30 mei 2016 om 12:30 uur

door

Thijs GILLEBAART

ingenieur in de Luchtvaart- en Ruimtevaart
geboren te Heemskerk, Nederland.

Dit proefschrift is goedgekeurd door de

promotor: Prof. dr. ir. drs. H. Bijl
copromotor: Dr. ir. A.H. van Zuijlen

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. ir. drs. H. Bijl,	Technische Universiteit Delft
Dr. ir. A.H. van Zuijlen	Technische Universiteit Delft
Onafhankelijke leden:	
Prof. dr. S. Hickel	Technische Universiteit Delft
Prof. dr. G.A.M. van Kuik	Technische Universiteit Delft
Prof. dr. H. Jasak	University of Zagreb
Prof. dr. N.N. Sørensen	Technical University of Denmark
Prof. dr. C. Allen	University of Bristol

Front & Back: Cover art designed by my dear friend Floortje Zonneveld with assistance of Roeland Verhallen.

ISBN 000-00-0000-000-0

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

SUMMARY

Climate change is one of the biggest problems of the 21st century. Due to the large production of greenhouse gasses in the past, today and the future, climate change will worsen if appropriate measures are not taken. Therefore there is a need for transitioning from fossil fuel energy resources to renewable energy resources, to reduce the production of greenhouse gasses. One of these renewable energy resources is wind energy. Reducing the Cost of Energy (CoE) (i.e. price) for wind energy will be a strong incentive to transition faster from fossil fuels to renewable sources. Enlarging wind turbines potentially reduces the CoE, but comes with the cost of increasing load and stiffness requirements, due to the increase in fatigue and extreme loads.

One of the approaches to meet these new challenges is the smart rotor concept: wind turbine blades with actively controlled Trailing Edge Flaps. In the past decade feasibility studies (both numerical and experimental) have been performed to assess the applicability of smart rotors in future design strategies. These studies have shown the potential of Trailing Edge Flaps for smart rotors, but higher fidelity models (Computational Fluid Dynamics coupled with Computational Structural Dynamics) are required to assess the details (e.g. load distribution in both span-wise and chord-wise direction) of such a system. In addition, such a model will be a platform for analyzing new concepts for which no experimental prototype data is available. This fits within the trend of using higher fidelity models in combination with the engineering models for engineering applications to reduce the uncertainties in the analysis. For the aerodynamic model the often used incompressible Navier-Stokes equations are used. Since the rotors perform at high Reynolds numbers, resolving the boundary layer is a challenge. Using a body-fitted mesh has the benefit of having a high mesh density near the wall, while keeping the size of the mesh to a minimum. Ideally, this mesh moves along with the deformation imposed by the structure. Therefore, the Arbitrary Lagrangian Eulerian (ALE) formulation of the Navier-Stokes equations is used in this study, as also done in most other studies.

This thesis studies two issues within high fidelity modeling of smart rotors: 1) Computational Fluid Dynamics based Fluid-Structure Interaction models are computationally expensive and efforts should be focused on making them more efficient and 2) how well is a high fidelity model able to predict smart rotors/airfoils compared to experimental data and engineering models? To increase the efficiency, two parts of the FSI model contributing significantly to the computational work are considered: time integration for partitioned FSI models and mesh deformation. Consistent higher order time integration ensures that larger time steps can be used, decreasing the computational work required. Using the incompressible Navier-Stokes in combination with moving collocated grids (for mesh topology flexibility) imposes a challenge for consistent time integration. Consistent time integration for static grids has been a studied intensively due to the difficulties with the required non-linear momentum interpolation. Extending this

approach to moving grids is the first step towards consistent time integration for a FSI model. In static grids only a fixed orientated face flux is required. However, this face flux is the inner product of the face velocity and face normal times the face area. Changes in the face orientation (i.e. face normal), due to grid motion, are accounted for by a adding the face normal change flux term. A similar approach is derived for the change in face area, resulting in a second term compensating the flux for the varying face area. To achieve consistent time integration on collocated grids for incompressible flow, this is required for the fluxes from all time instances/stages used within the discretization. A circular cavity case is used to show time consistent integration for first, second and third order backward differencing schemes. Combined with the discussed consistent boundary condition, force coupling and structural model third order FSI is achieved for the academic circular cavity case. In addition, two benchmark FSI problems are used to show the increase in efficiency by using consistent time integration on collocated grids for incompressible flow.

The second part of the increase in efficiency is obtained by further developing Radial Basis Function mesh deformation into an efficient and robust method no longer requiring detailed a priori knowledge of the structural deformation. Within the state-of-the-art RBF mesh deformation two problems are identified: 1) a-priori knowledge and data of the structural deformation is required for proper control point selection, and 2) how to automate the correction of the boundary error resulting from the boundary interpolation needed due to the data reduction algorithm. The greedy data reduction algorithm is used to adaptively select a set of control points during a FSI simulation. By normalizing the boundary error by displacement amplitude, a set of control points is found representing the deformation shape independently of the amplitude. Only when the shape alters during a simulation a new set of control points is selected, always ensuring an efficient number of control points. To correct the boundary error, an automated explicit correction method is derived.

By means of a detailed analysis of a single high aspect ratio cell, the relation between the mesh quality and aspect ratio, first cell height, boundary error and the properties of the RBF function is derived. Based on the analysis two alternative boundary correction functions are proposed, which perform better within the domain where the properties of the correction function are dominant. A 2D airfoil with oscillating flap at varying Reynolds number (and thus with varying aspect ratio and first cell height) validates the proposed automated explicit boundary correction for a set of known correction functions and the newly derived functions. The same case and a 3D flexible tube are used to show the efficiency compared with other RBF mesh deformation methods for parallel computations. For a temporally varying deformation shape the adaptive RBF mesh deformation with clean re-selection is most efficient, while for a fixed or periodically varying deformation shape the adaptive RBF mesh deformation with re-use selection is most efficient. With the automated boundary correction function and adaptive selection algorithm a robust and efficient mesh deformation method is developed.

Finally, the increase in confidence/insight in the CFD based models is achieved by means of validation of unsteady flap aerodynamics and performing aero-servo-elastic simula-

tions of an airfoil with flap in gusty conditions using both CFD and an dynamic stall based unsteady aerodynamic model. Both steady-state and unsteady experimental validation data of a static wind turbine airfoil with oscillating flap is used to validate the CFD model together with an widely used engineering model. In steady-state reasonable agreement is found (especially in the linear regime), even though the forced-transition results indicate unexpected influences of the zig-zag strip. For the unsteady results the agreement becomes worse when increasing the reduced frequency and flap deflection amplitude. A detailed analysis of the pressure, lift, drag and moment distributions for specific cases revealed irregularities in the pressure distributions of the experimental data, especially near the flap hinge. Although the validation study provided refreshing insight in the global accuracy of the CFD and engineering model, it also motivates to analyze the experiments and related data in more detail.

Combining the CFD model with a 3 Degree of Freedom structural model, deformable flap and a controller the 2D Fluid-Structure-Control Interaction model is completed. For two types of gusts (cosine and Mexican hat) with varying reduced frequency at different angles of attack simulations are performed with both the high fidelity model as well as the engineering model. Adding a controller (and thus flap deflections) increases the relative differences, although the absolute differences are relatively small at attached flow cases. However, the flap deflection amplitude shows significant differences under all flow conditions. For intermediate gust frequencies (reduced frequency of 0.2) differences are largest, especially for partly or fully separated flow conditions. This indicates a possible limit of engineering models, although further experimental validation would provide a better insight.

Both the time integration method as well as the RBF method, are ready for large scale (3D) problems and thus for application within the FSCI model of a smart rotor. With the validation study and the direct comparison of the aero-servo-elastic response to a gust, first steps are made to increase the confidence of the method, or at least to quantify its accuracy compared to experiments and an engineering model.

SAMENVATTING

Klimaatverandering is één van de grootste problemen van de 21ste eeuw. Door de grote hoeveelheid geproduceerde broeikasgassen in het verleden, vandaag en in the toekomst, zal klimaatverandering verergeren als de juiste maatregelen niet genomen worden. Daarom is de wens er om over te gaan van fossiele brandstoffen als energie bron naar duurzame energie bronnen, zodat the productie van broeikasgassen verminderd. Een van deze duurzame energie bronnen is wind energie. Het verlagen van de kosten voor wind energie zal een sterke aansporing zijn om sneller om te schakelen van fossiele brandstoffen naar duurzame energie vormen. Wind molens groter te maken kan potentieel de kosten verlagen, al zal dit de stijfheidseisen en krachten vergroten, door de toename in vermoeiingskrachten en extreme krachten.

Een mogelijke oplossing voor deze nieuwe uitdagingen is het smart rotor concept: wind molen bladen met actief gestuurde trailing edge flaps. In het afgelopen decennium zijn er haalbaarheidsstudies (zowel experimenteel als numeriek) uitgevoerd om de toepasbaarheid van smart rotors in toekomstige ontwerpen strategieën vast te stellen. Deze studies laten het potentieel van trailing edge flaps zien, maar hoogwaardige modellen (Computational Fluid Dynamics gekoppeld met Computational Structural Dynamics) zijn nodig om de details (bv. kracht verdelingen in zowel span-wise als chord-wise richting) van het systeem te onderzoeken. Daarnaast, kan het model gebruikt worden als platform voor het analyseren van nieuwe concepten waarvoor geen experimentele prototype beschikbaar is. Dit past goed in de trend dat hoogwaardigere numerieke modellen in combinatie met ingenieurs modellen gebruikt worden in industriële toepassingen zodat de onzekerheden verkleind kunnen worden. Als aerodynamisch model worden de veelgebruikte incompressibele Navier-Stokes vergelijkingen gebruikt. Reynolds getallen voor wind molens zijn groot, wat ervoor zorgt dat het oplossen van de grenslaag een uitdaging is. Het gebruik maken van een body-fitted rooster heeft het voordeel dat er makkelijk een hoge cel dichtheid bij de wand kan worden toegepast, terwijl de grote van het rooster beperkt blijft. Idealiter, beweegt dit rooster mee met de vervorming van de structuur. Daarom worden de Arbitrary Lagrangian Eulerian (ALE) vorm van de Navier-Stokes vergelijkingen gebruikt in deze studie, zoals ook gedaan wordt in de meeste andere smart rotor studies.

Deze thesis bestudeert twee kwesties binnen het hoogwaardig modeleren van smart rotors: 1) Computational Fluid Dynamics gebaseerde Fluid-Structure Interaction modellen kosten veel computer kracht en dus focust het onderzoek zich op het efficiënter maken van deze modellen, en 2) hoe goed is een hoogwaardig model in het voorspellen van smart rotors/airfoils vergeleken met experimentele data en ingenieurs modellen? Om de efficiënte te verhogen, twee onderdelen van het FSI model die significant bijdragen aan de reken kosten worden bestudeerd: tijds-integratie voor gepartitioneerde FSI modellen en roostervervorming. Consistente hogere orde tijds-integratie zorgt er-

voor dat grotere tijdstappen gebruikt kunnen worden, wat resulteert in een verlaging van de reken kosten. Het gebruik van de incompressibele Navier-Stokes vergelijkingen in combinatie met bewegende collocated roosters (vanwege rooster topologie flexibiliteit), maakt consistente tijds-integratie een uitdaging. Consistent tijds-integratie voor statische collocated roosters is uitgebreid bestudeerd, vanwege de moeilijkheden met de benodigde niet-lineaire momentum interpolatie. Het uitbreiden van deze aanpak naar bewegende roosters is de eerste stap naar consistente tijds-integratie voor het FSI model. In statische roosters is de face-flux altijd georiënteerd in dezelfde richting. Echter, deze face-flux is een inwendig product tussen face-snelheid en face-normaal maal de face-oppervlakte. Veranderingen in de face-normaal door roostervervorming moet worden meegenomen in de face-flux. Eenzelfde aanpak is afgeleid voor de verandering van de face-oppervlakte, welke resulteert in een tweede term die meegenomen moet worden in de face-flux. Om consistente tijds-integratie op collocated roosters voor incompressibele stromingen te bereiken, moeten deze twee termen worden meegenomen voor alle tijds-instanties die gebruikt worden voor de tijds-integratie. Een cirkelvormige holte stroming probleem wordt gebruikt om consistent tijds-integratie te laten zien voor de eerste, tweede en derde orde backward differencing schema's. Gecombineerd met de consistente randvoorwaarden, krachtkoppeling en structurele vergelijkingen wordt derde orde tijds-integratie voor het FSI model bereikt. Daarnaast worden twee benchmark FSI problemen gebruikt om aan te tonen dat de consistente tijds-integratie leidt tot een hogere efficiëntie voor incompressibele stromingen op collocated roosters.

Het tweede deel van het verhogen van de efficiëntie is bereikt door het verder ontwikkelen van de radiale basisfunctie roosterververoming in een efficiënte en robuuste methode die niet langer gedetailleerde a priori kennis van de structurele vervorming nodig heeft. Twee problemen zijn geïdentificeerd binnen radiale basisfunctie roostervervorming: 1) a priori kennis en data van de structurele vervorming is nodig voor het goed selecteren van controle punten, en 2) hoe moet de randinterpolatiefout correctie, die nodig is door het gebruik van de data reductie techniek, worden geautomatiseerd? De greedy methode wordt gebruikt om adaptief de controle punten selectie te maken tijdens een FSI simulatie. Een set van controle punten die de vervormingsvorm goed representeert onafhankelijk van de amplitude wordt gevonden door het normaliseren van de randfout met de verplaatsingsamplitude. Alleen wanneer de vervormingsvorm varieert tijdens een simulatie wordt er een nieuwe set van controle punten geselecteerd, waardoor gedurende de simulatie een efficiënt aantal controle punten wordt gebruikt.

Doormiddel van een gedetailleerde analyse van een grote aspect ratio cel, is een relatie gevonden tussen de roosterkwaliteit en aspect ratio, eerste cel hoogte, randfout en de eigenschappen van radiale basisfunctie. Gebaseerd op deze analyse zijn er twee alternatieve randcorrectie functies voorgesteld. Deze twee alternatieve functies presteren beter binnen het domein waar de eigenschappen van de correctiefunctie dominant zijn. Roosterkwaliteit resultaten van een 2D airfoil met oscillerende flap op variërende Reynolds getallen (en dus variërende aspect ratio en eerste cel hoogte) laten zien dat beschreven automatische expliciete randcorrectie goed werkt voor een aantal bekende correctiefuncties en de twee nieuwe correctiefuncties. Hetzelfde probleem en een 3D flexibele buis worden gebruikt om de efficiëntie van de adaptieve methode te vergelijken

met bestaande methodes. Voor een tijdsvariërende vervormingsvorm is de meeste efficiënte methode de adaptieve RBF roostervervorming met complete herselectie. Voor een periodisch variërende vervormingsvorm is de meeste efficiënte methode de adaptieve RBF roostervervorming met behoudende herselectie. Met deze geatomiseerde randcorrectiefunctie en het adaptieve selectie algoritme is er een robuuste en efficiënte roostervervormingsmethode ontwikkeld.

Uiteindelijk, om inzicht te vergaren in CFD gebaseerde modellen is een validatie studie gedaan met een oscillerende flap en het vergelijken van CFD resultaten en ingenieurs model resultaten voor een aero-servo-elastic simulatie van een airfoil met flap in een windvlaag. Zowel steady-state als unsteady experimentele validatie data van een wind molen profiel met oscillerende flap wordt gebruikt om het CFD model en een veelgebruikte ingenieurs model te valideren. Voor de steady-state resultaten is een redelijke overeenkomst gevonden (vooral in het lineaire domein), hoewel de geforceerde-transitie resultaten onverwachtse invloeden van de zig-zag strip laten zien. Een gedetailleerde analyse van de druk, lift, drag en moment distributies laten onregelmatigheden zien in de druk verdelingen van de experimenten, vooral in de buurt van flap scharnier. Hoewel the validatie studie interessante inzichten geeft in de globale nauwkeurigheid van het CFD en ingenieurs model, motiveert het ook om de experimenten en gerelateerde data verder te bestuderen.

Vervolgens wordt het CFD model gecombineerd met een structuur model met 3 vrijheidsgraden, een vervormende flap en een controller, wat resulteert in het 2D Fluid-Structure-Control Interaction model. Voor twee windvlaag types (cosinus en Mexican hat) met variërende frequentie en invalshoek zijn simulaties uitgevoerd met het CFD en ingenieurs model. Het toevoegen van een controller (en dus flapvervorming) vergroot het verschil tussen de modellen. Hierbij moet wel vermeld worden dat de absolute verschillen relatief klein zijn voor aanliggende stromingen. Het grootste verschil is te zien in flapvervormings-amplitude voor alle invalshoeken. Voor tussenliggende windvlaag frequenties zijn de verschillen het grootste, vooral voor de gedeeltelijk losgelaten of volledig losgelaten stromingsomstandigheden. Deze resultaten geven een mogelijke limiet aan van de ingenieurs modellen, hoewel verdere experimentele validatie een beter inzicht zou moeten geven.

Zowel de tijds-integratie methode als de RBF methode zijn klaar om toegepast te worden voor grotere (3D) problemen zoals binnen het FSCI model van een smart rotor. Met de validatie studie en de vergelijking tussen een aero-servo-elastic reactie op een windvlaag, zijn eerste stappen gezet om het inzicht in de nauwkeurigheid van het hoogwaardige model ten opzichte van experimenten en een ingenieurs model te vergroten.

CONTENTS

1

INTRODUCTION

1

Climate change is one of the biggest problems of the 21st century. Due to the large production of greenhouse gasses in the past, today and the future, climate change will worsen if the appropriate measures are not taken. Therefore there is a need for transitioning from fossil fuel energy resources to renewable energy resources, to reduce the production of greenhouse gasses.

One of these renewable energy resources is wind energy. Increasing the acceptance and the application of renewable energy resources is partly driven by how competitive the renewable resources are compared to the fossil fuel resources. In the past decade, the importance of wind energy in the energy market has been growing significantly. Due to the competitive energy market, a strong focus is on reducing the Cost of Energy (CoE) for wind energy, such that a larger market position can be obtained.

Due to the focus on reduction of CoE, a clear trend is found in the wind turbine industry: enlarging wind turbines and their blades. The trend of the past decades is illustrated in Figure 1.1, coming from the Upwind project [1]. Simultaneously with the

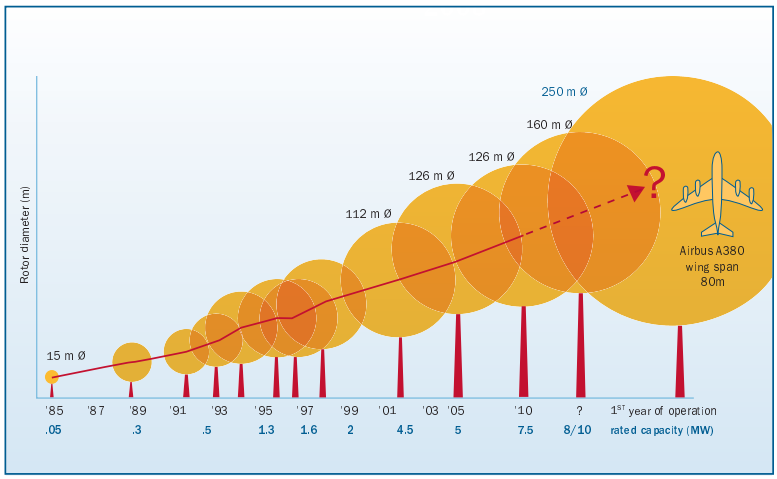

Figure 1.1: Illustration of wind turbine size and power trend in the past decades. Illustration is adopted from the Upwind project tech report from 2011 [1].

increasing size of turbines, also load and stiffness requirements increase strongly. Both the steady loads as well as the unsteady (fatigue and extreme) loads increase. One of the main drivers for the CoE are the maintenance cost (especially for offshore wind turbines). Maintenance is required due to the wear and tear by fatigue (unsteady) loading and gust loads. Especially the root bending moment is influenced by these steady and (unsteady) fatigue loads, since the increase in radius directly affects the bending moment. The increase in unsteady loads is governed by the larger variation in loading due to the bigger variation in unsteady flow conditions encountered while the blade rotates. This is partly because of the Atmospheric Boundary Layer (ABL) including the shear layer, from which a bigger part is swept by the larger blades. Also intrinsic unsteady behavior of turbulent flow, and gusts (i.e. extreme loads/events), which possibly hit parts of the turbine are the cause of increase of unsteady loads.

One of the approaches to meet these new challenges is the smart rotor concept [2]. Active and passive control devices are added to conventional blade designs, to increase the control authority of the turbines, dealing with the unsteady nature of the environment. These control devices can be used for reducing fatigue loads (an important design driver) and decreasing extreme loads, or both.

From the above it is clear how smart rotors by reducing fatigue and extreme loads, reduce maintenance cost and thereby contribute to a lower CoE and making renewable energy (in this particular case: wind energy) more competitive. Smart rotors are considered to start playing an important role in the near future, which is supported by the large increase in smart rotor research. But what are the current challenges and bottlenecks in smart rotor research? In this chapter a summary is given of the smart rotor concept, with the main focus on one of the most promising active control devices: (deformable) trailing edge flaps (TEF). After the introduction into smart rotors in Section 1.1, the high fidelity modeling of such a smart rotors is discussed in Section 1.2. Based upon these two sections the goal of the thesis and the corresponding outline are stated in Section 1.3.

1.1. SMART ROTORS

Barlas and van Kuik reviewed the developments in smart rotor control in 2011 [2]. They discussed the different concepts extensively: flaps, microtabs, camber control, active twist and boundary layer control. Each of these concepts aim for a (fast) change in lift to alleviate the unsteady loads and thereby reducing the fatigue and/or extreme loads. Barlas and van Kuik conclude that [2]:

> *Trailing edge flap control seems to be one of the most efficient of the proposed aerodynamic control surfaces. The change in lift and drag characteristics as well as the linearity, the bandwidth and the simplicity of this concept makes it attractive from the control point of view.*

Even though microtabs and camber control show similar ΔC_l compared to flaps, there are two disadvantages compared to a (deformable) TEF. Microtabs obtain their highest ΔC_l only in reducing the lift by causing flow separation. Secondly, microtabs do have non-linear dynamics when deploying, due to the non-linear aerodynamics associated with the change of the trailing edge flow development (Kutta condition). For camber control, the largest drawback is the strain the skin needs to be able to take throughout the section and blade. Although (a combination of) the other concepts are still studied and potentially beneficial for a wind turbine, in this study the focus is on the application of (Deformable) Trailing Edge Flaps in wind turbine blades. An illustration of the increase in size of the turbines combined with TEF is shown in Figure 1.2.

A TEF deforms the trailing edge according to appropriate input signals, such that the loads on the wing/blade are reduced. For the TEF this is done based on a conventional flap system, while a deformable TEF (DTEF) is based on a smoother deformation of the trailing edge part of the wing. Initial studies focussed on the proving the feasibility by studying the effects of a TEF on a 2D airfoil. This has been done both numerically using

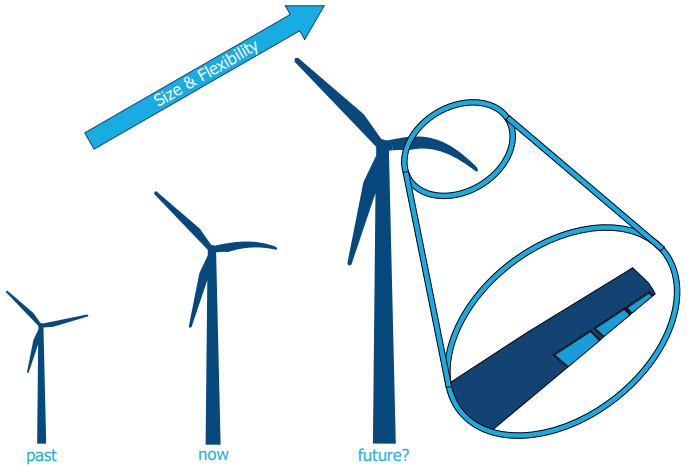

Figure 1.2: Illustration of wind turbine size increase and the application of Trailing Edge Flaps.

Theodorsen's approach [3, 4], potential flow [5] and experimentally [6–8]. From these studies it became clear that with a properly chosen control strategy the unsteady loads can be reduced significantly.

As continuation on these initial studies, more extensive (3D) numerical and experimental studies were performed [9–12]. From these studies, it became clear that adding flaps at the outer-board of the blades is a good strategy for load alleviation. In search of different strategies, combinations with individual pitch control are studied to further increase the control possibilities of modern wind turbines [13–15]. Bernhammer et al. analyzed the current state of smart rotor research and concluded that one of the future directions should be on a more detailed look at the applications of the TEF: damping of periodic loading, damping of stochastic nature of the inflow and flutter suppression, which are potential problems for future turbines [16].

Studies have shown the potential of Trailing Edge Flaps for smart rotors, but higher fidelity models are required to assess the details (e.g. load distribution in both span-wise and chord-wise direction) of such a system. In addition, such a model will be a platform for analyzing new concepts for which no experimental prototype data is available. This fits within the trend of using higher fidelity models for engineering applications to reduce the uncertainties in the analysis in combination with the engineering models currently used.

1.2. HIGH FIDELITY MODELING OF SMART ROTORS

A high fidelity model of a smart rotor consists out of the following parts: Computational Fluid Dynamics (CFD) aerodynamic model, Computational Structural Dynamics (CSD) structural model, mesh deformation, coupling of aerodynamic and structure, time integration and control. Such a model is also called a Fluid-Structure-Control interaction (FSCI) model, as illustrated in Figure 1.3.

CFD modeling of wind turbines has been applied for some years already. In 2011, Hansen and Madsen wrote an elaborate review on wind turbine aerodynamics, including the use of CFD [17]. As early as 1994, the first studies using CFD for wind turbines are reported [18–21]. Due to the experiments on the NREL Phase-II rotor, validation data became available [22]. After Sørensen et al. showed that CFD was a good model for predicting the pressure distribution along the blade [23], CFD for wind turbines became an active research area. Where in the first studies steady-state Reynolds Averaged Navier-Stokes were used, the current state-of-the-art involves Delayed Detached Eddy Simulation (DDES), resulting in unsteady simulations on large meshes often using High Performance Computing Clusters (HPCC) [24–26].

As Hansen and Madsen mention in their review paper, both incompressible and compressible solvers are used for the simulations of wind turbines. Generally, the Mach numbers are around 0.1 at the tips with a maximum below 0.3, making the assumptions of incompressibility valid. For the compressible solvers low Mach number preconditioning is used for efficient simulations. Often finite volume methods are used [24, 27–30], but finite element methods are also applied [31]. With the transition from steady-state calculations towards unsteady simulations of wind turbines, Fluid-Structure Interaction (FSI) for wind turbines became a possibility too.

In the past years high fidelity modeling of wind turbines using FSI has been gaining interest. Bazilevs et al. was the first to perform FSI simulations on a single blade of a NREL rotor in 2011 [31, 32]. Hereafter, Hsu and Bazilevs performed simulations on a full turbine [33]. They validated their results with the technical report [34] of the rotors and found good correspondence in results. Quickly after, other research groups started developing and presenting their methods and results for wind turbine blade FSI [27, 28, 30, 35, 36]. As for the studies mentioned above either an incompressible or a compressible flow model is used in combination with finite volume or finite element discretization. In this study an incompressible model is used, since the flow regime considered is well below Mach number of 0.3. However, it should be noted that a compressible model could be used equally well.

Since the rotors perform at high Reynolds numbers, resolving the boundary layer is a challenge. Using a body-fitted mesh has the benefit of having a high mesh den-

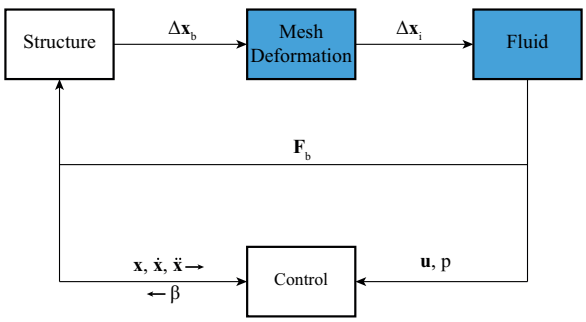

Figure 1.3: Schematic representation of a Fluid-Structure-Control-Interaction model. Time integration is applied over the complete model.

sity near the wall, while keeping the size of the mesh to a minimum. Ideally, this mesh moves along with the deformation imposed by the structure. Therefore, the Arbitrary Lagrangian Eulerian (ALE) formulation of the Navier-Stokes equations is used in this study, as also done in the studies presented above. Using the ALE formulation keeps the mesh size to its minimum, while still having a high accuracy of the loading on the blades, due to the moving/deforming body conforming mesh. Because of the high Reynolds numbers, high aspect ratio boundary layer grids are generally used around the body (blade, tower and nacelle), reducing the number of cells needed significantly, while still resolving the boundary layer accurately. Using the ALE formulation, in combination with a proper mesh deformation method, ensures that this boundary layer grid can be moved almost rigidly with the deformation of the wind turbine, resulting in a accurately resolved boundary layer for moving/deforming bodies. This is illustrated in Figure 1.4 with a moving body fitted mesh around an airfoil.

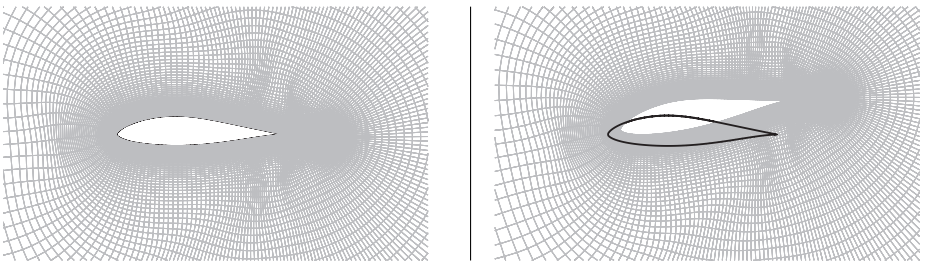

Figure 1.4: Boundary layer mesh with high aspect ratio cells surrounding an airfoil. Airfoil translates vertically and rotates around its quarter chord point, while the high aspect ratio mesh is moved along side with it. Black thick line indicates the original location of the airfoil.

Performing FSCI on smart rotors is only recently starting to gain interest. Heinz et al. started by assessing a 2D elastically suspended airfoil with controlled flap for different conditions flow conditions and controller designs [37]. They concluded that the CFD model is able to predict significant reduction of the loads. However, one of problems they encountered was the computational work required for the high fidelity model, when compared to the engineering models available. Another 2D rigid airfoil study focussed on the effectiveness of the flap by varying the flap length and the phase difference between the unsteady loading and the flap actuation [38]. Bergami et al. compared different computational models (CFD, vortex panel method and a engineering model) for a prescribed flap motion. Significant differences were found between both the steady results, unsteady results and also the computational time (in which the CFD model was the most expensive). They suggest a detailed experimental campaign to assess accuracy of the different models.

Jost et al. performed, as one of the first, 3D simulations of a rigid smart rotor blade with a deflected flap [39]. They performed a feasibility study of the methods used, and concluded that deforming the mesh around the flap with RBF is an efficient method of simulating wind turbine blades with flaps. Finally, within INNWIND and AVATAR (two European projects), part of the effort is also allocated to developing FSCI models of smart rotors with active flap control based on CFD combined with FSI [40, 41]. Here higher

fidelity models are developed to both give a more detailed insight in the physics, and to improve upon the engineering models available.

By combining the studies on FSI on wind turbine blades with the newest developments in smart rotor research, smart rotor FSCI will be one of the next steps in high fidelity research of smart rotors. One of the bigger drawbacks of high fidelity models is the significantly higher computational work required, while the accuracy of high fidelity smart rotor models are not fully assessed yet. The main focus in this thesis is on CFD, mesh deformation and time integration (highlighted in Figure 1.3). Structural modeling, FSI coupling and control are equally important for efficiency and accuracy, but developments within these parts are out of the scope of this thesis.

1.3. THESIS GOAL & OUTLINE

From the current developments within academia and industry it becomes clear that there are two major issues within high fidelity modeling of smart rotors: 1) CFD based FSI models are computationally expensive and efforts should be focussed on making them more efficient and 2) how well is a high fidelity model able to predict smart rotors/airfoils compared to experimental data and engineering models? Therefore, the goal of this thesis is:

> *Increase the efficiency and confidence level of incompressible CFD for smart airfoils and rotors.*

To increase the efficiency, two parts of the FSI model contributing significantly to the computational work are considered: time integration for partitioned FSI models and mesh deformation. Consistent higher order time integration ensures that larger time steps can be used, decreasing the computational work required. Mesh deformation is intrinsically needed when using the preferred ALE formulation of the Navier-Stokes equations. Ideally, the mesh deformation does not contribute significantly to the computational work and is robustly applicable without extensive a priori knowledge of the structural deformation/displacement. Confidence in the CFD based models is achieved by means of validation of unsteady flap aerodynamics. Finally, aero-servo-elastic simulations of an airfoil with flap using both CFD and an dynamic stall based unsteady aerodynamic model under different gust conditions are used to increase the confidence for 3D FSCI simulations in the future.

1.3.1. TIME CONSISTENT FLUID-STRUCTURE INTERACTION

Due to the unsteady nature of the physics, time integration is a crucial part of the model. Consistent (higher order) time integration can significantly reduce the computational time for FSI models [42]. For wind turbine simulations, the aerodynamics are often modeled by the incompressible Navier-Stokes equations. Due to the inclusion of the coupling between the aerodynamics and the structure (FSI), the ALE form of the equations are used. Due to the complexity of meshing wind turbine (blades) the preferred type of grids are collocated (cell centered), having the benefit of dealing with (partially) unstructured meshes relatively easy. OpenFOAM is such a tool: an open source polyhedral cell finite volume method [43].

Consistent time integration of the incompressible Navier-Stokes on static collocated meshes is non-trivial, due to the momentum interpolation needed [44–46]. After Rhie-Chow introduced the momentum interpolation (see [47]), Shen et al. showed the method was not time consistent due to pressure oscillation for smaller time steps [44]. From these studies it is clear that time integration on *static* collocated grids for the incompressible Navier-Stokes is non-trivial. Having *moving* grids, does add an additional time varying component in the already non-trivial methods for momentum interpolation for incompressible flow on collocated grids. Therefore, time consistent FSI with incompressible flow on moving collocated grids is discussed in detail in Chapter 2.

1.3.2. RADIAL BASIS FUNCTION MESH DEFORMATION

Mesh deformation is an important part for any FS(C)I model, since stability, accuracy and efficiency are all influenced by the mesh quality. Radial Basis Function (RBF) mesh deformation has been gaining interest due to robust and accurate (i.e. high mesh quality) results [48]. Since RBF mesh deformation shows potential as mesh deformation technique, applying it in a high fidelity Smart Rotor model would be of great interest.

However, most studies either show small test cases or only steady-state like results. Can RBF mesh deformation be properly applied to large scale unsteady calculations with local (flap) deformation, while being robust, efficient, accurate and preferably almost user-independent? How to use RBF mesh deformation efficiently is studied in Chapter 3.

1.3.3. MODEL VALIDATION AND APPLICATION

Experimental validation is an important step for any model, and maybe even more so for high fidelity models. The model should be validated over a range of parameters, known to influence the characteristics of the flow and flap behavior. At the same time a comparison with other models, especially widely used engineering models, will give insight in the differences of the models compared with experiments. Recently, an extensive experimental campaign has been performed for an airfoil with oscillating flap under different conditions and with different actuation properties [49]. With this experimental data, a thorough validation can be performed for both the high fidelity model and an engineering model, as discussed in Chapter 4. With the validated model, the final two compo-

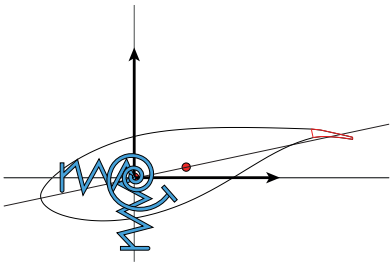

Figure 1.5: Illustration of the coupling of the structural model of a 2D airfoil. Three springs are present and an deformable trailing edge flap, highlighted at the back of the airfoil.

nents are added to the high fidelity model: structural coupling (illustrated in Figure 1.5)

and flap controller. With the completed 2D high fidelity model a more extensive study can be performed on accuracy compared to an engineering model, when coupling to the structure is taken into account and the flap is controlled instead of making a prescribed motion. Since engineering models are often used within the feasibility and design studies of the smart rotor, comparing the high fidelity model will give insight for which conditions a high fidelity and engineering model differ the most, giving an indication where future research should focus on when validating and further developing both the high fidelity and the engineering models. This part is presented in Chapter 5.

REFERENCES

[1] N. Fichaux, J. Beurskens, P. H. Jensen, and J. Wilkes, *Upwind*, Tech. Rep. March (2011).

[2] T. Barlas and G. van Kuik, *Review of state of the art in smart rotor control research for wind turbines,* Progress in Aerospace Sciences **46**, 1 (2010).

[3] T. Buhl, M. Gaunaa, and C. Bak, *Potential Load Reduction Using Airfoils with Variable Trailing Edge Geometry,* Journal of Solar Energy Engineering **127**, 503 (2005).

[4] L. Librescu, S. Na, P. Marzocca, C. Chung, and M. K. Kwak, *Active aeroelastic control of 2-D wing-flap systems operating in an incompressible flowfield,* in *44th AIAA/ASME/ASCE/AHS Structures, Structural Dynamics, and Materials Conference*, Vol. April (Norfolk, Virginia, USA, 2003) pp. 1–11.

[5] S. Basualdo, *Load alleviation on wind turbine blades using variable airfoil geometry,* Wind Engineering **29**, 169 (2005).

[6] C. Bak, M. Gaunaa, P. B. Andersen, T. Buhl, P. Hansen, and K. Clemmensen, *Wind Tunnel Test on Wind Turbine Airfoil with Adaptive Trailing Edge Geometry,* in *45th AIAA Aerospace Sciences Meeting and Exhibit*, January (Reno, Nevada, USA, 2007) pp. 1–12.

[7] C. Bak, M. Gaunaa, P. B. Andersen, T. Buhl, P. Hansen, and K. Clemmensen, *Wind tunnel test on airfoil Risø-B1-18 with an Active Trailing Edge Flap,* Wind Energy **13**, 207 (2010).

[8] P. Baek, J.-G. Jérémiaz, P. Kramer, and M. Gaunaa, *Experimental Comparison of a Miniflap to a Trailing Edge Flap on a Wind Turbine Airfoil,* EWEA conference , 1 (2011).

[9] P. B. Andersen, L. Henriksen, M. Gaunaa, C. Bak, and T. Buhl, *Deformable trailing edge flaps for modern megawatt wind turbine controllers using strain gauge sensors,* Wind Energy **13**, 193 (2010).

[10] T. Barlas, W. van Wingerden, A. Hulskamp, G. M. van Kuik, and H. N. Bersee, *Smart dynamic rotor control using active flaps on a small-scale wind turbine: aeroelastic modeling and comparison with wind tunnel measurements,* Wind Energy **17**, n/a (2012).

[11] S. Navalkar, J. van Wingerden, E. van Solingen, T. Oomen, and G. van Kuik, *Subspace Predictive Repetitive Control for wind turbine load alleviation using trailing edge flaps,* 2014 American Control Conference , 4422 (2014).

[12] J. Smit, L. O. Bernhammer, S. T. Navalkar, L. Bergami, and M. Gaunaa, *Sizing and control of trailing edge flaps on a smart rotor for maximum power generation in low fatigue wind regimes,* Wind Energy **17**, n/a (2015).

[13] M. A. Lackner and G. A. M. van Kuik, *A comparison of smart rotor control approaches using trailing edge fl aps and individual pitch control,* Wind Energy **13**, 117 (2010).

[14] X. Shen, X. Zhu, and Z. Du, *Wind turbine aerodynamics and loads control in wind shear flow,* Energy **36**, 1424 (2011).

[15] S. T. Navalkar, J. W. van Wingerden, and G. A. M. van Kuik, *Individual blade pitch for yaw control,* Journal of Physics: Conference Series **524**, 1 (2014).

[16] G. Bir and J. Jonkman, *Aeroelastic Instabilities of Large Offshore and Onshore Wind Turbines,* Journal of Physics: Conference Series **75**, 1 (2007).

[17] M. O. L. Hansen and H. Aagaard Madsen, *Review Paper on Wind Turbine Aerodynamics,* Journal of Fluids Engineering **133**, 114001 (2011).

[18] M. O. L. Hansen, J. A. Michelsen, and N. N. Sørensen, *Navier-Stokes solver for rotating wing,* in *Conference proceedings of 5th European Wind Energy Association conference and exhibition* (Thessaloniki, Greece, 1994) pp. 557–561.

[19] M. O. L. Hansen, J. N. Sorensen, J. A. Michelsen, and N. N. Sorensen, *A Global Navier-Stokes rotor prediction model,* The 35th Aerospace Sciences Meeting & Exhibit **AIAA-97-09** (1997).

[20] N. N. Sørensen and M. O. L. Hansen, *Rotor performance predictions using a Navier-Stokes method,* AIAA Journal , 52 (1998).

[21] E. Duque, C. van Dam, and S. Hughes, *Navier-Stokes simulations of the NREL Combined Experiment Phase II rotor,* 37th Aerospace Sciences Meeting and Exhibit , 143 (1999).

[22] L. Fingersh, D. Simmds, M. Hand, D. Jager, J. Cotrell, M. Robinson, S. Schreck, and S. Larwood, *Wind Tunnel Testing of NREL's Unsteady Aerodynamics Experiment,* in *Proc. of the AIAA 39th Aerospace Sciences Meeting & Exhibit* (2001).

[23] N. N. Sørensen, J. A. Michelsen, and S. Schreck, *Navier-Stokes predictions of the NREL phase VI rotor in the NASA Ames 80 ft x 120 ft wind tunnel,* Wind Energy **5**, 151 (2002).

[24] N. Sørensen, B. A., and Z. F., *3D CFD computations of transitional flows using DES and a correlation based transition model,* Wind Energy **14**, 77 (2011).

[25] Y. Li, K.-J. Paik, T. Xing, and P. M. Carrica, *Dynamic overset CFD simulations of wind turbine aerodynamics,* Renewable Energy **37**, 285 (2012).

[26] T. Rinehart, S. Medida, T. Kalra, and J. D. Baeder, *RANS Simulations of Sandia 100 - m Wind Turbine Blade: Effect of Leading-Edge Tubercles,* 32nd AIAA Applied Aerodynamics Conference , 1 (2014).

[27] T. Guo, Z. Lu, D. Tang, T. Wang, and L. Dong, *A CFD/CSD model for aeroelastic calculations of large-scale wind turbines,* Science China Technological Sciences **56**, 205 (2013).

[28] D. O. Yu and O. J. Kwon, *A Coupled CFD-CSD Method for Predicting HAWT Rotor Blade Performance,* 51st AIAA Aerospace Sciences Meeting , 1 (2013).

[29] F. Zahle, C. Bak, and N. Troldborg, *Comprehensive Aerodynamic Analysis of a 10 MW Wind Turbine Rotor Using 3D CFD,* 32nd ASME Wind Energy Symposium , 1 (2014).

[30] M. Carrión, R. Steijl, M. Woodgate, G. Barakos, X. Munduate, and S. Gomez-Iradi, *Aeroelastic analysis of wind turbines using a tightly coupled CFD–CSD method,* Journal of Fluids and Structures **50**, 392 (2014).

[31] Y. Bazilevs, M.-C. Hsu, I. Akkerman, S. Wright, K. Takizawa, B. Henicke, T. Spielman, and T. E. Tezduyar, *3D simulation ofwind turbine rotors at full scale. Part I:Geometry modeling and aerodynamics,* International Journal for Numerical Methods in Fluids **65**, 207 (2011).

[32] Y. Bazilevs, M.-C. Hsu, J. Kiendle, R. Wuchner, and K.-U. Bletzinger, *3D simulation ofwind turbine rotors at full scale. Part II: Fluid–structure interaction modeling with composite blades,* International Journal for Numerical Methods in Fluids **65**, 236 (2011).

[33] M.-C. Hsu and Y. Bazilevs, *Fluid–structure interaction modeling of wind turbines: simulating the full machine,* Computational Mechanics **50**, 821 (2012).

[34] J. Jonkman, S. Butterfield, W. Musial, and G. Scott, *Definition of a 5-MW reference wind turbine for offshore system development*, Tech. Rep. February (2009).

[35] D. O. Yu and O. J. Kwon, *Time-accurate aeroelastic simulations of a wind turbine in yaw and shear using a coupled CFD-CSD method,* Journal of Physics: Conference Series **524**, 012046 (2014).

[36] F. Shakib, D. Corson, D. Griffith, and T. Ashwill, *Investigating Aeroelastic Performance of Multi-Megawatt Wind Turbine Rotors Using CFD,* The 8th UK Altair Technology Conference , 1 (2013).

[37] J. Heinz, N. N. Sørensen, and F. Zahle, *Investigation of the load reduction potential of two trailing edge flap controls using CFD,* Wind Energy **14**, 449 (2011).

[38] T. Wolff, J. R. Seume, and L. U. Hannover, *Numerical investigation of an airfoil with morphing trailing edge,* in *Proceedings of the 10th PhD Seminar on Wind Energy in Europe* (Orleans, France, 2014) pp. 114–118.

[39] E. Jost, A. Fischer, T. Lutz, and E. Krämer, *CFD studies of a 10 MW wind turbine equipped with active trailing edge flaps,* in *Proceedings of the 10th PhD Seminar on Wind Energy in Europe*, October (Orleans, France, 2014) pp. 119–122.

[40] Avatar, *AVATAR (AdVanced Aerodynamic Tools of lArge Rotors),* (2015).

[41] Innwind, *Innovative wind conversion systems (10-20mw) for offshore applications (innwind.eu),* (2015).

[42] A. H. van Zuijlen, A. D. Boer, and H. Bijl, *Higher Order Time Integration for 3D Fluid-Structure Interaction,* in *International Conference on Computational Methods for Coupled Problems in Science and Engineering*, edited by M. Papadrakakis, E. Onate, and B. Schrefler (Barcelona, 2007) pp. 1–4.

[43] H. Jasak, *Department of Mechanical Engineering Imperial College of Science, Technology and Medicine*, Ph.D. thesis, Imperial College of Science, Technology and Medicine (1996).

[44] W. Z. Shen, J. A. Michelsen, and J. N. Sørensen, *Improved Rhie-Chow interpolation for unsteady flow computations,* AIAA Journal **39**, 2406 (2001).

[45] B. Yu, W.-Q. Tao, J.-J. Wei, Y. Kawaguchi, T. Tagawa, and H. Ozoe, *Discussion on momentum interpolation method for collocated grids of incompressible flow,* Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology **42**, 141 (2002).

[46] B. Yu, Y. Kawaguchi, W.-Q. Tao, and H. Ozoe, *Checkerboard pressure predictions due to the underrelaxation factor and time step size for a nonstaggered grid with momentum interpolation method,* Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology **41**, 85 (2002).

[47] C. Rhie and W. Chow, *A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation,* in *3rd Joint Thermophysics, Fluids, Plasma and Heat Transfer Conference* (1982) pp. 1–12.

[48] A. de Boer, M. S. van der Schoot, and H. Bijl, *Mesh deformation based on radial basis function interpolation,* Computers and Structures **85**, 784 (2007).

[49] C. J. Simão Ferreira, A. G. Salcedo, and T. Gillebaart, *Unsteady measurements of the DU95W180 airfoil with oscillating flap,* Wind Energy **to be subm** (2016).

2

TIME CONSISTENT FLUID STRUCTURE INTERACTION ON COLLOCATED GRIDS FOR INCOMPRESSIBLE FLOW

My aim is not to be consistent with my previous statements on a given question, but to be consistent with truth as it may present itself to me at a given moment. The result has been that I have grown from truth to truth.

Mahatma Gandhi

Computational time is a large drawback of any high-fidelity model. Ensuring that the methods used are efficient will reduce this drawback. Time integration is one of the components where an increase in efficiency can be achieved. The first step would be to ensure consistent time integration. Consistent time integration on collocated grids for incompressible flow has been studied for static grids using the PISO method, in which the independencies on time-step size and under-relaxation has been studied in detail. A step by step analysis of a time consistent fluid-structure interaction (FSI) method for incompressible flow on collocated grids is presented. The method consist of: face normal and area correction for moving grids, treatment of velocity boundary conditions for no-slip walls, time integration of structure equations and fluid force interpolation to structure. Third order FSI is demonstrated, showing an increase in efficiency for higher order methods. Finally, the proposed method of time consistent FSI on collocated grids for incompressible flows is demonstrated by applying it to a three-dimensional flow over an elastic structure in a channel and the cylinder flap FSI benchmark case of Turek and Hron.

Parts of this chapter have been published in Gillebaart et al., Time Consistent Fluid Structure Interaction on Collocated Grids for Incompressible Flow, Computer Methods in Applied Mechanics and Engineering (2016), 298, pp. 159-182, http://dx.doi.org/10.1016/j.cma.2015.09.025

2.1. INTRODUCTION

The application of high fidelity models for aero-elastic analysis has been growing over the past years, also in the Wind Energy world. In many cases partitioned Computational Fluid Dynamics (CFD) - Computational Structural Dynamics (CSD) coupling is used. Even though computer power continues to increase, efficient FSI methods are required to increase the applicability of high fidelity models. One of the main contributors to the cost of a FSI computation is the unsteady nature of the physics, resulting in a time resolved simulation. The solution to limit the number of time steps is to use time consistent (higher order) methods. Time consistency ensures that for a decreasing time step the error decreases with the order of the discretization scheme used. Potentially this leads to the use of a larger time step combined with a higher order scheme, resulting in a reduction of the computational time.

For flow simulations on moving grids, it has been shown that the Discrete Geometric Conservation Law (DGCL) needs to be satisfied to prevent errors in the form of artificial mass sources and to preserve the non-linear stability properties of the temporal discretization scheme [1, 2]. The DGCL ensures that a uniform flow remains uniform when the grid is moving/deforming. Farhat and Geuzaine also showed that only satisfying the DGCL does not ensure consistent order behavior [2]. Depending on the chosen model and discretization technique additional effort is needed for time consistency on moving grids (and for FSI). For compressible flows time consistent FSI has been shown for second order schemes [3, 4]. More recently, higher order FSI by using implicit/explicit Runge-Kutta (IMEX) time integration schemes has been shown [5, 6]. However, to our knowledge time consistent FSI on collocated grids for incompressible flow has not yet been presented, mainly due to the difficulties of time consistency on moving grids for collocated (unstructured) grids.

In this thesis the widely applied finite volume formulation of the incompressible Navier-Stokes are used. A collocated grid approach is chosen, because of its flexibility of applying it to both structured and unstructured grids. To solve the Navier-Stokes equations an iterative Pressure Implicit with Splitting of Operators (PISO) algorithm is used [7], which requires a momentum interpolation scheme, on which many studies have been performed [8–10]. The original interpolation from Rhie and Chow (see [11]) did not ensure time consistent behavior due to pressure oscillation for smaller time steps as shown by Shen et al. in 2001 [8]. Additionally, Yu et al. showed that some of the proposed interpolation schemes are still time-step dependent and/or under-relaxation factor dependent [9]. They proposed a new set of momentum interpolation schemes, ensuring the solution to be independent of under-relaxation and time step (for steady state). Recently, a study has shown time consistency for unsteady aerodynamics on static grids for collocated grids using different momentum interpolation algorithms [12]. Higher order ESDIRK schemes have been applied to this discretization method on static grids by Kazemi-Kamyab et al. [13]. However, time consistency on moving grids for incompressible flows on collocated grids has only been shown by Tuković and Jasak [14]. They have shown time consistency for unsteady aerodynamic on moving collocated grids [14] by using Yu et al. [9] their approach, although they did not consider solid moving boundaries, which are required for FSI applications. Additionally, a clear and detailed description, and demonstration of time consistent FSI for incompressible flows on collocated

grids is missing.

The goal of this chapter is to describe and show consistent time-order behavior for a FSI solver for collocated grids using the segregated (momentum interpolation) approach for the incompressible Navier-Stokes equations. The main focus is on time consistency on moving collocated grids for incompressible flow, while consistent time integration for the used structural models is considered to be well known. For the Navier-Stokes equations the backward differencing (BDF) schemes are considered for time integration. First, second and third order methods are used. The key ingredient of ensuring consistent order behavior is to have the correct face velocities used for the pressure equation. To ensure a clear description and verification is split up in 5 sections: the PISO algorithm in Arbitrary Lagrangian Eulerian (ALE) form, time consistency on static grids (based on Yu et al [9] and Tuković and Jasak [14]), time consistency on moving grids, moving wall boundary conditions and time consistency for Fluid-Structure-Interaction. For static and moving grids a verification of the given description will be given by means of an academic test case. Two structural models are considered: a non-linear 3 degrees of freedom rigid body, and a non-linear elastic solid. In the results section the academic test case is used to show the influence of the moving wall boundary conditions, time integration and fluid force coupling of the structure, mesh topology and under-relaxation. Finally, a 3-dimensional flow over an elastic beam in a channel [15] and the fixed cylinder with an attached flexible flap FSI benchmark [16] are used to illustrate the applicability for realistic cases of the described method.

2.2. METHODS

The time consistent fluid-structure-interaction model consists of the following two parts: consistent fluid dynamics equations on moving grids and consistent structural equations for fluid coupling. Before an in depth analysis of the time consistent Navier-Stokes equations on moving grids is given, time consistent fluid dynamics on static grids is discussed. After the discussion on the fluid model equations, the time consistency for structural models is discussed with a focus on the application of coupling these equations to the fluid equations.

2.2.1. NAVIER-STOKES EQUATIONS ON COLLOCATED GRIDS USING PISO

For fluid simulations the Navier-Stokes equations (NS) are used, which can be approximated in different forms: Reynolds Averaged Navier Stokes (RANS), Detached Eddy Simulation (DES), Large Eddy Simulation (LES) or Direct Numerical Simulation (DNS). Since unstructured meshes (or hybrid of structured and unstructured) are preferred for their flexibility in meshing, collocated grids are considered. Solving the NS equations is done by means of the PISO algorithm [7], using momentum interpolation described by Yu et al. [9]. A finite volume discretization for unstructured grids is used, as presented by Jasak [17] and later summarized by Tuković and Jasak [14]. For deforming domains, the Arbitrary Eulerian Lagrangian (ALE) formulation of the NS equations is used (since the grid has to move with the structural deformation). The ALE formulation of the mass conservation and momentum equation of the incompressible Navier-Stokes equations per

control volume is:

$$\int_{V_C} (\nabla \cdot \mathbf{u})\, dV = 0, \tag{2.1}$$

$$\frac{\partial}{\partial t}\int_{V_C} \mathbf{u} dV + \oint_{S_C} \mathbf{n}\cdot(\mathbf{u}-\mathbf{u}_m)\mathbf{u} dS - \int_{V_C} \nabla\cdot(\nu\nabla\mathbf{u}) dV = -\int_{V_C} \frac{\nabla p}{\rho} dV. \tag{2.2}$$

Here, V_c is the cell volume, $\mathbf{u}$ the velocity vector, S_c the cell surface area, $\mathbf{n}$ the cell surface normal vector, $\mathbf{u}_m$ the mesh velocity vector, ν the kinematic viscosity, p the pressure and ρ is the density. For the PISO algorithm all terms on the left hand side are discretized, resulting in the discretized momentum equation:

$$\begin{aligned} \frac{1}{\Delta t}\sum_{k=n+1}^{n+1-p} c^k\left(\mathbf{u}^k V^k\right) + \sum_f \left(\phi-\phi_m\right)\mathbf{u}_f^{n+1} + \sum_f S_f \mathbf{n}_f \cdot \left(\nu\nabla\mathbf{u}^{n+1}\right)_f &= -\int_{V_C}\frac{\nabla p}{\rho} dV \\ &= -\nabla\left(p/\rho\right)V \\ &= -\sum_f S_f \left(p/\rho\right)_f \end{aligned} \tag{2.3}$$

where $\mathbf{u}^k$ is the velocity solution at a specific time step, c^k is its corresponding coefficient for the used discretization scheme, p is the order of the scheme and V^k is the cell volume at the specific time step. In the convective term, ϕ is the fluid flux at the face, ϕ_m is the mesh flux at the face, and $\mathbf{u}_f$ is the face velocity. S_f and $\mathbf{n}_f$ are the face area and face normal, respectively. This equation is solved for the velocity $\mathbf{u}^{n+1}$ with the latest solution of p/ρ at the r.h.s.. With this solution the Poisson equation for the pressure is formulated. Firstly, Equation 2.3 is divided by the cell volume and p/ρ is defined as $\tilde{p}$. Secondly, each of the terms of Equation 2.3 can be split into a diagonal term ($a\mathbf{u}^{n+1}$), its off-diagonal (neighboring) term ($B\mathbf{u}^{n+1}$) and its source terms $\mathbf{q}$:

$$a\mathbf{u}^{n+1} + B\mathbf{u}^{n+1} = \mathbf{q} - \left(\nabla\tilde{p}\right)^{n+1}. \tag{2.4}$$

Here a is the diagonal term of the control volume and B is the matrix containing the off-diagonal contributions of the discretization for the specific control volume. Finally, grouping the off diagonal matrix times the velocity solution ($B\mathbf{u}^{n+1}$) with the source term ($\mathbf{q}$) results in a compact form of the momentum equation:

$$a\mathbf{u}^{n+1} = \mathbf{H}\left(\mathbf{u}^{n+1}\right) - \left(\nabla\tilde{p}\right)^{n+1} \tag{2.5}$$

Here $\mathbf{H} = \mathbf{q} - B\mathbf{u}^{n+1}$ is the vector per control volume containing influence of the neighboring cells and the source terms. Often under-relaxation is used to solve the momentum equation (Equation (2.3)). Under-relaxation is performed by increasing the diagonal dominance:

$$\underbrace{(1/\alpha)\,a}_{\tilde{a}}\mathbf{u}_{m+1}^{n+1} = \mathbf{H}\left(\mathbf{u}_{m+1}^{n+1}\right) - \underbrace{(1-(1/\alpha))\,a}_{a^e}\mathbf{u}_m^{n+1} - \left(\nabla\tilde{p}\right)_m^{n+1}, \tag{2.6}$$

where, α is the under-relaxation factor, $\tilde{a}$ is the under-relaxed diagonal term, a^e is the explicit part of the diagonal term on the right hand side, subscript $m+1$ indicates the current iterative solution and subscript m indicated the previous iterative solution. In

further derivations the diagonal of the original set of equations (a) should be used and not the under-relaxed diagonal ($\tilde{a}$). The splitting into diagonal (a) and off-diagonal ($\mathbf{H}$) terms is done to obtain the velocity solution to the momentum equation without the pressure gradient term. This is needed to formulate the Poisson equation for the pressure, as explained in more detail Issa et al. [7]. $\mathbf{H}$ is obtained by using the latest velocity solution to Equation (2.3) $\mathbf{u}_{m+1}^{n+1}$. Rewriting Equation 2.5, such that an equation for $\mathbf{u}^{n+1}$ is obtained, results in:

$$\mathbf{u}^{n+1} = \frac{\mathbf{H}}{a} - \frac{\left(\nabla \tilde{p}\right)^{n+1}}{a} = \mathbf{u}_*^{n+1} - \frac{\left(\nabla \tilde{p}\right)^{n+1}}{a}. \tag{2.7}$$

Here $\mathbf{u}_*^{n+1} = \frac{\mathbf{H}}{a}$ is the velocity solution of the momentum equation without the pressure gradient influence. Taking the divergence of Equation 2.7 results in the continuity equation:

$$\nabla \cdot \left(\mathbf{u}^{n+1}\right) = \sum_f \left(\mathbf{u}^{n+1}\right)_f \cdot \mathbf{n}_f^{n+1} S_f^{n+1} = \sum_f \phi^{n+1} = 0. \tag{2.8}$$

Here, $(\ldots)_f$ is used to indicate that the enclosed variable is required on the faces of the grid. The pressure equation is obtained by combining Equation 2.7 and 2.8:

$$\begin{aligned} \nabla \cdot \left(\frac{\left(\nabla \tilde{p}\right)^{n+1}}{a} \right) &= \sum_f \frac{\left(\nabla \tilde{p}\right)_f^{n+1} \cdot \mathbf{n}_f^{n+1} S_f^{n+1}}{(a)_f} \\ &= \nabla \cdot \left(\frac{\mathbf{H}}{a} \right) = \sum_f \left(\frac{\mathbf{H}}{a} \right)_f \cdot \mathbf{n}_f^{n+1} S_f^{n+1} = \sum_f \left(\mathbf{u}_*^{n+1}\right)_f \cdot \mathbf{n}_f^{n+1} S_f^{n+1} = \sum_f \phi_*^{n+1} \end{aligned} \tag{2.9}$$

After the pressure is obtained from Equation 2.9, the final divergence free flux is calculated using the pressure flux from the Poisson Equation:

$$\phi^{n+1} = \phi_*^{n+1} - \frac{\left(\nabla \tilde{p}\right)_f^{n+1} \cdot \mathbf{n}_f^{n+1} S_f^{n+1}}{\overline{(a)}_f}, \tag{2.10}$$

where $\overline{(\ldots)}_f$ indicates a linear interpolation from the cell centers to the faces of the enclosed variable. Equations (2.3) and (2.9) are solved consecutively after which Equation (2.10) is used to update the velocity. This process is repeated until the pressure and velocity are converged, after which the solution is progressed to the next time step. A more detailed description of the discretization, formulation and solving procedure of the equations can be found in [17]. For the right hand side of the pressure equation (Equation 2.9) the non-divergence free face fluxes (ϕ_*^{n+1}) are needed. These fluxes can be split into two parts:

$$\phi_*^{n+1} = \frac{\overline{(\mathbf{H}_r)}_f}{\overline{(a)}_f} \cdot \mathbf{n}_f^{n+1} S_f^{n+1} + \frac{(\mathbf{H}^o)_f}{\overline{(a)}_f} \cdot \mathbf{n}_f^{n+1} S_f^{n+1} = \phi_r + \phi^o, \tag{2.11}$$

where the part depending on $\mathbf{u}^{n+1}$ and source terms is indicated by ϕ_r, and the part depending on the velocities from the previous time steps, which originates from the temporal term, is indicated by ϕ^o. This last part is where the temporal discretization scheme

comes into play and must be constructed depending on the chosen scheme. Depending on the time discretization of the temporal term in the momentum equation, $\mathbf{H}^o$ contains a set of velocity solutions from previous times. A general formulation of this term is given for static and moving grids in the next sections.

2.2.2. STATIC GRIDS

For static grids there is no change in mesh geometry, resulting in cell volumes, surface area and surface normals to be time independent. This simplifies the contribution of the $\mathbf{H}^o$ term to the fluxes. The general BDF form of $\mathbf{H}^o$ for static grids is:

$$\left(\mathbf{H}^o\right)_f = \frac{1}{\Delta t}\sum_{k=n}^{n+1-p} c^k \left(\mathbf{u}^k\right)_f, \tag{2.12}$$

with the superscript k as time step indicator, $\mathbf{u}^k$ is the velocity solution from a previous time step, c^k is its corresponding coefficient for the used discretization scheme and p is the order of the BDF scheme. In Equation 2.13, 2.14 and 2.15 $\mathbf{H}^o$ is given for first order, second and third order backward difference, respectively:

$$\left(\mathbf{H}^o\right)_f = \frac{(\mathbf{u}^n)_f}{\Delta t} \tag{2.13}$$

$$\left(\mathbf{H}^o\right)_f = 2\frac{(\mathbf{u}^n)_f}{\Delta t} - \frac{1}{2}\frac{\left(\mathbf{u}^{n-1}\right)_f}{\Delta t} \tag{2.14}$$

$$\left(\mathbf{H}^o\right)_f = 3\frac{(\mathbf{u}^n)_f}{\Delta t} - \frac{3}{2}\frac{\left(\mathbf{u}^{n-1}\right)_f}{\Delta t} + \frac{1}{3}\frac{\left(\mathbf{u}^{n-2}\right)_f}{\Delta t}. \tag{2.15}$$

Here, $\mathbf{u}^n$, $\mathbf{u}^{n-1}$ and $\mathbf{u}^{n-2}$ are the cell-centered velocity solution from the three previous time steps, respectively. Since $\mathbf{H}^o$ needs to be known on the faces, the velocities at the previous time steps need to be known on the faces. When velocities are linearly interpolated to the faces (indicated by $\overline{(\ldots)}_f$), consistent time order behavior is not present and the steady state solution is time step dependent [8]. Linearly interpolated velocities are not divergence free, since the fluxes are constructed based on the non-linear momentum interpolation (such as presented by Yu et al in 2002 [9]) to create a divergence free flux (see Equation 2.10). When constructing $\mathbf{H}^o$, these fluxes should be used to create a divergence free face velocity solution:

$$\begin{aligned}
\phi^o &= \frac{(\mathbf{H}^o)_f}{\overline{(A)}_f}\cdot\mathbf{n}_f S_f = \frac{1}{\overline{(A)}_f \Delta t}\sum_{k=n}^{n+1-p} c^k \left(\mathbf{u}^k\right)_f \cdot \mathbf{n}_f S_f \\
&= \frac{1}{\overline{(A)}_f \Delta t}\sum_{k=n}^{n+1-p} c^k \left[\left(\mathbf{u}^k_{\parallel}\right)_f + \left(\mathbf{u}^k_{\perp}\right)_f\right]\cdot\mathbf{n}_f S_f \\
&= \frac{1}{\overline{(A)}_f \Delta t}\sum_{k=n}^{n+1-p} c^k \left(\mathbf{u}^k_{\perp}\right)_f \cdot \mathbf{n}_f S_f = \frac{1}{\overline{(A)}_f \Delta t}\sum_{k=n}^{n+1-p} c^k \phi^k,
\end{aligned} \tag{2.16}$$

where $(\mathbf{u}_{\perp})_f$ is the face normal velocity and $\left(\mathbf{u}_{\parallel}\right)_f$ is the face tangential velocity. For static grids only the inner product of the normal face velocity with the face normal is required,

leading to the face flux ϕ^k (which is oriented normal to the face). The tangential velocity is not required for static grids, since the face normal is constant in time, causing the inner product between the face normal and the tangential component of the face velocity to be zero.

From the above solution, the definition of the normal face velocity is determined and given by:

$$\left(\mathbf{u}_{\perp}^{k}\right)_f = \frac{\phi^k}{S_f}\mathbf{n}_f. \tag{2.17}$$

Combining Equation 2.16, 2.11 and 2.9 the right hand side of the pressure equation is constructed.

VERIFICATION

For verification a circular cavity case is used. The domain is shown in Figure 2.1, where the outer circle has a unit radius and the inner circle a radius of 0.1 meter. An incompressible fluid is used with a kinematic viscosity (ν) of 1 m^2/s. The outer boundary has a rotational velocity of 10 rad/s, resulting in a velocity magnitude of 10 m/s. During the first 0.1 seconds the speed is smoothly increased from 0 to 10 rad/s to prevent any discontinuities in the initial conditions. For the pressure a zero gradient condition is used. On the inner boundary a slip condition is used for the velocity and a zero gradient condition for the pressure. Spatial discretization is done with central differencing on all terms, while sub-iterations are performed until a residual of 10^{-10} is reached. To

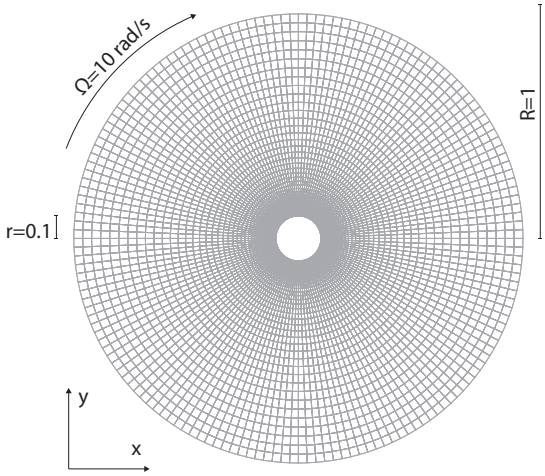

Figure 2.1: Circular cavity case and its structured mesh. 160 points are used in the circumference and 81 in the radial direction.

determine the time order the flow field at 0.2 seconds is taken, which is approximately at 50% of the transient part of this flow. Often the velocity field is used for demonstration of time consistency, however for fluid-structure-interaction the pressure will dominate the forces generally and thus it is crucial that in addition to the velocity also the pressure shows time consistent behavior. In Figure 2.2 the ∞-norm error of the pressure and

velocity fields are given for three backward differencing schemes. It must be noted that BDF1 and BDF2 are unconditionally stable, while the BDF3 scheme is only conditionally stable for implicit time integration.

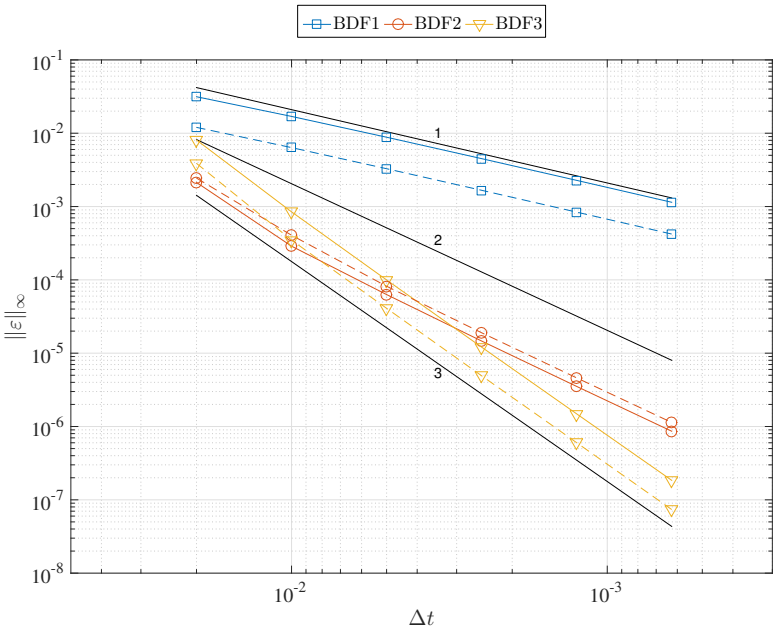

Figure 2.2: Pressure (−) and velocity (−−) error for static grid of circular cavity case. Three backward differencing schemes are shown, resulting in a 1^{st}, 2^{nd} and 3^{rd} order behavior of the pressure and velocity error. For all errors the fields of the BDF3 scheme with $\Delta t = 0.000125$ are used as reference fields.

2.2.3. MOVING GRIDS

For moving grids the geometry of the mesh is time dependent, resulting in time dependent cell volumes, face areas and face normals. Tuković and Jasak proposed a solution for the face velocities for the BDF2 scheme in their study [14]. However, a general formulation and detailed explanation is missing. The general BDF form of $\mathbf{H}^o$ for moving grids is:

$$\left(\mathbf{H}^o\right)_f = \frac{1}{\Delta t} \sum_{k=n}^{n+1-p} c^k \overline{\left(\frac{V^k}{V^{n+1}}\right)}_f \left(\mathbf{u}^k\right)_f . \tag{2.18}$$

Equation 2.19, 2.20 and 2.21 give $\mathbf{H}^o$ for first, second and third order backward difference, respectively.

$$\left(\mathbf{H}^o\right)_f = \overline{\left(\frac{V^n}{V^{n+1}}\right)}_f \frac{\left(\mathbf{u}^n\right)_f}{\Delta t} \tag{2.19}$$

$$\left(\mathbf{H}^o\right)_f = 2\overline{\left(\frac{V^n}{V^{n+1}}\right)}_f \frac{\left(\mathbf{u}^n\right)_f}{\Delta t} - \frac{1}{2}\overline{\left(\frac{V^{n-1}}{V^{n+1}}\right)}_f \frac{\left(\mathbf{u}^{n-1}\right)_f}{\Delta t} \tag{2.20}$$

$$\left(\mathbf{H}^o\right)_f = 3\overline{\left(\frac{V^n}{V^{n+1}}\right)}_f \frac{\left(\mathbf{u}^n\right)_f}{\Delta t} - \frac{3}{2}\overline{\left(\frac{V^{n-1}}{V^{n+1}}\right)}_f \frac{\left(\mathbf{u}^{n-1}\right)_f}{\Delta t} + \frac{1}{3}\overline{\left(\frac{V^{n-2}}{V^{n+1}}\right)}_f \frac{\left(\mathbf{u}^{n-2}\right)_f}{\Delta t} \tag{2.21}$$

For static grids, only the face flux is needed to create the divergence free face velocities. However, for moving grids the face normal, cell volume and face area are time dependent. The volume ratios follow automatically from the temporal term shown in Equation 2.3, while the time dependent normal and face area are used to formulate the face velocity. The normal component of the face velocity is similar to the one given in Equation 2.17. However, the face flux (ϕ^k) cannot be used to construct the tangential component, which is required for ϕ^0 for moving grids:

$$\begin{aligned}
\phi^o &= \frac{1}{\overline{(A)}_f \Delta t} \sum_{k=n}^{n+1-p} c^k \overline{\left(\frac{V^k}{V^{n+1}}\right)}_f \left(\mathbf{u}^k\right)_f \cdot \mathbf{n}_f^{n+1} S_f^{n+1} \\
&= \frac{1}{\overline{(A)}_f \Delta t} \sum_{k=n}^{n+1-p} c^k \overline{\left(\frac{V^k}{V^{n+1}}\right)}_f \overbrace{\left[\left(\mathbf{u}_\parallel^k\right)_f + \left(\mathbf{u}_\perp^k\right)_f\right] \cdot \mathbf{n}_f^{n+1} S_f^{n+1}}^{\tilde{\phi}^k} \\
&= \frac{1}{\overline{(A)}_f \Delta t} \sum_{k=n}^{n+1-p} c^k \overline{\left(\frac{V^k}{V^{n+1}}\right)}_f \tilde{\phi}^k.
\end{aligned} \tag{2.22}$$

For the tangential component the linear interpolated velocities are used, while for the normal component the face fluxes are used. The face velocity is constructed from the normal and tangential component for moving grids:

$$\left(\mathbf{u}^k\right)_f = \overbrace{\overline{\left(\mathbf{u}^k\right)}_f - \mathbf{n}_f^k \left(\overline{\left(\mathbf{u}^k\right)}_f \cdot \mathbf{n}_f^k\right)}^{\left(\mathbf{u}_\parallel^k\right)_f} + \overbrace{\frac{\phi^k}{S_f^k}\mathbf{n}_f^k}^{\left(\mathbf{u}_\perp^k\right)_f}. \tag{2.23}$$

Both the tangential component and the normal component are required to obtain the flux contribution from the previous velocity solutions on the grid at t^{n+1}. This is done by calculating the adjusted face flux for the previous time steps, denoted as $\tilde{\phi}^k$ in Equation 2.22.

For a better understanding of the influence of the time dependent face areas and face normals on the adjusted face flux ($\tilde{\phi}^k$), two particular mesh deformation cases are considered. Firstly, the case where only the face normals change and face areas (and cell volumes) remain constant, and secondly, the case where the face normals remain

constant and face areas (and cell volumes) change in time. From the general $\tilde{\phi}^k$ given in Equation 2.22, the $\tilde{\phi}^k$ for these two cases is derived. For the first case, where $\mathbf{n}_f^{n+1} \neq \mathbf{n}_f^k$, $\Delta\mathbf{n}_f^k = \mathbf{n}_f^{n+1} - \mathbf{n}_f^k$ and $S_f^{n+1} = S_f^k$, the flux $\tilde{\phi}^k$ becomes:

$$\tilde{\phi}^k = \left[\left(\mathbf{u}_\parallel^k\right)_f + \left(\mathbf{u}_\perp^k\right)_f\right] \cdot \left(\mathbf{n}_f^k + \Delta\mathbf{n}_f^k\right) S_f^{n+1} = \phi^k + \underbrace{\left(\mathbf{u}^k\right)_f \cdot \Delta\mathbf{n}_f^k S_f^{n+1}}_{\mathbf{n}_f \text{ correction}}. \tag{2.24}$$

Here, $\Delta\mathbf{n}_f^k$ is the change in face normal from t^k to t^{n+1}. From this analysis it can be seen that $\tilde{\phi}^k$ consists of the static grid component (ϕ^k) and a face normal correction.

For the second case a similar analysis is made as above. When $\mathbf{n}_f^{n+1} = \mathbf{n}_f^k$ and $S_f^{n+1} \neq S_f^k$, the flux $\tilde{\phi}^k$ becomes:

$$\tilde{\phi}^k = \left[\left(\mathbf{u}_\parallel^k\right)_f + \left(\mathbf{u}_\perp^k\right)_f\right] \cdot \mathbf{n}_f^{n+1} S_f^{n+1} = \frac{S_f^{n+1}}{S_f^k}\phi^k. \tag{2.25}$$

Only the fluxes of the previous time steps contribute to $\tilde{\phi}^k$, and no linear interpolation of the cell-centered velocities is needed in this particular case. The fluxes are adjusted based on the added flux due to the change in face area. From these two particular cases, the adjusted face flux $\tilde{\phi}^k$ can be written as two separate terms correcting the face flux for changes in normal and surface area:

$$\begin{aligned}
\tilde{\phi}^k &= \left[\left(\mathbf{u}_\parallel^k\right)_f + \left(\mathbf{u}_\perp^k\right)_f\right] \cdot \mathbf{n}_f^{n+1} S_f^{n+1} \\
&= \left(\mathbf{u}^k\right)_f \cdot \mathbf{n}_f^{n+1} S_f^{n+1} - \left(\mathbf{u}^k\right)_f \cdot \mathbf{n}_f^k S_f^{n+1} + \left(\mathbf{u}^k\right)_f \cdot \mathbf{n}_f^k S_f^{n+1} \\
&= \left(\mathbf{u}^k\right)_f \cdot \Delta\mathbf{n}_f^k S_f^{n+1} + \left(\mathbf{u}_\perp^k\right)_f \cdot \mathbf{n}_f^k S_f^{n+1} \\
&= \underbrace{\left(\mathbf{u}^k\right)_f \cdot \Delta\mathbf{n}_f^k S_f^{n+1}}_{\text{normal correction } t^k} + \underbrace{\frac{S_f^{n+1}}{S_f^k}\phi^k}_{\text{area correction } t^k}.
\end{aligned} \tag{2.26}$$

From this the grid correction form of ϕ^o follows:

$$\phi^o = \frac{1}{\overline{(A)}_f \Delta t} \sum_{k=n}^{n+1-p} c^k \overline{\left(\frac{V^k}{V^{n+1}}\right)}_f \left[\left(\mathbf{u}^k\right)_f \cdot \Delta\mathbf{n}_f^k S_f^{n+1} + \frac{S_f^{n+1}}{S_f^k}\phi^k\right]. \tag{2.27}$$

VERIFICATION

The same circular cavity case as for the static grids is used, but with the internal mesh moving. To show the validity of the analysis where $\tilde{\phi}^k$ can be seen as a surface normal correction and surface area correction, the mesh motion is constructed from two motions: rotational motion, where the area remains constant and the normals change, and a radial expanding motion, where the area changes, but the normal remains constant. The motion is described by:

$$\theta = \frac{A_\theta}{2}\left(1-\cos\left(\pi f t\right)\right) \tag{2.28}$$

$$r_{0.55} = \frac{A_r}{2}\left(1-\cos\left(4\pi f t\right)\right)\sin\left(2\pi f t\right), \tag{2.29}$$

with $f = 5Hz$, $A_\theta = \pi/9$ and $A_r = 0.05m$. Here θ is the angle and $r_{0.55}$ is the radial motion at a radius of 0.55 (or halfway between inner and outer circle). The radial motion is interpolated using Radial Basis Functions (RBF) mesh deformation with the Thin Plate Spline (TPS) [18]. The remainder of the case is exactly the same as described for the static grid. Before the three different schemes are shown, the results for only rotational or expanding motion are shown for $\tilde{\phi}^k$ consisting of only normal correction, only area correction or both corrections. In Figures 2.3 and 2.4 the order behavior is shown for the two cases. For the rotational case (shown in Figure 2.3) consistent order behavior is

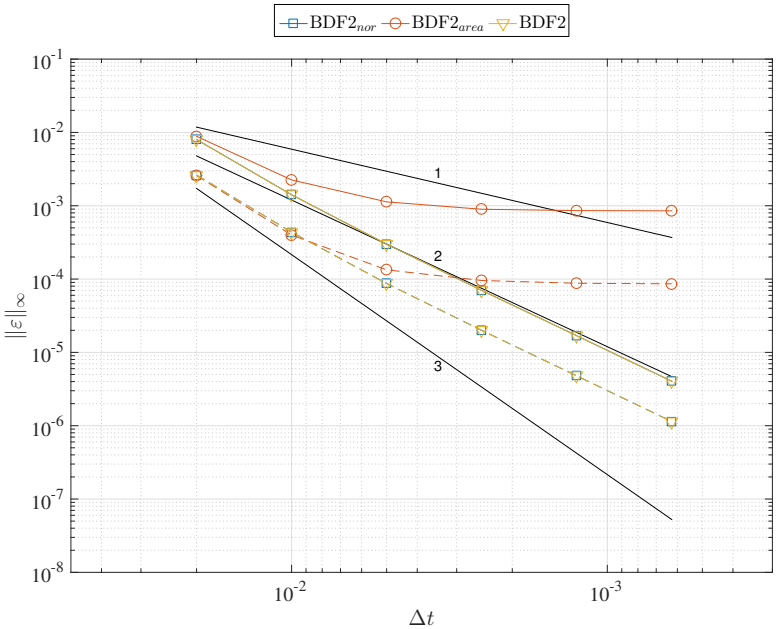

Figure 2.3: Pressure (−) and velocity (−−) error for rotating mesh of circular cavity case. Three different formulas for $\tilde{\phi}^k$ are used: only rotational correction, only expanding correction and both corrections. For all errors the solution of the BDF2 scheme with $\Delta t = 0.000125$ is used as reference solution.

found for the full correction form and the normal correction form of $\tilde{\phi}^k$. In this case the area correction is equal to the static grid form since the face areas do not vary in time. Figure 2.3 shows clearly that normal correction is crucial for time consistent behavior for changing face normals. A similar analysis is made for the expanding case: correct order behavior is found for the full correction form and the area correction form of $\tilde{\phi}^k$. In this particular case the normal correction form is equal to the static grid approach. Again it is clear that a correction for the changing area surface is crucial for consistent

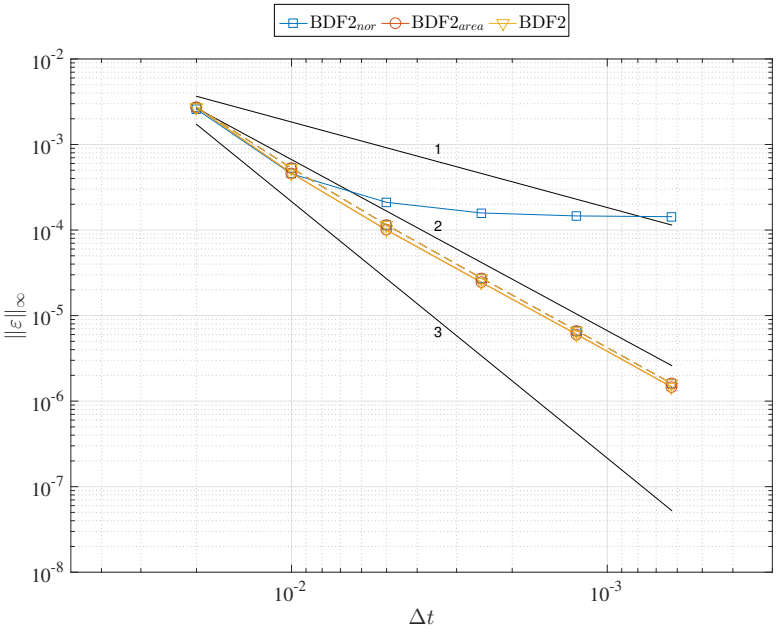

Figure 2.4: Pressure (−) and velocity (−−) error for expanding mesh of circular cavity case. Three different formulas for $\tilde{\phi}^k$ are used: only rotational correction, only expanding correction and both corrections. For all errors the solution of the BDF2 scheme with $\Delta t = 0.000125$ is used as reference solution.

behavior on expanding meshes. Only when $\tilde{\phi}^k$ consists of both correction terms, time consistent behavior is found in both cases. To demonstrate the validity and generality of the proposed solution (stated in Equation 2.27 and 2.22), the time order for 3 backward different schemes is given in Figure 2.5, where the mesh motion is both rotating and expanding.

2.2.4. BOUNDARY CONDITIONS

In addition to using the correct face fluxes in the pressure equation, the velocity must be calculated accordingly on moving boundaries for the momentum equation. A similar approach as for the face velocities is used to calculate the velocity on the boundaries: splitting the velocity into a normal and tangential component [19]. The normal component is derived from the mesh fluxes which follow from the Discrete Geometric Conservation Law (DGCL) [1]. From the DGCL the mesh fluxes (ϕ_m) are derived based on the swept volumes:

$$\phi_m = \sum_{k=n+1}^{n+2-p} d^k(c) \frac{(\Delta V)_f^k}{\Delta t}. \tag{2.30}$$

Here ΔV_f is the swept volume of a face and $d^k(c)$ are the DGCL coefficient, which are a function of the time discretization scheme coefficients c^k. In Equations 2.31, 2.32 and

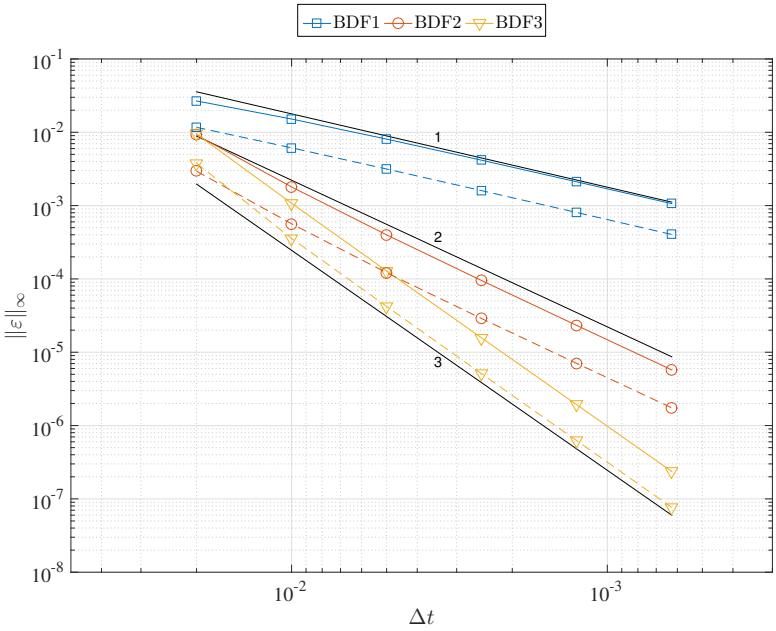

Figure 2.5: Pressure (−) and velocity (−−) error for moving mesh of circular cavity case. Three backward differencing schemes are shown, resulting in a 1^{st}, 2^{nd} and 3^{rd} order behavior of the pressure and velocity error. For all errors the solution of the BDF3 scheme with $\Delta t = 0.000125$ is used as reference solution.

2.33 the fluxes are given for the BDF1, BDF2 and BDF3 scheme, respectively.

$$\phi_m = \frac{\Delta V_f^{n+1}}{\Delta t} \tag{2.31}$$

$$\phi_m = \frac{3}{2}\frac{\Delta V_f^{n+1}}{\Delta t} - \frac{1}{2}\frac{\Delta V_f^{n}}{\Delta t} \tag{2.32}$$

$$\phi_m = \frac{11}{6}\frac{\Delta V_f^{n+1}}{\Delta t} - \frac{7}{6}\frac{\Delta V_f^{n}}{\Delta t} + \frac{1}{3}\frac{\Delta V_f^{n-1}}{\Delta t} \tag{2.33}$$

The DGCL and the related mesh fluxes are time discretization scheme dependent as can be seen in the generalized equation in 2.30. From this mesh flux the normal component of the wall velocity ($(\mathbf{u}_\perp)_{wall}$) must be computed to ensure that the DGCL is still satisfied. The normal component of the wall velocity is given by:

$$(\mathbf{u}_\perp)_{wall} = \frac{\phi_m}{S_{wall}}\mathbf{n}_{wall}. \tag{2.34}$$

This results in a different normal velocity when simply applying a time discretization scheme to the position in time. However, the tangential component ($(\mathbf{u}_\parallel)_{wall}$) cannot be derived from the flux (which is always normal to the surface), and is therefore derived from the face position in time ($\mathbf{x}_{wall}^k$) using the same temporal discretization scheme as

for the temporal term in the momentum equation. The wall velocity based on the same temporal discretization as for the temporal term in the momentum equation is:

$$\left(\mathbf{u}_{bdf}\right)_{wall} = \frac{1}{\Delta t}\sum_{k=n+1}^{n+1-p} c^k \mathbf{x}^k_{wall}. \tag{2.35}$$

Based on this velocity a tangential component of the wall velocity can be constructed. Combining Equations 2.34 and 2.35 results in the formula for the velocity boundary condition on moving walls:

$$\mathbf{u}_{wall} = (\mathbf{u}_{\perp})_{wall} + \underbrace{\left[\left(\mathbf{u}_{bdf}\right)_{wall} - \mathbf{n}_{wall}\left(\left(\mathbf{u}_{bdf}\right)_{wall}\cdot\mathbf{n}_{wall}\right)\right]}_{(\mathbf{u}_{\parallel})_{wall}}. \tag{2.36}$$

For the pressure a zero gradient condition is used on the (moving) wall. A more consistent implementation is suggested by Gresho et al. [20], which evaluates the momentum equation on the boundary to obtain the pressure boundary condition. However, this is a non-trivial task for unstructured collocated grids and is therefore left to be assessed in future studies. Therefore, in this study the often applied zero-gradient condition is used.

2.2.5. STRUCTURAL MODELS

In this section the two structural models that are used for the full FSI test cases are described. A 3 degrees of freedom rigid body is described, to be used with the circular cavity case. Secondly, the non-linear elastic equations for the beam case are described.

3 DEGREES OF FREEDOM RIGID BODY

A 3 Degrees of Freedom (DoF) rigid body used in this study is governed by the non-linear set of equations:

$$m\ddot{x}_s + c_x\dot{x}_s + k_x x_s = F_x + ml\dot{\theta}_s^2\cos(\theta_s) + ml\ddot{\theta}_s\sin(\theta_s) \tag{2.37}$$

$$m\ddot{y}_s + c_y\dot{y}_s + k_y y_s = F_y + ml\dot{\theta}_s^2\sin(\theta_s) - ml\ddot{\theta}_s\cos(\theta_s) \tag{2.38}$$

$$\left(I_{CG} + ml^2\right)\ddot{\theta}_s + c_\theta\dot{\theta}_s + k_\theta\theta_s = F_\theta + ml\ddot{x}_s\sin(\theta_s) - ml\ddot{y}_s\cos(\theta_s), \tag{2.39}$$

where m is the mass per unit depth, k is the spring stiffness in the respective degree of freedom, F is the aerodynamic force in the respective degree of freedom, F_θ is the counter clockwise positive aerodynamic moment around the rotational center, l is the distance between the center of gravity and the rotational center positive in the horizontal direction, θ_s is the pitch angle deformation and I_{CG} is the moment of inertia around the center of gravity. The values used are depicted in Table 2.1.

THREE-DIMENSIONAL BEAM MODEL

The configuration of the structural domain for the three-dimensional beam is described by the displacement $\boldsymbol{v}^s$. An elastic and compressible structure is assumed, and the governing equation is given by the balance of momentum:

$$\rho^s\frac{\partial \boldsymbol{v}^s}{\partial t} + \rho^s\left(\nabla\boldsymbol{v}^s\right)\boldsymbol{v}^s = \nabla\cdot\boldsymbol{\sigma}^s + \rho^s\boldsymbol{g} \qquad \text{in } \Omega^s, \tag{2.40}$$

Table 2.1: Structural properties for three degree of freedom rigid body model

c_x	0 N s/m	k_x	4 N/m
c_y	0 N s/m	k_y	4 N/m
c_θ	0 N s m/rad	k_θ	4 N m/rad
m (per unit depth)	1 kg/m	I_{CG}	0.05 kg m^2
RC	(0 0.1) m	l	0.01 m

with density ρ^s, stress tensor $\boldsymbol{\sigma}^s$ and body force $\boldsymbol{g}$. Equation (2.40) is modified to use the total Lagrangian description, i.e. with respect to the initial reference state Γ^s, resulting in

$$\rho^s \frac{\partial^2 \boldsymbol{v}^s}{\partial t^2} = \nabla \cdot \left(J \boldsymbol{\sigma}^s \boldsymbol{F}^{-T} \right) + \rho^s \boldsymbol{g} \qquad \text{in } \Omega^s, \tag{2.41}$$

where the deformation gradient tensor $\boldsymbol{F}$ is defined as $\boldsymbol{F} = \boldsymbol{I} + \nabla \boldsymbol{u}^s$, and the Jacobian J is the determinant of the deformation gradient tensor $\boldsymbol{F}$. By applying the constitutive law for the St. Venant-Kirchhoff material, the Cauchy stress tensor $\boldsymbol{\sigma}^s$ is found by applying

$$\boldsymbol{\sigma}^s = \frac{1}{J} \boldsymbol{F} \left(\lambda^s (\operatorname{tr} \boldsymbol{E}) \boldsymbol{I} + 2\mu^s \boldsymbol{E} \right) \boldsymbol{F}^T, \tag{2.42}$$

with $\boldsymbol{E} = \frac{1}{2} \left(\boldsymbol{F}^T \boldsymbol{F} - \boldsymbol{I} \right)$, and the shear modulus μ^s [21].

The structural domain is discretized with a second order finite volume method as described by Tuković and Jasak in [22]. Here the total Lagrangian description is used to model the structural domain instead of an updated Lagrangian approach as discussed in [22]. The second derivative in time of the displacement is discretized via first or second order backward differencing. Note that Tuković and Jasak also use a time discretization based on backward differencing for the structural domain.

2.2.6. FLUID-STRUCTURE INTERACTION

In this section the coupling of the fluid and the structure is discussed. First, the partitioned fluid-structure interaction approach is discussed, after which a more detailed discussion is given on force coupling for Runge-Kutta time integration scheme.

PARTITIONED FLUID-STRUCTURE INTERACTION

Partitioned fluid-structure interaction is considered, due to the fact that highly specialized solvers have already been developed for fluid as well as structure simulations. It is desired to reuse these applications for fluid-structure interaction computations where separate fluid and structure solvers need to be coupled.

With regard to the fluid-structure interaction problem, the fluid and solid solver are considered as black boxes. Hence, only input and output information is accessible.

Thus, at each time step the response of the fluid solver F_f is defined as

$$\boldsymbol{y} = F_f(\boldsymbol{x}), \tag{2.43}$$

where $\boldsymbol{x}$ denotes the displacement of the fluid-structure interface, and $\boldsymbol{y}$ denotes the force acting on the fluid-structure interface. The response of the solid solver F_s is

$$\boldsymbol{x} = F_s\left(\boldsymbol{y}\right). \tag{2.44}$$

Typically, at every time step the fixed point equation

$$\boldsymbol{x} = F_s \circ F_f\left(\boldsymbol{x}\right) \tag{2.45}$$

must be satisfied, which can be written as the FSI interface residual $\boldsymbol{R}$

$$\boldsymbol{R}\left(\boldsymbol{x}\right) = F_s \circ F_f\left(\boldsymbol{x}\right) - \boldsymbol{x}. \tag{2.46}$$

At the fluid-structure interface Γ^{fs}, the balance of stresses (i.e. dynamic boundary condition) is enforced through

$$\boldsymbol{\sigma}^f \boldsymbol{n} = \boldsymbol{\sigma}^s \boldsymbol{n} \qquad \text{on } \Gamma^{fs}, \tag{2.47}$$

with the unit vector $\boldsymbol{n}$ normal to the fluid-structure interface Γ^{fs}, and the stress tensors $\boldsymbol{\sigma}^f$ and $\boldsymbol{\sigma}^s$. Also, at the fluid-structure interface the velocities $\boldsymbol{v}^f$ and $\boldsymbol{v}^s$ must be equal (i.e. kinematic boundary condition):

$$\boldsymbol{v}^f = \boldsymbol{v}^s \qquad \text{on } \Gamma^{fs}. \tag{2.48}$$

The partitioning error is minimized at every time step by performing multiple coupling iterations, i.e. the fluid and solid solvers are called multiple times within each time step. Standard approaches used to solve the strongly coupled fluid-structure interaction problem are the Gauss-Seidel method [23], fixed under-relaxation [24], Aitken under-relaxation [25], and the IQN-ILS method [26, 27].

TIME CONSISTENT PARTITIONED FLUID-STRUCTURE INTERACTION

Time consistency for fluid-structure-interaction is obtained by ensuring a time consistent fluid discretization, time consistent structural discretization and the correct force coupling. The first two parts have been described, the latter in this section and should be treated with care. The best and most straightforward approach is to use the same time discretization scheme for both the fluid and structure, in which the coupling of the forces and displacement is easy. However, often BDF schemes are used for the fluid equations, while the structural equations are often solved by Runge-Kutta schemes. Coupling the fluid and structure solvers consistently with these two different time integration schemes is not trivial, although consistent time integration is still possible. This is illustrated with the often applied explicit Runge-Kutta 4 (RK4) time discretization scheme [28], which is a multi-stage method. The general form of the explicit Runge-Kutta method is:

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \sum_{i=1}^{s} \Delta t b^i \mathbf{k}^i \tag{2.49}$$

$$\begin{aligned} \mathbf{k}^i &= \mathbf{f}^i\left(t^n + c^i\Delta t, \mathbf{x}^n + \Delta t \sum_{j=1}^{i-1} a^{ij}\mathbf{k}^j\right) \\ &= \mathbf{f}^i_{fluid}\left(t^n + c^i\Delta t\right) + \mathbf{f}^i_{structure}\left(t^n + c^i\Delta t, \mathbf{x}^n + \Delta t \sum_{j=1}^{i-1} a^{ij}\mathbf{k}^j\right). \end{aligned} \tag{2.50}$$

Here, $\mathbf{x}^{n+1}$ is the solution at t^{n+1}, $\mathbf{x}^n$ is the solution at t^n, b^i is the stage coefficient for stage i, k^i is the stage solution at $t^n + c^i \Delta t$, c^i is the state time coefficient for stage i, a^{ij} is the stage coefficient of stage j for stage i, $\mathbf{f}^i_{fluid}$ is the fluid influence for stage i and $\mathbf{f}^i_{structure}$ is the structure influence for stage i. In addition, the sum of a^{ij} over j is equal to its corresponding time coefficient c^i. Special attention should be payed to the fluid forces ($\mathbf{f}^i_{fluid}$), which should be known at $t^n + c^i \Delta t$ according to the scheme in Equation 2.50. However, for Fluid-Structure-Interaction these forces come from the fluid solver, which can have a different time discretization scheme and are therefore not available at these specific times and a different choice is necessary for each of the stage times. Directly taking the solution from the fluid equations at t^{n+1} for all stage times results in a first order approximation:

$$\mathbf{f}^i_{fluid}\left(t^n + c^i \Delta t\right) \approx \mathbf{f}^i_{fluid}\left(t^{n+1}\right) = \mathbf{F}^{n+1}_{fluid} \qquad \text{for} \quad i = 1..s \tag{2.51}$$

Using a mid-point rule for the fluid force:

$$\mathbf{f}^i_{fluid}\left(t^n + c^i \Delta t\right) \approx \mathbf{f}^i_{fluid}\left(t^{n+1/2}\right) = \frac{\mathbf{F}^{n+1}_{fluid} + \mathbf{F}^n_{fluid}}{2} \qquad \text{for} \quad i = 1..s, \tag{2.52}$$

and using this for each stage (from $i = 1$ to $i = s$) results in a second order approximation. Finally, a linear interpolation to the stage time $t^n + c^i \Delta t$ is proposed:

$$\mathbf{f}^i_{fluid}\left(t^n + c^i \Delta t\right) \approx \overline{\left(\mathbf{f}^i_{fluid}\right)}_{t^n + c^i \Delta t} = c^i \mathbf{F}^{n+1}_{fluid} + \left(1 - c^i\right) \mathbf{F}^n_{fluid} \quad \text{for} \qquad i = 1..s, \tag{2.53}$$

from which is expected to behave similar to the mid-point interpolation in Equation 2.52. With a similar approach 3^{rd} of 4^{th} order can reached by using additional time solutions (e.g. $\mathbf{F}^{n-1}_{fluid}$) in combination with higher order interpolation of the forces (e.g. cubic interpolation).

2.3. RESULTS

To show the functionality of the model described above, two cases are used. First the cylinder cavity will be used to verify time consistency for a forced (prescribed) motion and for a coupled motion (FSI). In addition, the FSI case is performed with an unstructured mesh and under-relaxation for the fluid to show the independence of the time integration scheme on the mesh and under-relaxation. Secondly, the often studied/applied 3D flexible beam in a channel is used to show the consistency of the method in a more realistic engineering application [15]. Finally, the FSI-3 benchmark of Turek and Hron [16] is simulated to assess the efficiency of the time consistent method.

2.3.1. FORCED MOTION AND FLUID-STRUCTURE-INTERACTION

Generally meshes are moved due to a moving body. This can be done either by coupling it to a structural model (and performing FSI) or by prescribing a motion (without coupling it to a structural model). Both these cases are shown here for the cavity case: one with prescribed motion and one coupled to a 3 degrees of freedom structure. In section 2.2.3 time consistency for moving meshes (without moving body) has been shown.

The major difference with a moving body case is the boundary condition on the moving boundary (wall), which should be as Equation 2.36 described in section 2.2.4. To verify the proposed boundary condition and show the importance of the correct application of it the circular cavity case is used. The motion of the inner cylinder is described by

$$y = \frac{A_x}{2}\left(1-\cos\left(\pi f t\right)\right)\sin\left(2\pi f t\right) \tag{2.54}$$

$$\theta = \frac{A_\theta}{2}\left(1-\cos\left(\pi f t\right)\right)\sin\left(2\pi f t\right), \tag{2.55}$$

with $A_x = 0.3m$, $A_\theta = 20^o$ and $f = 2.5Hz$. RBF interpolation is used to deform the internal mesh, while the outer cylinder is static in time. In Figure 2.6 the pressure and velocity errors for the three backward difference schemes are shown at the same end time ($0.2s$) as for the previous cases. In addition results are shown when the tangential velocity at

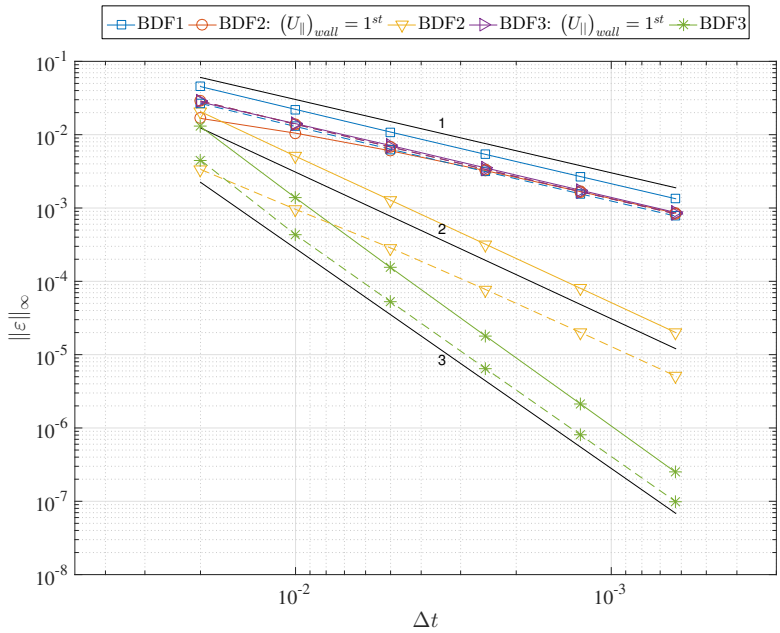

Figure 2.6: Pressure (−) and velocity (−−) error for moving body mesh of circular cavity case. Three backward differencing schemes are shown, resulting in a 1^{st}, 2^{nd} and 3^{rd} order behavior of the pressure and velocity error. In addition, the effect of estimating the tangential boundary velocity with a first order scheme is shown for BDF2 and BDF3. For all errors the solution of the BDF3 scheme with $\Delta t = 0.000125$ is used as reference solution and the end time is $0.2s$.

the wall is approximated with a first order approach (by only taking a BDF1 scheme on the face center position). The normal velocity is still derived from the DGCL, since otherwise mass conservation is not satisfied. Here the importance of the correct boundary condition becomes apparent, since both BDF2 and BDF3 reduce to first order.

Next the inner cylinder is coupled to the 3 DoF rigid body model described in section 2.2.5. This is done by using Aitken's under-relaxation with an initial relaxation factor of

0.7 and resetting this every new time step. The fluid and solid equations are solved until a residual lower than $1e^{-12}$ is reached. Sub-iterations are performed until the coupling residual is below $1e^{-8}$. In Figure 2.7 the influence of the choice of force interpolation is illustrated by means of the the order on the rotation (θ). The three force interpolations discussed in Equations 2.51, 2.52 and 2.53 are used. In addition BDF2 and BDF3 time integration is also used for the structure. Although the RK4 scheme can reduce to 1^{st}

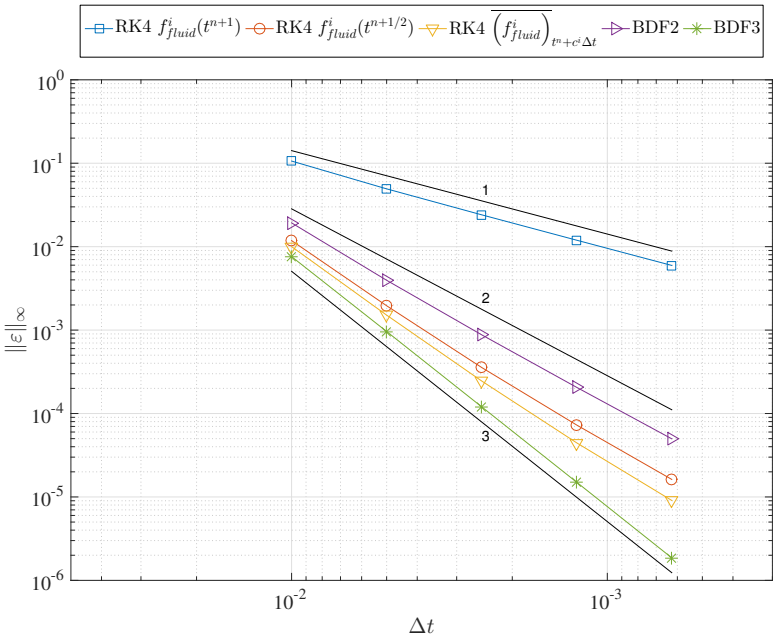

Figure 2.7: Rotation error for 3 DoF body of circular cavity case. Three different force interpolations are shown used for the Runge-Kutta 4 scheme, resulting in a 1^{st} and 2^{nd} order behavior of the displacement error. In all cases the fluid equations are discretized with BDF3 and the displacement for this FSI case is 3^{rd} order if the structure is also discretized with BDF3. For all errors the solution of the BDF3 scheme with $\Delta t = 0.000125$ is used as reference solution and the end time is $0.2s$.

order in FSI cases, by carefully choosing the force interpolation, 2^{nd} order FSI on is still achieved. Additionally, a lower error is obtained when the force is linearly interpolated at the stage time (see Equation 2.53) based on the forces at t^n and t^{n+1} instead of mid-point interpolation (at $t^{n+1/2}$) (see Equation 2.52). Instead of the RK4 scheme also the backward differencing schemes can be used for the time discretization of the structure. Choosing the same scheme as for the fluid discretization together with the appropriate boundary conditions ensures a time order consistent FSI method for collocated grids, as shown in Figure 2.8.

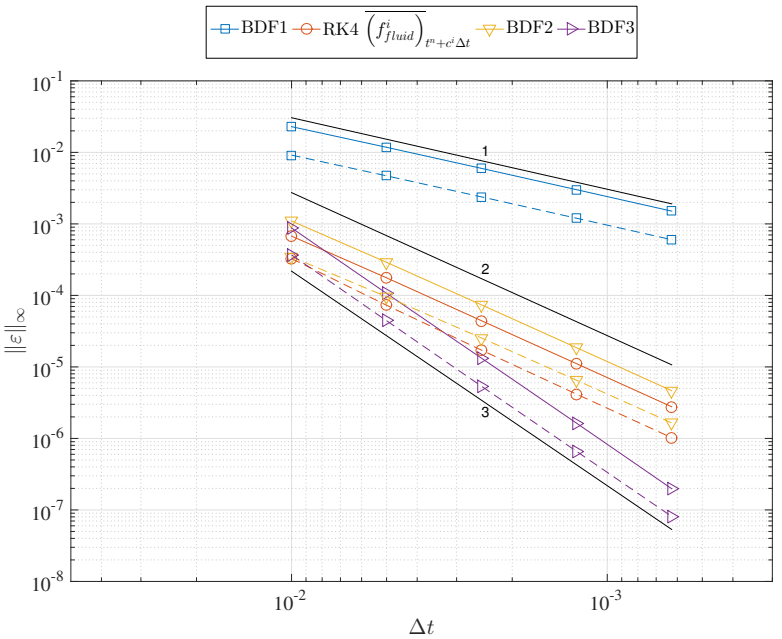

Figure 2.8: Pressure (−) and velocity (−−) error for 3 DoF body of circular cavity case. Fluid and structure are discretized with the three backward differencing schemes. Additionally, for the 2^{nd} order fluid discretization, also the Runge-Kutta 4 time discretization for the structure with linearly interpolated stage time force is shown. For all errors the solution of the BDF3 scheme with $\Delta t = 0.000125$ is used as reference solution and the end time is $0.2s$.

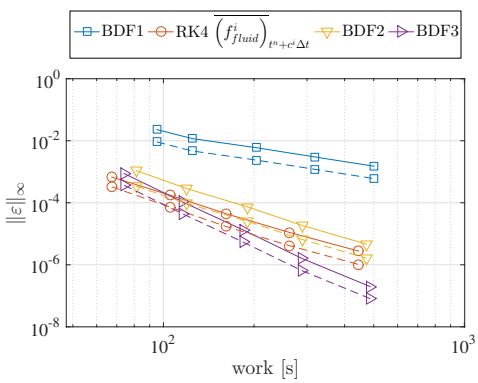

Figure 2.9: Work (in seconds) versus pressure (−) and velocity (−−) error for the 3DoF body in a circular cavity

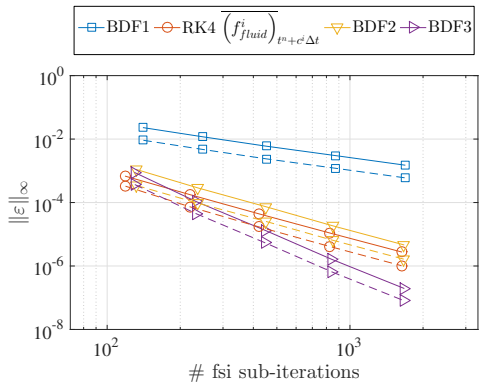

Figure 2.10: Total number of FSI Aitken's iterations versus pressure (−) and velocity (−−) error for the 3DoF body in a circular cavity

With the time consistent FSI method, the efficiency of the different presented schemes is studied. The accuracy results presented in Figure 2.8 are plotted against the computational time and against the number FSI iterations as shown in Figure 2.9 and Figure 2.10,

respectively.

In Figure 2.9 and Figure 2.10 a similar trend as in Figure 2.8 is found. As a results, a higher order consistent FSI time integration is significantly more efficient in both computational time and number of FSI sub-iterations. However, the benefit depends on the desired accuracy as can be seen in Figure 2.9. For an equal amount of work a higher accuracy can be obtained by using a higher order scheme, or a larger time step can be used for the same accuracy. It should be noted that this problem is not an inherent strongly coupled problem. In addition, the computational time is also highly dependent on the coupling algorithm, tolerance selection for fluid, solid and coupling equations, parallelization, computer power. When considering the results in Figure 2.9 and Figure 2.10, the relative differences should be considered, while the absolute values are merely an indication of the order of magnitude.

2.3.2. UNSTRUCTURED MESH

A unstructured mesh (see Figure 2.11) is combined with the same structural model as explained in section 2.2.5. In Figure 2.12 the pressure and velocity errors are shown for the BDF3 scheme for both the unstructured and structured mesh. The mesh topology has no influence on the time order, supporting the general structure of the time consistent formulation on moving grids.

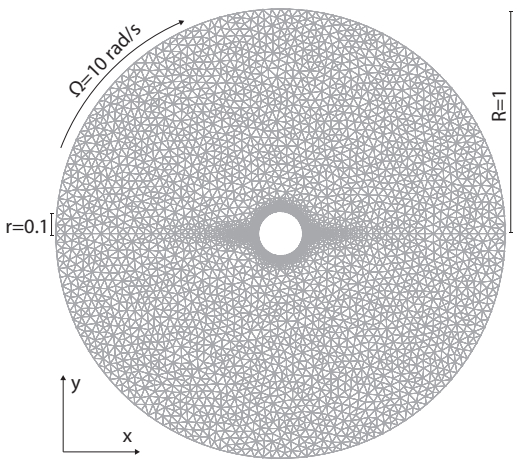

Figure 2.11: Unstructured mesh of circular cavity case

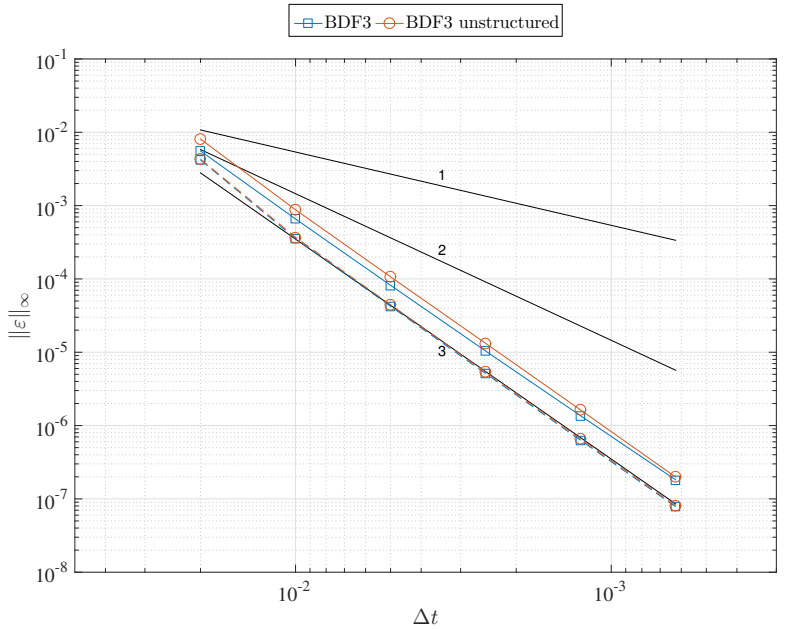

Figure 2.12: Pressure (−) and velocity (−−) error for 3 DoF body of circular cavity case with a structured and unstructured mesh. Fluid and structure are discretized with the BDF3 scheme. For both cases the solution of the BDF3 scheme with $\Delta t = 0.000125$ is used as reference solution and the end time is $0.2s$.

2.3.3. FLUID EQUATIONS UNDER-RELAXATION

All results shown until now are performed without under-relaxation for the fluid equations. However, under-relaxation is often used to stabilize the segregated set of non-linear equations of the PISO method. Previous studies have shown potential problems with under-relaxation for the PISO method [9]. Therefore, the influence of the under-relaxation of the fluid equations is assessed in this section. For the pressure, the pressure update is under-relaxed, before it is used in the next PISO loop. For the velocity, the momentum equation matrix is under-relaxed. This is done as shown in Equation 2.6. When the diagonal (A in Equation 2.5) is used to construct the non-divergence free face fluxes, it should be the original diagonal without under-relaxation (A) and not the under-relaxed diagonal ($\tilde{A}$). The same holds for H, which should contain the original source terms, without the diagonal terms added due to under-relaxation ($A^e\mathbf{u}_m^{n+1}$). When this is done properly, time consistent FSI is achieved as shown in Figure 2.13. It can be seen that the BDF3 results for both with and without under-relaxation provide the same solution to the academic case.

2.3.4. THREE-DIMENSIONAL FLOW OVER AN ELASTIC STRUCTURE

A three dimensional incompressible flow is considered over a flexible beam fixed to a wall, discussed in [15]. The problem is considered to be symmetric in the x/y plane. Thus, the simulation is only performed in one half of the domain. The width and height of the computational domain are 0.4 m, where the length is 1.5 m. The structure consists

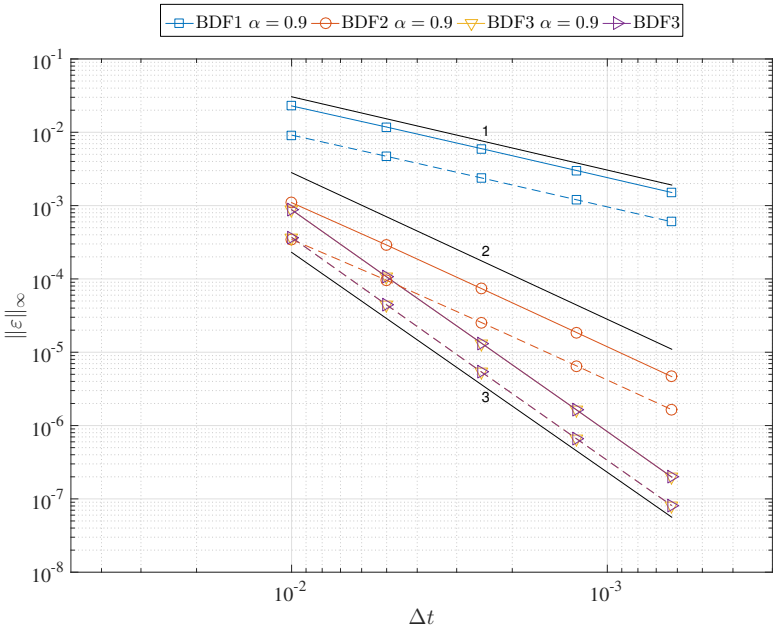

Figure 2.13: Pressure (−) and velocity (−−) error for 3 DoF body of circular cavity case with under-relaxation for the pressure and velocity. Fluid and structure are discretized with the three backward differencing schemes. Additionally, for the 2^{nd} order fluid discretization, also the Runge-Kutta 4 time discretization for the structure with linear interpolated stage time force is shown. For all errors the solution of the BDF3 scheme with Δt = 0.000125 is used as reference solution and the end time is 0.2s.

of a rectangular block with a width and height of 0.2 m, and a length of 0.1 m. The mesh of the flow domain is shown in Figure 2.14. At the inflow boundary a parabolic velocity profile is imposed as a Dirichlet boundary condition with peak velocity $v_{max} = 0.3\,\mathrm{m/s}$. On the outflow boundary the zero gradient condition is imposed on the velocity and the pressure has a Dirichlet boundary condition with zero pressure. A no-slip condition is used used on the remaining boundaries. The incompressible fluid has a density $\rho = 10^3\,\mathrm{kg/m^3}$, and kinematic viscosity $\nu = 10^{-3}\,\mathrm{m^2/s}$. The density of the solid is $10^3\,\mathrm{kg/m^3}$, the Poisson ratio $\nu^s = 0.4$ and the Young's modulus E is set to $10^4\,\mathrm{N/m^2}$, which is a smaller value as in [15] resulting in a larger displacement of the beam. Radial basis function interpolation is used to deform the fluid mesh. The IQN-ILS method [26, 27] is used to perform the coupling iterations on the FSI interface until the L2-norm of the FSI residual (2.46) normalized by the L2-norm of the interface displacement is smaller or equal to 10^{-9}. Matching grids are used at the FSI interface. From $t = 0\,\mathrm{s}$ until $t = 5\,\mathrm{s}$, the solid is assumed to be fixed. Thereafter, a starting procedure is applied to use a smooth increase of the applied pressure and viscous force on the solid in time as

$$F(t) = \begin{cases} 0 & 0 < t <= 5 \\ F\,[0.5 - 0.5\cos(\pi\,(t-5))] & 5 < t <= 6 \\ F & t > 6 \end{cases}, \tag{2.56}$$

where F represents the applied force on the structure. Figure 2.15 shows the pressure

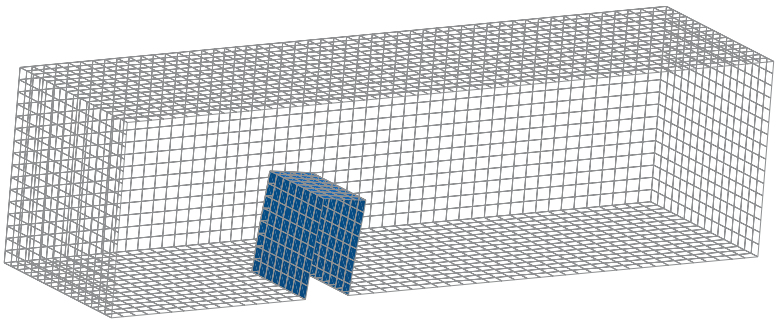

Figure 2.14: Three-dimensional flow over an elastic structure. Computational grid of the flow domain.

and velocity errors for the first and second order time integration schemes. The error is computed at $6 < t <= 8$ s for the velocity and pressure fields. The figures show that first and second order accuracy in time is achieved. Figure 2.18 shows the lift and drag over time for the different computations performed. As is to be expected, a large difference in lift and drag is observed when comparing the first and second order time integration schemes. This highlights the importance of using a second or possibly higher order time integration scheme for fluid-structure interaction simulations in order to accurately predict important parameters as lift and drag. The computational work (in seconds) for a similar error decreases for a higher order scheme, as shown in Figure 2.16. This effect diminishes for larger time steps, because of the increase in non-linearity between time steps. The number of FSI sub-iterations versus error supports this, because the total number of FSI iterations decreases for smaller time steps, as shown in Figure 2.17. In terms of efficiency, it is of crucial importance to have consistent time integration for FSI calculations.

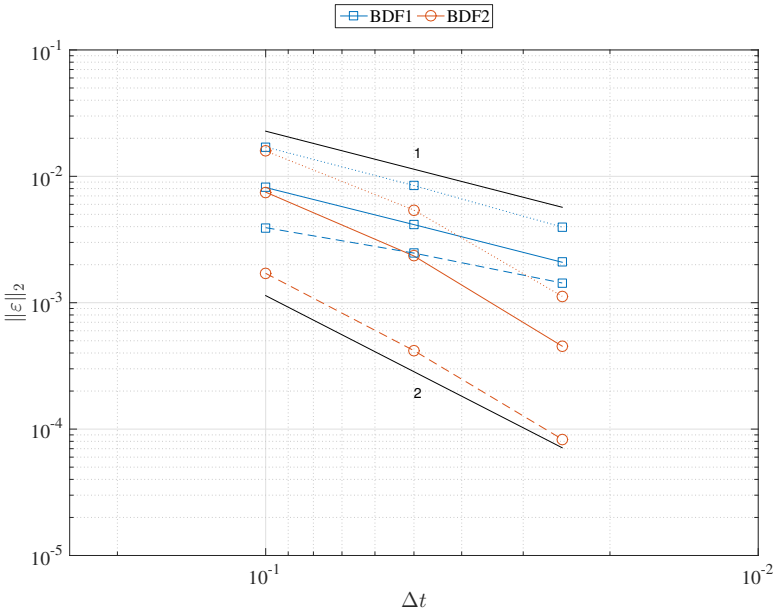

Figure 2.15: Pressure (−), velocity (−−) and displacement (···) error for flexible beam case. Fluid and structure are discretized with two backward differencing schemes: BDF1 and BDF2. For all errors the solution of the BDF2 scheme with $\Delta t = 0.0125$ is used as reference solution.

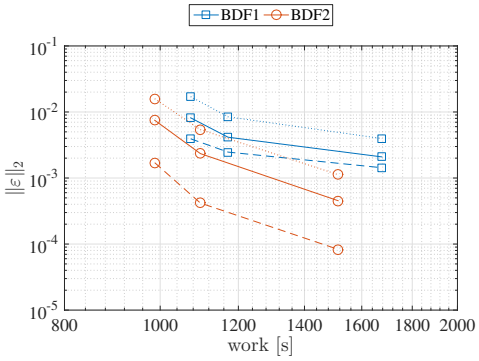

Figure 2.16: Work (in seconds) versus pressure (−), velocity (−−) and displacement (···) error for the flexible beam in a channel case.

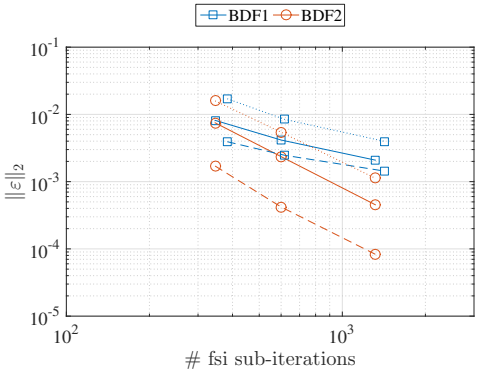

Figure 2.17: Total number of FSI IQN-ILS iterations versus pressure (−), velocity (−−) and displacement (···) error for the flexible beam in a channel case.

2.3.5. FIXED CYLINDER WITH AN ATTACHED FLEXIBLE FLAP FSI BENCHMARK

Finally, the fixed cylinder with an attached flexible flap FSI benchmark case as discussed in [16]. The FSI-3 case is taken, since it results in an inherent dynamic result. The prob-

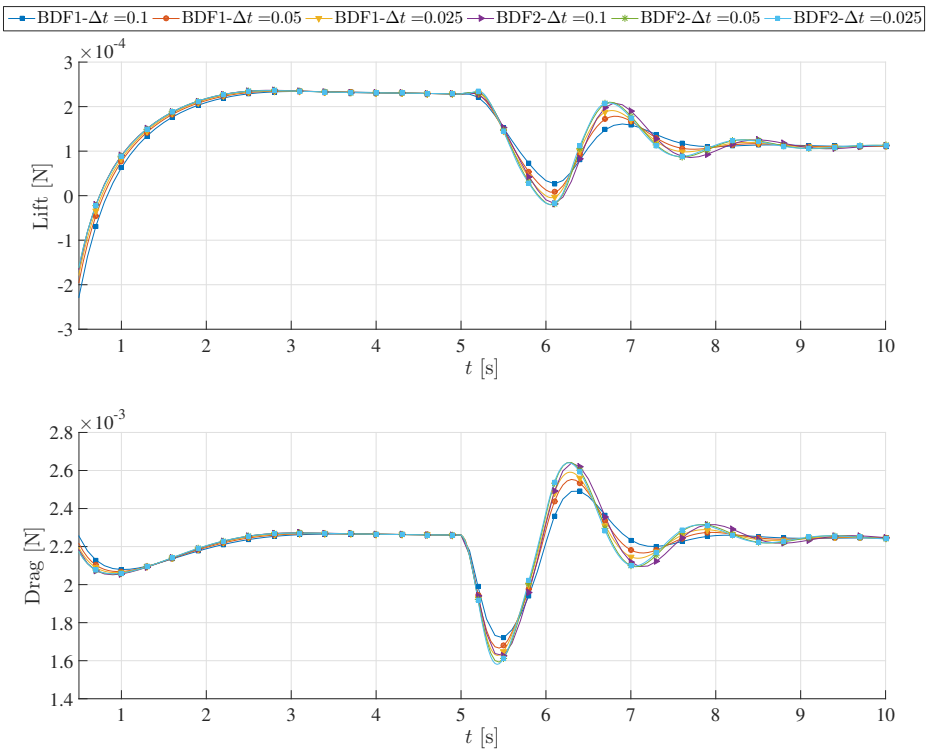

Figure 2.18: Lift (top) and drag (bottom) results for the flexible beam in a channel obtained with first and second order FSI time integration. Results for the time steps $\Delta t = [0.1, 0.05, 0.025]$ are shown.

lem is 2D, with the following dimensions: 2.5 m by 0.41 m. Part of the mesh of the flow and structural domain is shown in Figure 2.19, which consists out of 23924 fluid cells and 328 structural cells.

At the inflow boundary a parabolic velocity profile is imposed as a Dirichlet boundary condition with peak velocity $v_{max} = 3\,\text{m/s}$. On the outflow boundary the zero gradient condition is imposed on the velocity and the pressure has a Dirichlet boundary condition with zero pressure. A no-slip condition is used used on the remaining boundaries. The incompressible fluid has a density $\rho = 1\,\text{kg/m}^3$, and kinematic viscosity $\nu = 10^{-3}\,\text{m}^2/\text{s}$. For the structure the elastic beam model described in section 2.2.5 is used. The density of the solid is $1\,\text{kg/m}^3$, the Poisson ratio $\nu^s = 0.4$ and the Young's modulus E is set to $5.6 \cdot 10^{-6}\,\text{N/m}^2$. Radial basis function interpolation is used to deform the fluid mesh.

As for the beam case, the IQN-ILS method [26, 27] is used to perform the coupling iterations on the FSI interface until the L2-norm of the FSI residual (2.46) normalized by the L2-norm of the interface displacement is smaller or equal to 10^{-5}. Matching grids are used at the FSI interface.

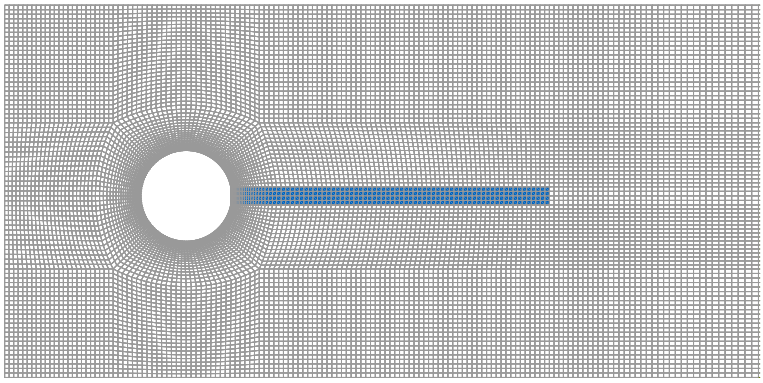

Figure 2.19: Cylinder flap FSI benchmark. Close up of computational grid of the flow and structural domain.

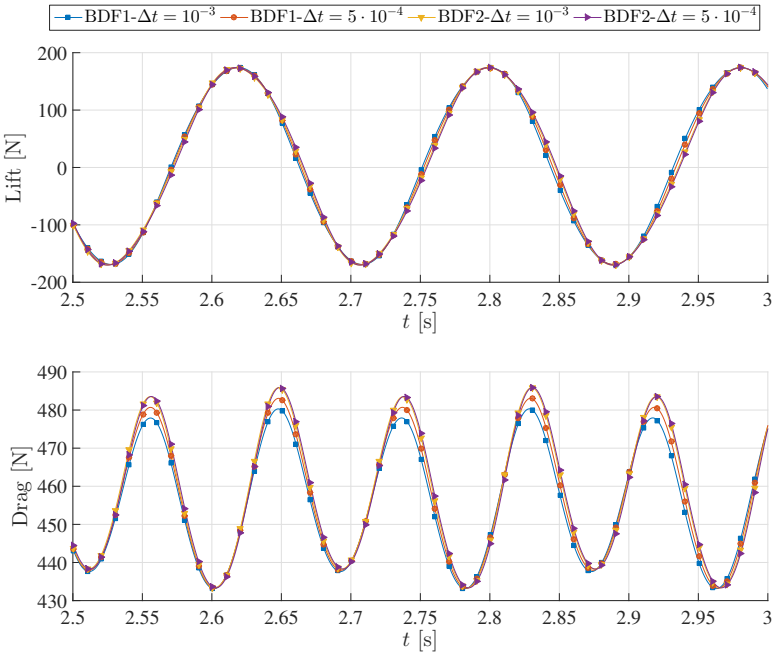

Figure 2.20: Lift (top) and drag (bottom) results for the cylinder flap case obtained with first and second order FSI time integration. Results for the time steps $\Delta t = [10^{-3}, 5 \cdot 10^{-4}]$ are shown.

Initially, the flow is developed without the structural model. This solution is imposed as initial condition for the FSI case. This simulation is ran for 3 s, where the first 2 seconds show influences of the transient phase. Figure 2.20 shows the periodic lift and drag over time (from 2.5 s to 3s) for two time steps (10^{-3}, $5 \cdot 10^{-4}$) and for two schemes: bdf1

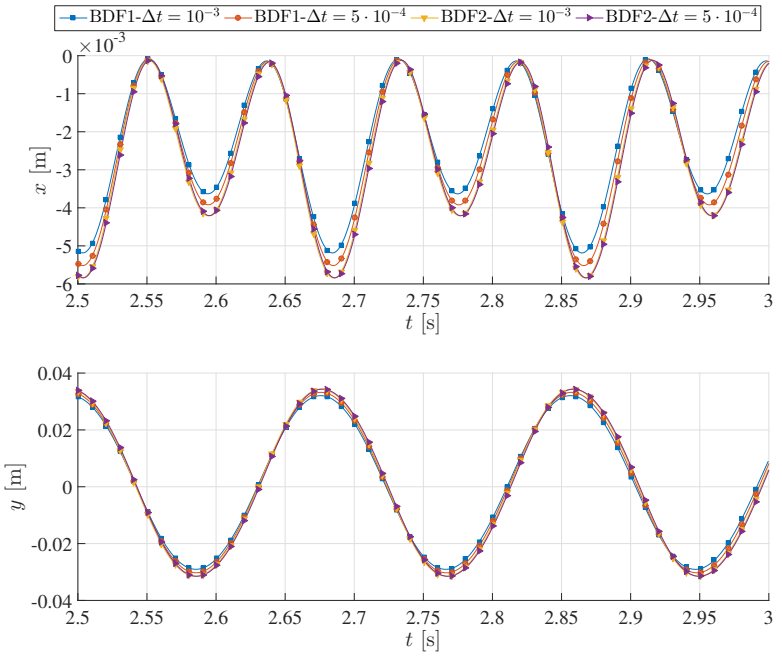

Figure 2.21: Tip displacement in x-direction (top) and y-direction (bottom) results for the cylinder flap case obtained with first and second order FSI time integration. Results for the time steps $\Delta t = [10^{-3}, 5 \cdot 10^{-4}]$ are shown.

and bdf2. For the same simulations the tip displacement is shown in Figure 2.21. As is to be expected, a large difference in lift, drag and displacement is observed when comparing the first and second order time integration schemes. This highlights the importance of using a second or possibly higher order time integration scheme for fluid-structure interaction simulations in order to accurately predict important parameters as lift and drag. When comparing the results to the reported results by Turek and Hron [16], a good correspondence is found for the bdf2 results. In addition, the efficiency is assessed by looking at the computational time, which is shown in Table 2.2. Here, it can be seen that the bdf2 schemes results in a significantly higher accuracy for less work than the bdf1 schemes. Again, the importance of a higher order scheme becomes apparent, since the computational work can be reduced by reducing the time step when using a higher order time integration scheme.

Table 2.2: Computational work in seconds and in average number of FSI sub-iterations per time step for the cylinder flap case [16].

	$\Delta t = 10^{-3}$		$\Delta t = 5 \cdot 10^{-4}$	
bdf1	29006 s	7.25 iter.	35423 s	7.00 iter.
bdf2	20918 s	6.92 iter.	30856 s	6.81 iter.

2.4. CONCLUSION & DISCUSSION

A step by step analysis of a high order time consistent FSI method for incompressible flow on collocated grids is presented in this Chapter. The method consist of: face normal and area correction for moving grids, treatment of velocity boundary conditions for no-slip walls and fluid force interpolation to the structure.

The crucial part for time consistency on moving grids is the fluxes of the previous time solutions. Either these should be incorporated correctly by constructing the face velocities from a normal and a tangential component, or by means of a face normal and face area correction on the flux. These corrections ensure correct order behavior. When not applied the solution might be less accurate but still usable depending on the required level of accuracy required. However, when appropriately done, consistent time order behavior is achieved on moving grids for the BDF1, BDF2 and BDF3 time integration schemes. From the detailed general description for the BDF time integration schemes, the consistent implementation of other (higher) order schemes can be derived for moving grids.

For moving bodies with a no-slip wall, the velocity should be constructed from the DGCL and a correct evaluation of the tangential velocity. The DGCL is used to construct the normal component, while the tangential velocity should be discretized with an equivalent scheme as for the temporal scheme in the momentum equation to ensure correct order behavior. It is shown that discretizing the tangential velocity with a lower order scheme reduces the order of the FSI time integration to that lower order.

When coupled to the structural solver, the forces should be (if needed) interpolated such that the structural equations obtain the same order as the fluid equations. For the backward difference scheme this is straightforward, since the force is needed at t^{n+1}, which follows directly from the fluid equations. However, for the often used explicit Runge-Kutta 4 scheme, this leads to a first order behavior. This happens because the force is needed at the stage time $(t^n + c^i \Delta t)$. It is shown that second order behavior can be obtained by linear interpolating the force to the stage time. It is also possible to obtain third order behavior with the RK4 scheme by interpolating the fluid force with a third order interpolation. In this chapter third order FSI is obtained by using the BDF3 scheme for both fluid and structure.

Additionally, the efficiency of higher order (BDF) time integration for FSI is assessed. For the academic circular cavity case it is found that the efficiency increases for higher order time integration. Both computational work and total number of FSI sub-iterations decrease for the same accuracy when increasing the order of the scheme. This is achieved by using a larger time step.

In addition to the academic test case, also for a non-academic case it is shown that the proposed FSI time integration for incompressible flows on collocated grids is consistent. This is done by means of a three dimensional flow over an elastic beam in a channel. Again, it is shown that the higher order schemes deliver a higher efficiency, both in terms of computational work and the number of FSI sub-iterations. For larger time steps, this effect diminishes, due to higher non-linearity between two time steps.

Finally, the Turek cylinder flap FSI benchmark case is simulated. The results stress the importance of having (higher order) consistent time integration for FSI simulations. The bdf1 results show large differences for the two different time steps, while the bdf2

results already show convergence at the used time steps. In addition, the computational time of the bdf2 time integration is lower than for the bdf1 time integration for the same time steps, resulting in a higher efficiency for the 2^{nd} order scheme.

Time consistent FSI on collocated grids provides a potential reduction in computational time, because a larger time step can be used for the same accuracy. The detailed description provides a basis for developing higher order FSI schemes for incompressible flows on collocated grids. Recently, higher order temporal discretization has been studies on static grids, which could now be studied on moving grids, resulting in a potentially even larger reduction of computational time.

REFERENCES

[1] I. Demirdžić and M. Perić, *Space conservation law in finite volume calculations of fluid flow,* International Journal for Numerical Methods in Fluids **8**, 1037 (1988).

[2] C. Farhat and P. Geuzaine, *Design and analysis of robust ALE time-integrators for the solution of unsteady flow problems on moving grids,* Computer Methods in Applied Mechanics and Engineering **193**, 4073 (2004).

[3] P. Geuzaine, C. Grandmont, and C. Farhat, *Design and analysis of ALE schemes with provable second-order time-accuracy for inviscid and viscous flow simulations,* Journal of Computational Physics **191**, 206 (2003).

[4] C. Farhat, *CFD on moving grids: from theory to realistic flutter, maneuvering, and multidisciplinary optimization,* International Journal of Computational Fluid Dynamics **19**, 595 (2005).

[5] A. H. van Zuijlen, A. de Boer, and H. Bijl, *Higher-order time integration through smooth mesh deformation for 3D fluid-structure interaction simulations,* Journal of Computational Physics **224**, 414 (2007).

[6] B. Froehle and P.-O. Persson, *A high-order discontinuous Galerkin method for fluid–structure interaction with efficient implicit–explicit time stepping,* Journal of Computational Physics **272**, 455 (2014).

[7] R. I. Issa, *Solution of the implicitly discretised fluid flow equations by operator-splitting,* Journal of Computational Physics **62**, 40 (1986).

[8] W. Z. Shen, J. A. Michelsen, and J. N. Sørensen, *Improved Rhie-Chow interpolation for unsteady flow computations,* AIAA Journal **39**, 2406 (2001).

[9] B. Yu, W.-Q. Tao, J.-J. Wei, Y. Kawaguchi, T. Tagawa, and H. Ozoe, *Discussion on momentum interpolation method for collocated grids of incompressible flow,* Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology **42**, 141 (2002).

[10] B. Yu, Y. Kawaguchi, W.-Q. Tao, and H. Ozoe, *Checkerboard pressure predictions due to the underrelaxation factor and time step size for a nonstaggered grid with momentum interpolation method,* Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology **41**, 85 (2002).

[11] C. Rhie and W. Chow, *A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation,* in *3rd Joint Thermophysics, Fluids, Plasma and Heat Transfer Conference* (1982) pp. 1–12.

[12] D. K. Kolmogorov, W. Z. Shen, N. N. Sørensen, and J. N. Sørensen, *Fully consistent SIMPLE-like algorithms on collocated grids,* Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology **67**, 101 (2015).

[13] V. Kazemi-Kamyab, A. H. van Zuijlen, and H. Bijl, *Analysis and application of high order implicit Runge–Kutta schemes to collocated finite volume discretization of the incompressible Navier–Stokes equations,* Computers & Fluids **108**, 107 (2015).

[14] Z. Tuković and H. Jasak, *A moving mesh finite volume interface tracking method for surface tension dominated interfacial fluid flow,* Computers & Fluids **55**, 70 (2012).

[15] T. Richter, *Goal-oriented error estimation for fluid-structure interaction problems,* Computer Methods in Applied Mechanics and Engineering **223-224**, 28 (2012).

[16] S. Turek and J. Hron, *Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow,* in *Fluid-Structure Interaction,* edited by H.-J. Bungartz and M. Schafer (Springer Berlin Heidelberg, 2006) pp. 371–385.

[17] H. Jasak, *Department of Mechanical Engineering Imperial College of Science, Technology and Medicine,* Ph.D. thesis, Imperial College of Science, Technology and Medicine (1996).

[18] A. de Boer, M. S. van der Schoot, and H. Bijl, *Mesh deformation based on radial basis function interpolation,* Computers and Structures **85**, 784 (2007).

[19] A. H. van Zuijlen, A. D. Boer, and H. Bijl, *Higher Order Time Integration for 3D Fluid-Structure Interaction,* in *International Conference on Computational Methods for Coupled Problems in Science and Engineering,* edited by M. Papadrakakis, E. Onate, and B. Schrefler (Barcelona, 2007) pp. 1–4.

[20] P. Gresho and R. Sani, *On pressure boundary conditions for the incompressible Navier-Stokes equations,* Journal for Numerical Methods in Fluids **7**, 1111 (1987).

[21] P. Cardiff, A. Karač, and A. Ivanković, *A large strain finite volume method for orthotropic bodies with general material orientations,* Computer Methods in Applied Mechanics and Engineering **268**, 318 (2014).

[22] Z. Tuković and H. Jasak, *Updated Lagrangian finite volume solver for large deformation dynamic response of elastic body,* Transactions of Famena **31**, 55 (2007).

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

[24] M. Cervera, R. Codina, and M. Galindo, *On the computational efficiency and implementation of block-iterative algorithms for nonlinear coupled problems,* Engineering Computations **13**, 4 (1996).

[25] D. P. Mok, W. A. Wall, and E. Ramm, *Accelerated iterative substructure schemes for instationary fluid-structure interaction,* in *First MIT Conference on Computational Fluid and Solid Mechanics* (2001) pp. 1325–1328.

[26] J. Degroote, R. Haelterman, S. Annerel, P. Bruggeman, and J. Vierendeels, *Performance of partitioned procedures in fluid-structure interaction,* Computers and Structures **88**, 446 (2010).

[27] J. Degroote, *Partitioned Simulation of Fluid-Structure Interaction,* Archives of Computational Methods in Engineering **20**, 185 (2013).

[28] E. Hairer and G. Wanner, *Algebraically Stable and Implementable Runge-Kutta Methods of High Order,* Siam Journal on Numerical Analysis **18**, 1098 (1981).

3

ADAPTIVE RADIAL BASIS FUNCTION MESH DEFORMATION USING DATA REDUCTION

Evolution taught us that adaptation is the key to survival.

Gemma Malley

Mesh deformation is required in the proposed high fidelity model, however it should be both efficient (computationally affordable) and robust. Radial Basis Function (RBF) mesh deformation is one of the most robust mesh deformation methods available. Using the greedy (data reduction) method in combination with an explicit boundary correction, results in an efficient method as shown in literature. However, to ensure the method remains robust, two issues are addressed: 1) how to ensure that the set of control points results in an accurate representation of the geometry in time and 2) how to use/automate the explicit boundary correction, while ensuring a high mesh quality. Adaptively selecting the control points based on the boundary error ensures a robust set of control points. For the explicit boundary correction two new radial basis correction functions are derived and proposed based on a single deforming cell problem. Based on the same single cell problem a automated procedure for the correction radius is derived. This proposed automated procedure is verified while varying the correction function, Reynolds number (and thus first cell height and aspect ratio) and boundary error for an engineering case. Finally, the parallel efficiency is studied for the two adaptive methods, unit displacement and prescribed displacement for both the CPU as well as the memory formulation with a 2D oscillating and translating airfoil with oscillating flap, and the 3D flexible locally deforming tube. By automating most of the RBF mesh deformation, a robust, efficient and almost user-independent mesh deformation method is presented.

Parts of this chapter have been submitted in Gillebaart et al., Adaptive Radial Basis Function Mesh Deformation Using Data Reduction, Journal of Computational Physics (submitted)

3.1. INTRODUCTION

High-fidelity models for aero-elastic analysis are gaining interest due to the increase of computing power and the always present desire to obtain accurate results in several areas: for example for (large scale) wind turbines [1], micro-aerial vehicles (MAV) [2], cardiovascular systems [3] and aerospace [4]. When the FSI model is based on the Arbitrary Eulerian Lagrangian (ALE) formulation of the fluid equations, both the fluid and structure mesh need to be deformed from their initial state. For the structural mesh (and its equations) this is often straightforward due to its Lagrangian formulation, while for the fluid mesh this can be a challenge in terms of accuracy, efficiency and stability.

To assess the performance of the mesh deformation algorithm, four criteria are generally considered:

- resulting mesh quality
- robustness
- efficiency
- user dependency/input required

Mesh quality is of key importance for accuracy and convergence of the system of fluid equations solved. Robustness is considered to be important to ensure that a single algorithm can be used for a wide variation of applications. Thirdly, since FSI calculations are expensive by nature, the mesh deformation algorithm should add the least possible extra computational time to the complete FSI simulation. Finally, an often forgotten performance criterion is user dependency. The more expert knowledge is needed for proper use of the method, the less robust such a method will be.

In literature several options have been discussed for deforming a fluid mesh. The spring analogy [5], Laplacian [6] or Bi-Harmonic smoothing [7] and linear elasticity approach [8] are all grid connectivity based algorithms, which are considered to be relatively expensive since they solve a system of equation equal to the mesh size. Transfinite interpolation [9] is a robust and simple method, but only applicable on structured meshes. Radial basis functions (RBF) [10] results in high quality meshes (thus robust and accurate), while being applicable to both structured, unstructured and hybrid meshes (due its point-to-point nature). Therefore, Radial Basis Function mesh deformation has been gaining interest [11–14].

However, the original formulation can become expensive for large problems, hence acceleration techniques have been applied. Jakobsson and Amiognon [15] were the first to reduce the problem size by reducing the number of used boundary points. A similar approach has been taken by Rendall and Allen, who proposed a more efficient error-based selection method to reduce the problem size [11] (i.e. the greedy method), combined with an explicit correction to ensure the exact geometry is maintained [12]. MMore recently, a multi-level subspace RBF interpolation (based on a double-edge greedy method) has been designed to enhance the selection efficiency [16] and combining RBF with the Delaunay graph method shows promising results [17]. For the greedy based reduction methods a deformation of the boundary is required. As proposed by Rendall and

Allen this can either be: a unit displacement (all boundary points have a unit displacement in all translational directions), predefined deformation (e.g. structural modes) or the actual deformation. Rendall and Allen suggested using the unit displacement, since this makes the selection independent of the actual deformation (at the cost of a reduction of accuracy of the boundary geometry and/or efficiency) [13]. Besides the reduction of boundary points, the parallelization of the method has been studied [14]. Here they did not consider any reduction algorithm and used the full set boundary points. Rendall and Allen briefly discussed parallelization, but did not study or discuss it in depth [12].

Even though a unit displacement as selection deformation has its benefits (only needed once and simple to implement), it potentially causes a reduction in accuracy or efficiency. Therefore, a more realistic prescribed deformation or the actual deformation could be used to select a set of boundary points. Here the prescribed deformation needs a priori knowledge (i.e. obtaining/creating a realistic deformation based on e.g. structural modes), which is undesirable, since it increases the user-dependency. Ideally, the actual deformation is used for selection, since it is already available and represents the structure deformation best. However, how to approach the changing deformations over time during a (FSI) simulation? And how does this influence the accuracy and efficiency of the RBF mesh deformation method? In addition, to have an efficient (parallel scalable) method, the number of control points should be limited or ideally mesh independent. However, this has not been shown yet, and is unknown until now. In addition, with a subset of boundary points as control points, a boundary error will exist. A simple explicit master-slave like correction [18] can be applied, as suggested by Rendall and Allen [12]. However, how does the correction influence the mesh quality and how to use radial basis functions efficiently for this: how to choose the function and its radius?

The goal of this chapter is to present an adaptive radial basis function mesh deformation and show its (parallel) performance by analyzing the following criteria: (parallel) efficiency (cost of computation), mesh quality and user dependency. Secondly, a detailed analysis of the influence of the explicit boundary correction on the mesh quality is performed, after which an automated procedure for the explicit correction step is proposed and verified. Additionally, a question/assumption for the (parallel) efficiency and robustness of the RBF method will be answered: is the number of control points mesh independent for a fixed selection criterion? With this new knowledge, the proposed adaptive method will be compared to the unit displacement and prescribed method to assess its robustness, efficiency and user dependency.

First, the radial basis function mesh deformation method is described for both a CPU intensive formulation and a memory intensive formulation. Here also the theoretical cost of the two formulations is discussed. After this, the greedy algorithm is shortly summarized. Based on the original greedy algorithm the adaptive method is described. Consequently, the boundary correction step and its radius and functions are discussed based on an analytical analysis of a single high aspect ratio cell. The result section consist of three parts: 1) mesh independency of the selection algorithm and the verification of the proposed automated boundary correction method, 2) influence of selection deformation on accuracy and efficiency, and 3) computational work and parallel efficiency of the discussed methods and their CPU and memory formulation.

3.2. METHODS

Radial Basis Function mesh deformation is a point-to-point method. Such a method deforms the points/vertices of the mesh based on the motion of a set of (boundary) points. Generally the motion of the boundary points is known and will be used to deform the internal mesh. As illustrated in Figure 3.1 the domain exists out of internal ($\vec{\mathbf{x}}_i \in \Omega_i$), moving boundary ($\vec{\mathbf{x}}_m \in \Gamma_m$), static boundary ($\vec{\mathbf{x}}_s \in \Gamma_s$), boundary ($\vec{\mathbf{x}}_b = (\vec{\mathbf{x}}_s \cup \vec{\mathbf{x}}_m) \in \Gamma_b = (\Gamma_s \cup \Gamma_m)$) and control ($\vec{\mathbf{x}}_c \subseteq \vec{\mathbf{x}}_s \cup \vec{\mathbf{x}}_m$) points.

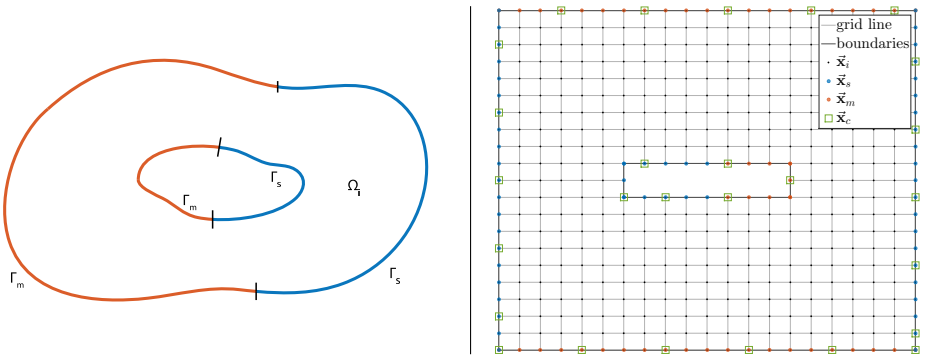

Figure 3.1: Continuous and discrete schematic representation of internal, boundary, moving boundary, static boundary and control points.

3.2.1. GOVERNING EQUATIONS

The radial basis function mesh deformation method is governed by two parts: solving a linear system of equations and evaluating a second system of equations. As discussed by De Boer et al. [10], the basis of the RBF mesh deformation is to interpolate the boundary deformation to the internal mesh by means of radial basis functions:

$$\vec{\mathbf{s}}(\vec{\mathbf{x}}) = \sum_{j=1}^{n_b} \vec{\alpha}_j \phi\left(\left\|\vec{\mathbf{x}} - \vec{x}_{b_j}\right\|\right), \tag{3.1}$$

where $\vec{x}_{b_j} = \left[x_{b_j}, y_{b_j}, z_{b_j}\right]$ are the boundary points at which the displacement is known, ϕ is the radial basis function as function of the Euclidean distance $\|\vec{\mathbf{x}}\|$, n_b are the number of boundary points and $\vec{\alpha}_j = \left[\alpha_{x_j}, \alpha_{y_j}, \alpha_{z_j}\right]$ are the function coefficients. For general RBF interpolations, an additional polynomial is added to the equation to ensure exact interpolation of uniform functions. However, when applying RBF interpolation to mesh deformation uniform displacement is not considered, since no interpolation would be needed. Therefore, the polynomial is dismissed, as done by Rendall and Allen [11]. In addition to the interpolation function the following condition must be satisfied:

$$\vec{s}\left(\vec{x}_{b_j}\right) = \Delta\vec{x}_{b_j}, \tag{3.2}$$

where $\Delta\vec{x}_{b_j}$ are the displacements of the boundary point j in three directions: $\Delta\vec{x}_{b_j} = \left[\Delta x_{b_j}, \Delta y_{b_j}, \Delta z_{b_j}\right]$. Combining this condition with Equation (B.1) the coefficients $\vec{\boldsymbol{\alpha}}$ are

determined from:

$$\Delta\vec{\mathbf{x}}_b = \Phi_{b,b}\vec{\boldsymbol{\alpha}}, \tag{3.3}$$

which can be solved for the coefficients as:

$$\vec{\boldsymbol{\alpha}} = \Phi_{b,b}^{-1}\Delta\vec{\mathbf{x}}_b. \tag{3.4}$$

Here $\Phi_{b,b}$ is the matrix containing n_b by n_b RBF evaluations: $\Phi_{b_i,b_j} = \phi\left(\left\|\mathbf{x}_{b_i} - \mathbf{x}_{b_j}\right\|\right)$. With the obtained coefficients ($\vec{\boldsymbol{\alpha}} = \left[\boldsymbol{\alpha}_x, \boldsymbol{\alpha}_y, \boldsymbol{\alpha}_z\right]$) the internal point deformation is interpolated:

$$\Delta\vec{x}_{in_i} = \vec{s}\left(\vec{x}_{in_i}\right) = \sum_{j=1}^{n_b} \vec{\alpha}_j \phi\left(\left\|\vec{x}_{in_i} - \vec{x}_{b_j}\right\|\right), \tag{3.5}$$

where $\Delta\vec{x}_{in_i} = \left[\Delta x_{in_i}, \Delta y_{in_i}, \Delta z_{in_i}\right]$. The displacement in each spatial direction are interpolated separately with the corresponding set of coefficients (e.g. $\boldsymbol{\alpha}_x$ for x-direction). In matrix notation the evaluation step becomes:

$$\Delta\vec{\mathbf{x}}_{in} = \Phi_{in,b}\vec{\boldsymbol{\alpha}}, \tag{3.6}$$

where $\Phi_{in,b}$ is the matrix containing n_i by n_b RBF evaluations. Using Equation 3.4 to obtain the coefficients and calculating the interpolation to the internal points with Equation 3.5 is known as the CPU formulation. For radial basis function mesh deformation two formulations are presented: a CPU intensive formulation, which evaluates the RBF's each time, and a memory intensive formulation, which saves the evaluation matrix to memory. Rendall and Allen have mentioned that the above set of equations can be written in a compact (memory intensive) form as follows [11]:

$$\Delta\vec{\mathbf{x}}_{in} = \Phi_{in,b}\Phi_{b,b}^{-1}\Delta\vec{\mathbf{x}}_b = H\Delta\vec{\mathbf{x}_b}, \tag{3.7}$$

where $H = \Phi_{in,b}\Phi_{b,b}^{-1}$. The size of H is n_{in} by n_b. An additional step is proposed, eliminating the static boundary points from this final formulation, the size of the matrix H can be reduced:

$$\Delta\vec{\mathbf{x}}_{in} = H\begin{bmatrix}\Delta\vec{\mathbf{x}}_m \\ \Delta\vec{\mathbf{x}}_s = \mathbf{0}\end{bmatrix} = H_{red}\Delta\vec{\mathbf{x}}_m. \tag{3.8}$$

Here $\Delta\vec{\mathbf{x}}_m$ are the displacements of the moving boundary points, $\Delta\vec{\mathbf{x}}_s$ are the displacements of the static boundary points, which are $\vec{\mathbf{0}}$ by definition. The reduced matrix, H_{red}, has the size n_{in} by n_m. Depending on the case, there might be a large difference between n_b and n_m, especially for internal flow problems (e.g. channel flow like cases [19, 20], artery flows [3, 21]). Generally the formulation in Equation 3.7 or 3.8 is not considered because of its memory cost. However, with the development of reduction algorithms, the parallelization possibilities and increase of memory availability, the memory cost are often reasonable, as will be shown in the next section.

3.2.2. MEMORY VERSUS CPU

Considering a memory intensive formulation of the RBF mesh deformation becomes clear when the estimated amount of work, in terms of Floating-point Operation per Sec-

ond (FLOPS), for the CPU intensive (Equation 3.4 and 3.6) and memory intensive (Equation 3.8) formulation are compared. Here the focus is on the evaluation of the internal displacement, since this is repeated many times during a simulation. The work (in FLOPS) for the CPU formulation in 3D space is:

$$\begin{aligned} W_{CPU} &= \overbrace{3n_b(2n_b-1)}^{W^{\boldsymbol{\alpha}}_{CPU}} + \overbrace{3n_{in}(2n_b-1)+W_{RBF}n_{in}n_b}^{W^{\Delta\vec{\mathbf{x}}_{in}}_{CPU}} \\ &\approx 6n_bn_b + (6+W_{RBF})\,n_{in}n_b, \end{aligned} \tag{3.9}$$

where $W^{\boldsymbol{\alpha}}_{CPU}$ is the cost of the matrix vector product of Equation 3.4, $W^{\Delta\vec{\mathbf{x}}_{in}}_{CPU}$ is the cost of the evaluation of the internal displacement described in Equation 3.6 and W_{RBF} is the number of FLOPS for the evaluation of the RBF (including the calculation of the radius), which depends on the chosen function. The cost of solving for the inverse of $\Phi_{b,b}$ is scaling with the 3^{rd} power, but is only depending on the significantly smaller n_b (and with selection on n_c). Additionally, it is executed a single time for a fixed set of control points:

$$W_{\Phi^{-1}_{b,b}} = \frac{2}{3}n_bn_bn_b \tag{3.10}$$

For the memory formulation in 3D space, the work for Equation 3.8 is:

$$W_{MEM} = 3n_{in}(2n_m-1) \approx 6n_{in}n_m. \tag{3.11}$$

For the memory intensive formulation there is additional work of constructing the matrix:

$$W_{Hred} = W_{\Phi^{-1}_{b,b}} + \overbrace{n_{in}n_m(2n_b-1)}^{W^{H\Phi^{-1}_{b,m}}_{MEM}} + W_{RBF}n_{in}n_b \approx 2n_{in}n_mn_b + W_{RBF}n_{in}n_b. \tag{3.12}$$

However, this only has to be done a single time for a fixed set of control points. Assuming $n_m << n_{in}$, results in a ratio between the work (FLOPS) of the two formulation of:

$$\frac{W_{CPU}}{W_{MEM}} = \frac{(6+W_{RBF})}{6} = 1 + \frac{W_{RBF}}{6} \tag{3.13}$$

From this it can be seen that its beneficial to use the memory formulation as soon as many evaluation steps need to be performed with the same set of boundary points. The increase in efficiency of the memory intensive formulation depends on the cost of a single basis function evaluation (W_{RBF}). For the most often used set of basis functions this will be shown. However, H_{red} needs to be stored, which has a costs in terms of memory. The matrix of doubles to be stored (H_{red}) is of the size n_{in} x n_m, resulting in a size in MB:

$$S_{H_{red}} = \frac{8n_{in}n_m}{10^6}. \tag{3.14}$$

Often the memory intensive method is discarded because of the size of H_{red}, because $n_{in}n_m$ scales rapidly with number of nodes in a mesh. However, the matrix can easily be decomposed for the internal points n_{in}, by means of domain decomposition. Assuming

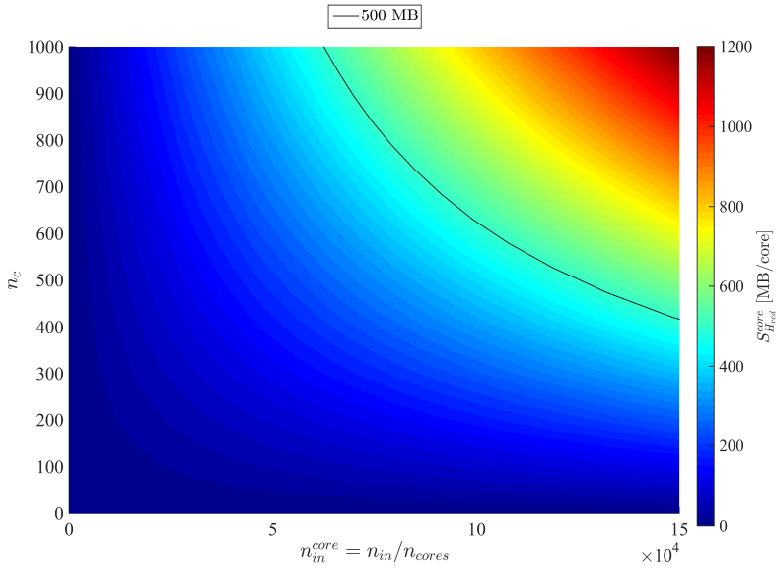

Figure 3.2: Size of H_{red}^{core} in Megabytes (MB) for varying n_{in}^{core} and n_c. The black line represents a limiting line of 500 MB.

that the number of nodes per core remains similar, the size of the matrix only scales with n_m, since all control points should be present on all cores. Therefore control point selection is crucial. This will reduce n_b and thus n_m to a subset of control points $\vec{\mathbf{x}}_c$ of size n_c. Is is assumed that n_c does not scale with the number of nodes for a specific problem but is purely problem dependent: n_c is related to representing the structural deformation with a set of points and not to the number of nodes on the fluid mesh boundary. This results in the following size of H_{red} per core in MB:

$$S_{H_{red}}^{core} = \frac{8 n_{in}^{core} n_{mc}}{10^6}. \tag{3.15}$$

where n_{in}^{core} are the number of internal points per core and n_{mc} are the number of moving control points. In Figure 3.2 the size of H_{red}^{core} is shown for a range of internal and control points.

3.2.3. CONTROL POINT SELECTION

For both formulations the number of control points is the most crucial parameter, since it remains constant when parallelizing the method and all control points must be available on all CPU's. Therefore a subset of the boundary points can be used (i.e. control points), which are related to structure/boundary deformation instead of the number of boundary points in the fluid mesh. As proposed by Rendall and Allen [11], the greedy algorithm is used to create subset of boundary points ($\vec{\mathbf{x}}_c$) to control the deformation of the internal points. This subset is obtained by an iterative procedure, adding the boundary point with the largest error to this subset, until a certain criteria is satisfied. The choice of convergence criteria is a crucial part of the method. Ideally, the selected set of

control points is able to deform the internal mesh accordingly based on deformed shape of the boundary. Secondly, the influence of the amplitude of the deformation on the set of selected points is minimal (for an appropriately chosen radius), since radial basis function behave linearly with the displacement. Therefore, the convergence criteria are normalized by the boundary displacement:

$$\frac{\|\vec{\boldsymbol{\varepsilon}}_b\|_2}{\|\Delta\vec{\mathbf{x}}_b\|_2} < \varepsilon_{tol} \qquad \&\& \qquad \frac{\|\vec{\boldsymbol{\varepsilon}}_b\|_\infty}{\|\Delta\vec{\mathbf{x}}_b\|_\infty} < \varepsilon_{tol} \tag{3.16}$$

where $\vec{\boldsymbol{\varepsilon}}_b$ is the boundary error, $\Delta\vec{\mathbf{x}}_b$ the boundary displacement and ε_{tol} is the user specified tolerance for the greedy selection. By normalizing the boundary error norm with the norm of the displacement the set of control points becomes (almost) independent of the amplitude and mainly dependent on the shape, which is desirable. In this way a set of points is chosen, which can present the shape until a certain accuracy (ε_{tol}). The greedy algorithm is summarized in more detail in A.1.

3.2.4. SELECTION SHAPE

For the selection algorithm a boundary deformation is required. In the studies performed until now the deformation shape is predefined with either a unit displacement or a prescribed displacement (e.g. modal shapes of structure). Where the unit displacement is straightforward to apply, having a modal shape available requires an additional effort (by performing a modal analysis on the structure). The drawback of the unit displacement is a less optimal selection, because of the simplification of the actual deformation. By selecting a set of points before the simulation based on a predefined deformation shape, either the geometry might not be captured accurately or a too large set of points is selected making the algorithm more expensive. In this thesis an adaptive way of selecting the control points is proposed: base the selection on the actual deformations/displacements during the simulation and update the set of control points if required as shown in Algorithm 1.

Based on the re-selection criteria ($\varepsilon_{tol}^{reselection}$) the greedy algorithm is re-executed during the FSI calculation to obtain a new set of control points. This ensures that the geometrical deformation is always captured to the same accuracy during the simulation. In addition, no pre-processing is needed: in the first time step a selection is made based on the first deformation. This does work well, since the selection is based on the deformation shape and is (almost) independent of the amplitude of deformation. The re-selection criteria is related to the greedy selection tolerance (ε_{tol}):

$$\varepsilon_{tol} = \rho\varepsilon_{tol}^{reselect} \tag{3.17}$$

where ρ varies between 0 and 1. By setting $\rho = 1$ the greedy selection is performed until the same tolerance as the re-selection. However, for more extreme deformation, it is more efficient to have the greedy selection to select a larger set of points than the desired minimal tolerance ($\varepsilon_{tol}^{reselect}$) to ensure the re-selection is performed less frequent. A value of ρ near 0.1 is advised based on experience of the authors.

Algorithm 1 Adaptive Greedy RBF ($\vec{\mathbf{x}}_b, \vec{\mathbf{x}}_c, \Delta\vec{\mathbf{x}}_b, \Delta\vec{\mathbf{x}}_c$)

for time = 0 **to** time = t_{end} **do**
  $\vdots$
  $\downarrow$
  $\Delta\vec{\mathbf{x}}_b$ # Updated boundary displacement
  $\vec{\boldsymbol{\varepsilon}}_b$ = Boundary Error($\vec{\mathbf{x}}_b, \vec{\mathbf{x}}_c, \Delta\vec{\mathbf{x}}_b, \Delta\vec{\mathbf{x}}_c$) # Get boundary error
  if $\frac{\|\vec{\boldsymbol{\varepsilon}}_b\|_2}{\|\Delta\vec{\mathbf{x}}_b\|_2} > \varepsilon_{tol}^{reselect}$ **then**
    $i_0 \leftarrow \Delta x_{b_{i_0}} = \max\left(\|\Delta\vec{\mathbf{x}}_b\|_2\right)$
    $\vec{\mathbf{x}}_c = \vec{\mathbf{x}}_b[i_0]$
    $\Delta\vec{\mathbf{x}}_c = \Delta\vec{\mathbf{x}}_b[i_0]$
    $\vec{\mathbf{x}}_c$ = Greedy($\vec{\mathbf{x}}_b, \vec{\mathbf{x}}_c, \Delta\vec{\mathbf{x}}_b, \Delta\vec{\mathbf{x}}_c$) # do Greedy
    $H_{red} = \Phi_{in,c}\left(\Phi_{c,c}^{-1}[:, 1:n_{mc}]\right)$
  end if
  $\Delta\vec{\mathbf{x}}_{in} = H_{red}\Delta\vec{\mathbf{x}}_{mc}$ # Deform internal fluid mesh
  $\Delta\vec{\mathbf{x}} = [\Delta\vec{\mathbf{x}}_{in}, \Delta\vec{\mathbf{x}}_b]$ # Set all fluid mesh points to new location
  $\downarrow$
  $\vdots$
end for

3.2.5. BOUNDARY CORRECTION

To complete the adaptive RBF mesh deformation, a correction step can be executed as proposed by Rendall and Allen [12]. For the boundary points not included in the control points subset an explicit boundary correction is applied to obtain the exact boundary representation. However, the boundary error should be sufficiently reduced with the RBF mesh deformation before the correction is applied to ensure a good mesh quality, as will be shown later. For the correction a compact supported radial basis function seems to be a good choice, since only a small part of the mesh needs to be moved and chances of overlapping of grid lines/cells is smaller. As proposed by Rendall and Allen the correctional displacement for the internal mesh points is determined by [12]:

$$\Delta\vec{\mathbf{x}}_{corr}\left(x_{in_i}\right) = \vec{\boldsymbol{\varepsilon}}_b \phi\left(\frac{\left\|\vec{x}_{in_i} - \vec{x}_{b_j}\right\|}{R_{corr}}\right) \tag{3.18}$$

where the correction displacement of a single point is only dependent on the boundary point closed to that point ($\vec{x}_{b_j}$). This allows a single nearest neighbor (boundary point) search, after which a single function evaluation per internal point is needed each time step. However, for the compact supported radial basis function a radius must be chosen: R_{corr}. To limit the influence of the explicit correction step, the radius should be chosen as small as possible, to ensure only a small part of the mesh is affected by the additional mesh deformation. However, at the same time the radius should be chosen big enough, such that the mesh quality is almost not influenced by the correction step. In Figure

3.3 an illustration is given of a too small correction radius, good correction radius and too large correction radius. As for the mesh deformation, the radius should be larger

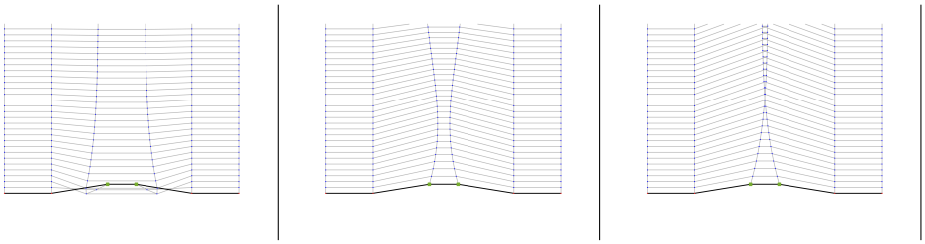

Figure 3.3: Illustration of too small correction radius (left, leading to overlapping at the boundary) a properly chosen correction radius (middle) and a too large correction radius (right, leading to overlapping in the internal mesh).

than the magnitude of the displacement. Here the magnitude is equal to the error at the boundary, which again is dependent on the tolerance of the greedy algorithm (ε_{tol}) and adaptive algorithm ($\varepsilon_{tol}^{reselect}$). A first approach is to linearly relate R_{corr} to the maximum error (ε_b^{max}) on the boundary:

$$R_{corr} = \gamma \varepsilon_b^{max}, \tag{3.19}$$

where γ is larger than 1 to ensure a smooth distribution of the displacement. This function is linearly dependent on the tolerance, as shown in Figure 3.4, which presents the theoretical limit lines for the correction function. The blue vertical dashed line, presents the line where the error is equal to the first cell height. Results to the left of this line should result in a valid mesh. However, for high aspect ratio cells (often present in fluid meshes), an additional criteria is derived, represented by the yellow dashed line in Figure 3.4. This criteria is derived based on a single high aspect ratio cell, with a small displacement at the top right corner, as shown in Figure 3.5. In Appendix A.2, the relation between the mesh quality of the first cell near the boundary and the aspect ratio and the displacement magnitude is derived:

$$\lambda = \tan^{-1}\left(\frac{\Delta C_x}{C_y^d}\right) = \tan^{-1}\left(\frac{1}{2}\frac{\xi A}{3+3\xi+\xi^2}\right), \tag{3.20}$$

where λ is the non-orthogonal parameter (angle between face center and cell center), A the aspect ratio and $\xi = \frac{\Delta y}{y_s}$ the ratio between displacement and cell height. This sensitivity could become the dominating factor in the correction radius determination, depending on the aspect ratio and the correction function used. Therefore, the sensitivity of correction radius and function on the first cell quality are derived. With the boundary correction function approach, the displacement at the top right corner point is depending on the correction function value and its corresponding correction radius:

$$\xi = \frac{\Delta y}{y_s} = \frac{\varepsilon_b \phi(r) - \varepsilon_b}{y_s} = \frac{\varepsilon_b}{y_s}\left(1-\phi(r)\right) = \frac{\varepsilon_b}{y_s}\left(1-\phi\left(\frac{y_s}{R_{corr}}\right)\right) \tag{3.21}$$

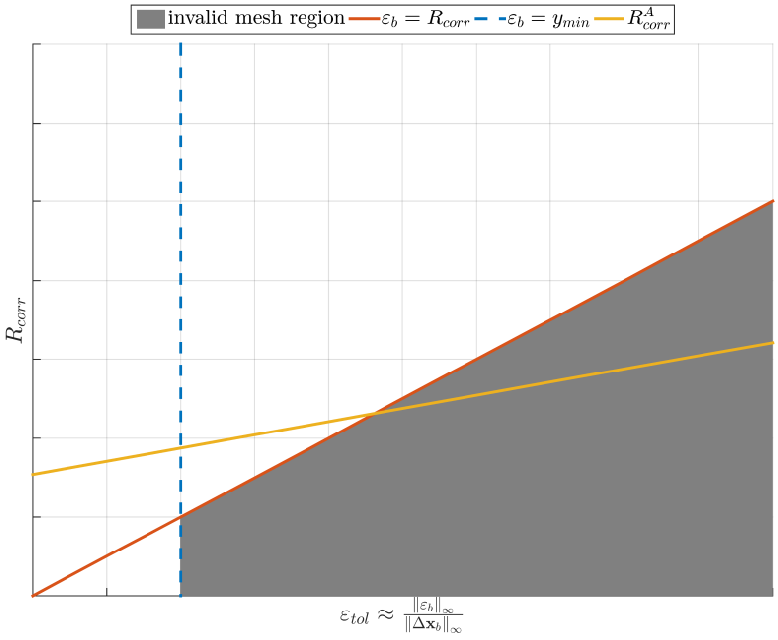

Figure 3.4: Overview of theoretical limiting lines for correction radius magnitude (R_{corr}) versus the selection tolerance (ε_{tol}). One invalid mesh region can be identified, when the correction radius is smaller than the boundary error and the boundary error is bigger than the first cell height. A second limitline is found by means of the analysis of a deforming single cell. This line is dependent on the boundary correction function used, aspect ratio and boundary error and first cell height.

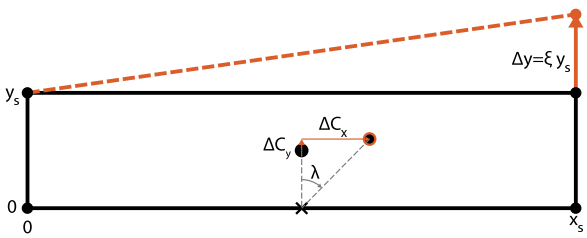

Figure 3.5: A single rectangular cell and its displacement of the right top point. Due to this displacement the cell center will move, causing in increase in non-orthogonality.

Substituting this into Equation (3.20) results in:

$$\frac{\Delta C_x}{C_y^d} = \frac{A}{2} \frac{\frac{\varepsilon_b}{y_s}\left(1-\phi(r)\right)}{3+3\frac{\varepsilon_b}{y_s}\left(1-\phi(r)\right)+\left[\frac{\varepsilon_b}{y_s}\left(1-\phi(r)\right)\right]^2}. \tag{3.22}$$

With the desire that the cell remains as close as possible to its original shape, $1-\phi(r) \approx 0$ for the top right point. Considering that

$$\lim_{r\to 0}\left(1-\phi(r)\right) = 0$$

, assuming $r << 1$ results in the desired limiting behavior of the correction function. Assuming $r << 1$, Equation (3.22) is simplified in:

$$\begin{aligned} \frac{\Delta C_x}{C_y^d} &\approx \frac{A}{2}\frac{\frac{\varepsilon_b}{y_s}\left(1-\phi(r)\right)}{3} \qquad \text{(for: } r << 1\text{)} \\ &= \frac{A}{6}\frac{\varepsilon_b}{y_s}\left(1-\phi(r)\right) = \frac{A}{6}\xi(r) \end{aligned} \tag{3.23}$$

The influence of the function on the non-orthogonality is determined by the behavior of $1-\phi(r)$ when $r << 1$. Or differently stated, when $R_{corr} >> y_s$. For the WendlandC0 function:

$$\phi(r) = (1-r)^2, \tag{3.24}$$

this becomes:

$$\begin{aligned} \frac{A}{6}\frac{\varepsilon_b}{y_s}\left(1-\phi(r)\right) &= \frac{A}{6}\frac{\varepsilon_b}{y_s}\left(1-[1-r]^2\right) \\ &= \frac{A}{6}\frac{\varepsilon_b}{y_s}\left(1-\left[1-2r+r^2\right]\right) = \frac{A}{6}\frac{\varepsilon_b}{y_s}\left(2r-r^2\right) \\ &\approx \frac{A}{6}\frac{\varepsilon_b}{y_s}2r \qquad \text{(for: } r << 1\text{)} \\ &= \frac{A}{3}\frac{\varepsilon_b}{y_s}r = \frac{A}{3}\frac{\varepsilon_b}{y_s}\frac{y_s}{R_{corr}} \end{aligned} \tag{3.25}$$

and for the WendlandC2 function:

$$\phi(r) = (1-r)^4(4r+1), \tag{3.26}$$

this is

$$\begin{aligned} \frac{A}{6}\frac{\varepsilon_b}{y_s}\left(1-\phi(r)\right) &= \frac{A}{6}\frac{\varepsilon_b}{y_s}\left(1-\left[(1-r)^4(4r+1)\right]\right) \\ &= \frac{A}{6}\frac{\varepsilon_b}{y_s}\left(1-\left[1-10r^2+20r^3-15r^4+4r^5\right]\right) \\ &= \frac{A}{6}\frac{\varepsilon_b}{y_s}\left(10r^2-20r^3+15r^4-4r^5\right) \\ &\approx \frac{A}{6}\frac{\varepsilon_b}{y_s}10r^2 \qquad \text{(for: } r << 1\text{)} \\ &= \frac{5A}{3}\frac{\varepsilon_b}{y_s}r^2 = \frac{5A}{3}\frac{\varepsilon_b}{y_s}\left(\frac{y_s}{R_{corr}}\right)^2 \end{aligned} \tag{3.27}$$

Comparing WendlandC0 and WendlandC2, it becomes apparent that the WendlandC2 would need a smaller R_{corr} (or larger r) for a similar non-orthogonality of the first cell. From this derivation it becomes clear that the important characteristic is the limit of 1 minus the weighting function used for the boundary correction step. With this knowledge two new function are derived. Using the following conditions:

$$\begin{aligned} 1-\phi(r) &= O\left(r^3\right) \qquad \text{(for: } r << 1\text{)} \\ \phi(0) &= 1 \qquad \phi'(0)=0 \qquad \phi''(0)=0 \\ \phi'(1) &= 0 \end{aligned}$$

results in:

$$\phi(r) = 1-4r^3+3r^4 \tag{3.28}$$

Having an additional condition:

$$\phi''(1) = 0$$

results in the 2^{nd} new function:

$$\phi(r) = 1-10r^3+15r^4-6r^5 \tag{3.29}$$

Since these functions are not found in literature by the authors, they are for the moment named GillebaartR3 and GillebaartR3a, respectively. Looking at the limiting case of GillebaartR3:

$$\begin{aligned} \frac{A}{6}\frac{\varepsilon_b}{y_s}\left(1-\phi(r)\right) &= \frac{A}{6}\frac{\varepsilon_b}{y_s}\left(1-\left[1-4r^3+3r^4\right]\right) = \frac{A}{6}\frac{\varepsilon_b}{y_s}\left(4r^3-3r^4\right) \\ &\approx \frac{A}{6}\frac{\varepsilon_b}{y_s}4r^3 \qquad \text{(for: } r << 1\text{)} \\ &= \frac{2A}{3}\frac{\varepsilon_b}{y_s}r^3 = \frac{2A}{3}\frac{\varepsilon_b}{y_s}\left(\frac{y_s}{R_{corr}}\right)^3 \end{aligned} \tag{3.30}$$

and the GillebaartR3a:

$$\begin{aligned} \frac{A}{6}\frac{\varepsilon_b}{y_s}\left(1-\phi(r)\right) &= \frac{A}{6}\frac{\varepsilon_b}{y_s}\lim_{r\to 0}\left(1-\left[1-10r^3+15r^4-6r^5\right]\right) \\ &= \frac{A}{6}\frac{\varepsilon_b}{y_s}\left(10r^3-15r^4+6r^5\right) \\ &\approx \frac{A}{6}\frac{\varepsilon_b}{y_s}10r^3 \qquad \text{(for: } r << 1\text{)} \\ &= \frac{5A}{3}\frac{\varepsilon_b}{y_s}r^3 = \frac{5A}{3}\frac{\varepsilon_b}{y_s}\left(\frac{y_s}{R_{corr}}\right)^3 \end{aligned} \tag{3.31}$$

Based on these limiting values a relation between the correction function radius (R_{corr}) and the aspect ratio, first cell height and boundary error is derived, representing the yellow dashed line in Figure 3.4. This results in a bound of the chosen correction radius,

based on a value for $\frac{\Delta C_x}{C_y^d} = 1$, which results in a $\lambda = 45^o$. For the WendlandC0 this becomes:

$$R_{corr}^A = \frac{A\varepsilon_b}{3} \tag{3.32}$$

for the WendlandC2 function:

$$R_{corr}^A = \sqrt{\frac{5}{3} A\varepsilon_b y_s} \tag{3.33}$$

for the GillebaartR3 function:

$$R_{corr}^A = \left(\frac{2}{3} A\varepsilon_b y_s^2\right)^{\frac{1}{3}} \tag{3.34}$$

and for the GillebaartR3a function:

$$R_{corr}^A = \left(\frac{5}{3} A\varepsilon_b y_s^2\right)^{\frac{1}{3}} \tag{3.35}$$

Finally, the proposed correction radius for the correction function is:

$$R_{corr} = \gamma \max\left(\varepsilon_b^{max}, R_{corr}^A\right), \tag{3.36}$$

where γ is the only parameter left to be determined.

3.2.6. BOUNDARY NON-ORTHOGONALITY

Based on the sensitivity analysis discussed in Appendix A.2, high Reynolds number calculations (with high aspect ratio meshes) could suffer from fast degradation of orthogonality near the boundary. This could happen without the influence of the boundary correction, but purely due to the RBF mesh deformation itself, since there is no direct method of including non-orthogonality within the mesh interpolation method. Generally RBF mesh deformation is good in preserving orthogonality. However, in the case of linear elastic deformations, where the displacements in the three directions are independent of each other, this it not the case. A initial solution is proposed by incorporating orthogonality explicitly in the formulation by adding additional points near the boundary. This can result in a significant reduction of the non-orthogonality at the boundary, as shown in Figure 3.6. Combined with the greedy algorithm, the initial preliminary results are promising. In Appendix B the description of the initial method in combination with the greedy algorithm are discussed. Preliminary results are shown for an academic test case, showing promising initial results.

3.3. RESULTS

Both in our work as in literature two assumptions are made: 1) that the number of control points selected by a greedy algorithm is grid independent, since it is related to the structural deformation, and 2) that a simple boundary correction to eliminate the final boundary error has no significant influence on the mesh quality when a proper function and radius is chosen. However, both of these assumptions have not been thoroughly validated. Therefore, a 2D airfoil test case is used to assess the validity of these two assumptions for a range of parameters. For the adaptive RBF mesh deformation two cases

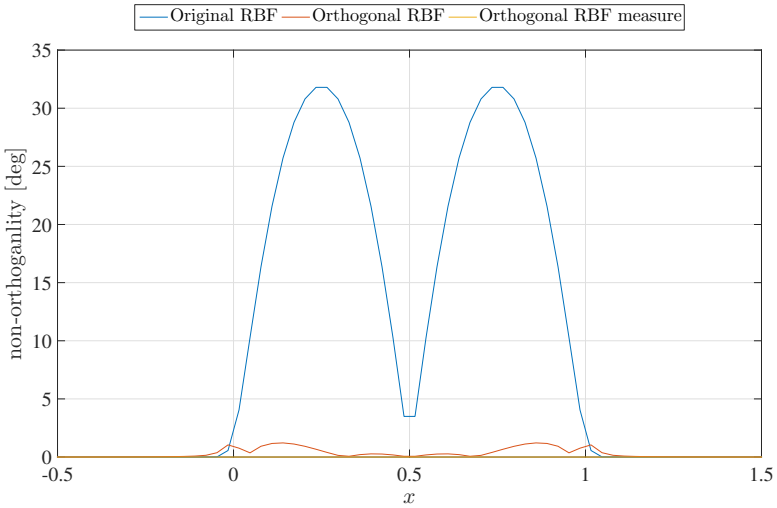

Figure 3.6: Non-Orthogonality on boundary surface for original RBF and Orthogonal RBF.

are considered to show how well it performs: a 2D airfoil with a double-hinged trailing edge flap (TEF) and a 3D tube case with flexible walls [21, 22]. These illustrate two types of problems found in mesh deformation applications: external flows (e.g. airfoil, aircraft wings, wind turbine blades) and internal flows (e.g. cardiovascular systems, pipe flows, gas turbines).

3.3.1. MESH DEPENDENCY AND BOUNDARY CORRECTION

To assess the mesh independency and the theoretical framework (derived in section 3.2.5) of the boundary correction and its influence on the mesh quality, a rotating and translating 2D airfoil with a double-hinge trailing edge flap is used. The motion of the airfoil and flap are described by:

$$x_t = \frac{A_x}{c}\sin(wt) \tag{3.37}$$

$$y_t = \frac{A_y}{c}\sin(wt) \tag{3.38}$$

$$\theta = A_\theta \sin(wt) \tag{3.39}$$

$$\beta = A_\beta \sin(wt), \tag{3.40}$$

where x_t is the translation in x-direction, y_t is the translation in y-direction, $\frac{A_x}{c}$ is the chord normalized amplitude in x-direction, $\frac{A_y}{c}$ is the chord normalized amplitude in y-direction, θ is the pitch angle around quarter chord point, A_θ is the pitch amplitude, β is the flap deflection, A_β is the flap rotation amplitude. The rotation point of the flap is either at the bottom of the airfoil surface for downward ($\beta > 0$) flap deflections and at the top of the airfoil surface for upwards ($\beta < 0$) flap deflections. For the cases considered the flap length is equal to $0.2c$, $A_\beta = 15^o$, $A_\theta = 30^o$, $\frac{A_x}{c} = 1.0$ and $\frac{A_y}{c} = 1.0$. In Figure 3.7

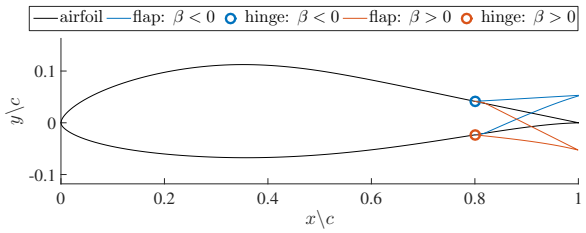

Figure 3.7: Illustration of the 2D airfoil with flap case. Two flap deflections (positive and negative) and the corresponding hinge locations are given.

the airfoil surface and two flap deflections (including the corresponding hinge location) are illustrated.

MESH INDEPENDENCY

In a single step the mesh is deformed based to the maximum displacement (at $wt = \pi/2$) of the airfoil with flap and the adaptive greedy algorithm (selection based on the actual deformation) is used to obtain a set of control points. To determine the mesh independency of the selected number of control points, the mesh is varied in size in both directions: number of points along the airfoil and number of points towards the far field. The resulting mesh characteristics are shown in Table 3.1. To check whether the greedy

Table 3.1: Meshes for airfoil flap case. N indicates the total number of points and n_b indicates the number of points on the boundary (airfoil and flap).

	mesh 1	mesh 2	mesh 3	mesh 4
N	6885	27587	108483	430328
n_b	64	128	256	512

selection does results in a mesh independent number of control points, the selection is performed with different tolerances (ε_{tol}): 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}. The resulting number of selected control points is shown in Figure 3.8. From the Figure it becomes clear that indeed the number of selected control points is mesh independent, and mainly dependent on the selection tolerance. For decreasing tolerance, the independency becomes weaker, since in the limit of selecting all control points (i.e. all boundary points) the relation will be linear, as indicated by the dashed line. However, using a selection tolerance stricter than 10^{-6} would create a very inefficient RBF mesh deformation. This becomes even more apparent when considering the boundary correction.

BOUNDARY CORRECTION

Based on the analysis presented in section 3.2.5 the proposed correction radius for the complete range of tolerances is given by Equation (3.36), which is depending on the used correction function, aspect ratio (A), first cell height (y_s) and maximum boundary error (ε_b^{max}). To validate the function and boundary error dependency, a typical mesh for a

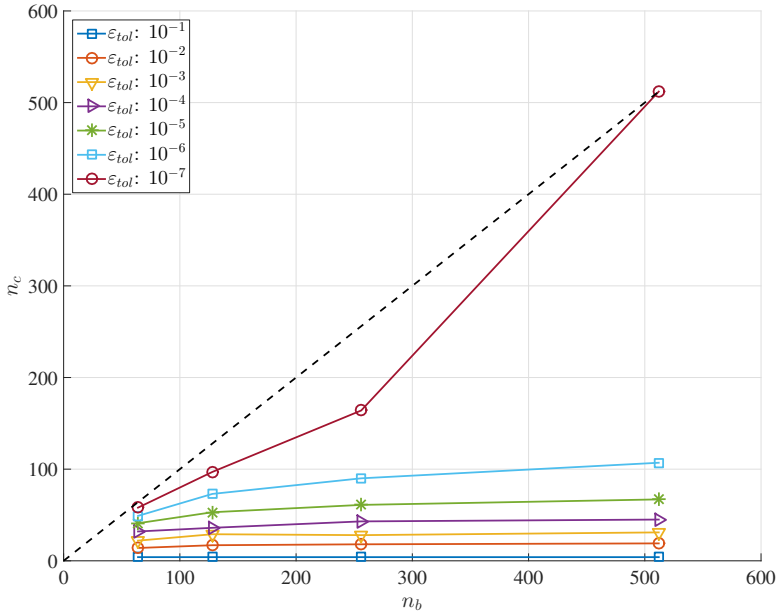

Figure 3.8: Number of control points for different selection tolerances (ε_{tol}) with a increasing mesh size. The dashes line indicates the maximum number of control points (i.e. the number of boundary points). Results are obtained while using the WendlandC2 function with a radius equal to 20 times the chord.

Reynolds number of $1 \cdot 10^6$ is used, having an aspect ratio of 637 and a first cell height of $2 \cdot 10^{-5}$ times the chord. To verify the analysis of the function dependency, the airfoil with flap case is used. In Figure 3.9 the normalized mesh quality is shown for varying selection tolerance and correction radius for 4 correction functions: WendlandC0, WendlandC2, GillebaartR3, GillebaartR3a. The normalized mesh non-orthogonality is:

$$\overline{\lambda}\left(\varepsilon_{tol}, R_{corr}\right) = \frac{\lambda_{corr}\left(\varepsilon_{tol}, R_{corr}\right) - \lambda_{no-corr}\left(\varepsilon_{tol}\right)}{90.0 - \lambda_{no-corr}\left(\varepsilon_{tol}\right)}, \tag{3.41}$$

from which the normalized mesh quality metric is obtained:

$$Q_{mesh} = 1 - \overline{\lambda} \tag{3.42}$$

resulting in a mesh quality metric between 0 and 1, where 0 indicates an invalid mesh and 1 a mesh as good as the mesh prior to the boundary correction. This metric isolates the influence of the boundary correction algorithm on the mesh quality. The reduction of mesh quality for lower correction radius (bottom right corner of Figure 3.9) is because the radius is smaller than the error, resulting in an invalid mesh due to negative volume cells. Increasing the correction radius above the error magnitude shows that a good mesh quality is obtained. However, when the tolerance is too relaxed (above 10^{-2}) the larger correction radius decreases the mesh quality, even until an invalid mesh can be obtained. The different behavior of the correction functions can be observed in Figure 3.9. For stricter selection tolerance the aspect ratio limit is dominating the mesh quality. For the WendlandC0 this limit holds even for all tolerances. For the WendlandC2

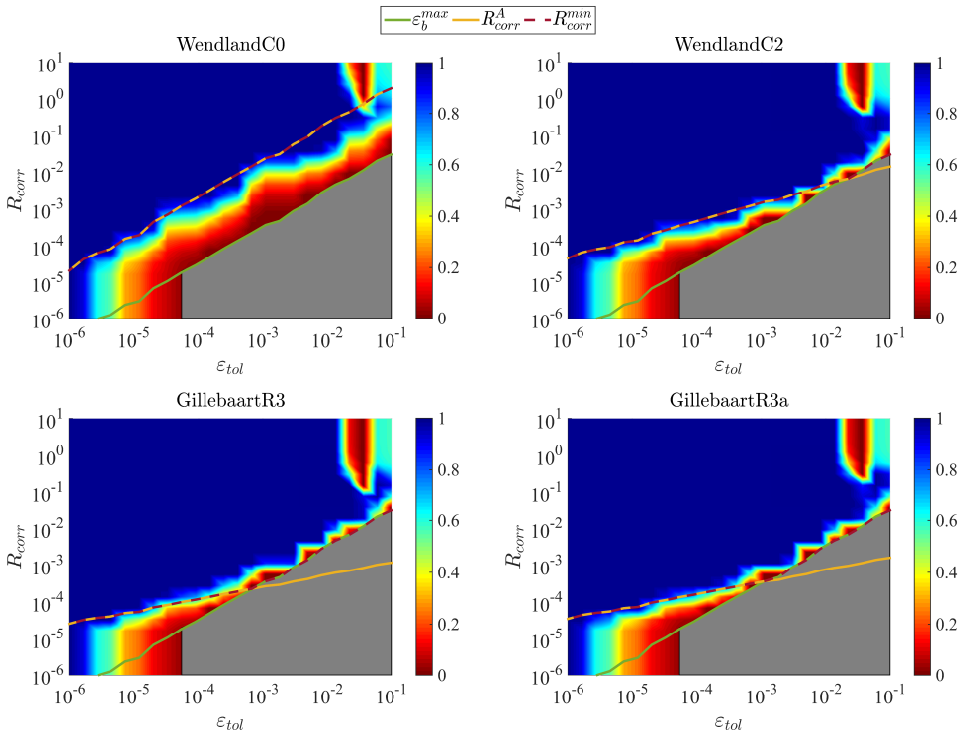

Figure 3.9: Normalized mesh quality for a airfoil with flap having a rotation, translation and flap deflection with an maximum aspect ratio of 650 and a first cell height of $6 \cdot 10^{-6}$ (an airfoil mesh for Reynolds number of 10^6). TPS is used for the RBF deformation, while varying the selection tolerance (x-axis), correction function and correction radius (y-axis). First line indicates the limiting line due to the boundary error (ε_b^{max}), the second line limiting line based on the analysis for maximum aspect ratio cells per correction function and the third line (dashed) the combination of the two. Four functions are shown: WendlandC0, WendlandC2 and the two newly derived GillebaartR3 and GillebaartR3a.

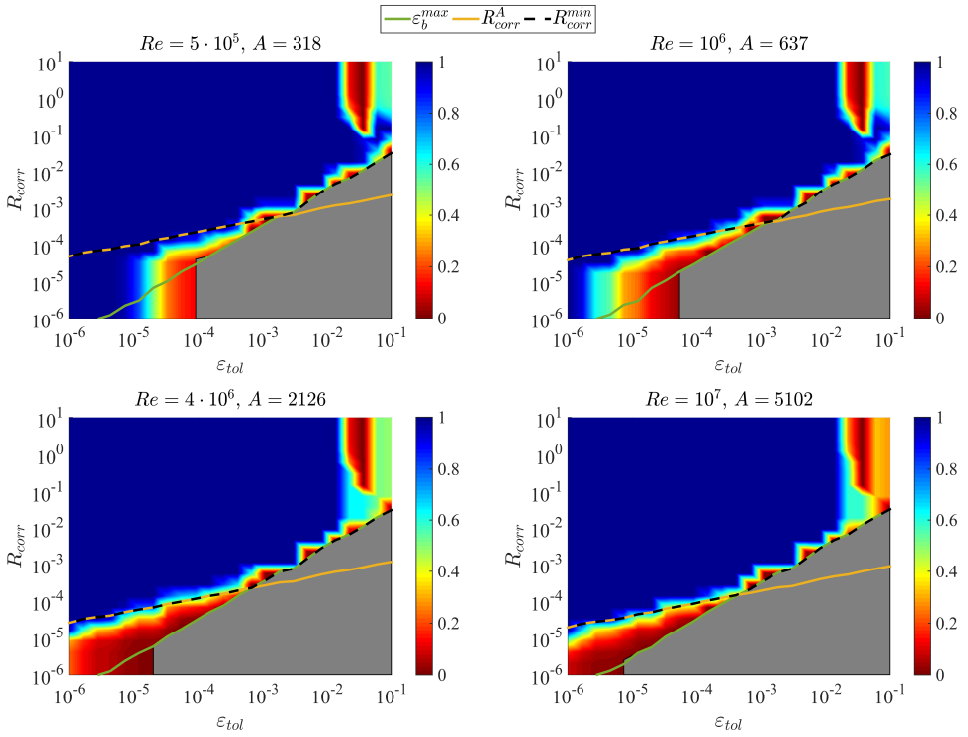

Figure 3.10: Normalized mesh quality for a airfoil with flap having a rotation, translation and flap deflection for four Reynolds numbers: $5 \cdot 10^5$, $1 \cdot 10^6$, $4 \cdot 10^6$ and $1 \cdot 10^7$. TPS is used for the RBF deformation, while varying the selection tolerance (x-axis) and the correction radius (y-axis). First line indicates the limiting line due to the boundary error (ε_b^{max}), the second line limiting line based on the analysis for maximum aspect ratio cells per correction function and the third line (dashed) the combination of the two. Results shown are for the function GillebaartR3.

this holds up to a selection tolerance of approximately 10^{-3}. For the two new functions the limit is dominant from a selection tolerance of 10^{-4}. It is clear that the proposed definition of the correction radius in Equation (3.36) is supported by the results presented in Figure 3.9, since the derived limit lines correspond well with the mesh quality limits in the numerical results. To further support the proposed correction radius function, three other meshes are used for different Reynolds numbers: $5 \cdot 10^5$, $1 \cdot 10^6$, $4 \cdot 10^6$ and $1 \cdot 10^7$ and their corresponding aspect ratios $318, 637, 2126, 5102$ and first cell heights $[4 \cdot 10^{-5},\ 2 \cdot 10^{-5},\ 6 \cdot 10^{-6},\ 2.5 \cdot 10^{-6}]$. Figure 3.10 shows the mesh quality for the same range of tolerances and correction radii, as in Figure 3.9. The correction function used is the GillebaartR3. The proposed limiting lines for the correction radius are shown in all plots. The results support the proposed equation for the correction radius in Equation (3.36). Combined with the results in Figure 3.9, the proposed equation for the correction radius is a robust and accurate method of determining the correction radius. To assess the required correction radius for the different functions, the mesh quality versus the correction radius is plotted for four different tolerances in Figure 3.11 for the mesh

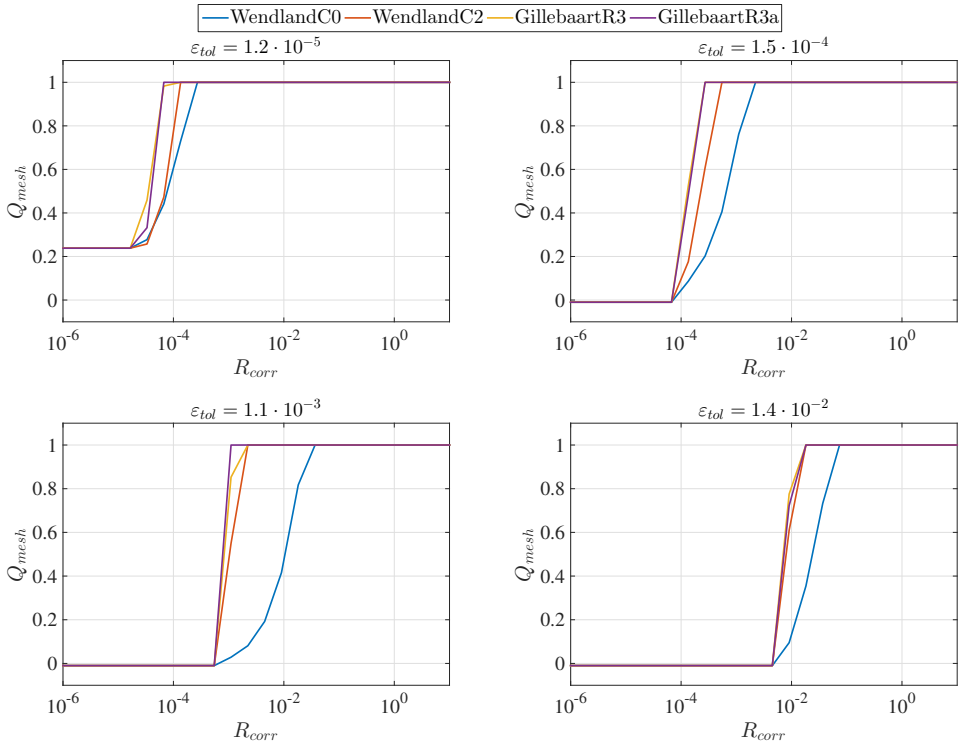

Figure 3.11: Normalized mesh quality for a airfoil with flap having a rotation, translation and flap deflection with an maximum aspect ratio of 650 and a first cell height of $6 \cdot 10^{-6}$ (an airfoil mesh for Reynolds number of 10^6). TPS is used for the RBF deformation correction function and correction radius (y-axis). Results are shown for four different tolerances: $1.2 \cdot 10^{-5}$, $1.5 \cdot 10^{-4}$, $1.1 \cdot 10^{-3}$ and $1.4 \cdot 10^{-2}$. Four functions are shown: WendlandC0, WendlandC2 and the two newly derived GillebaartR3 and GillebaartR3a.

associated with Reynolds number of $1 \cdot 10^6$. Here the effect of the aspect ratio limiting becomes apparent. For the functions with a higher power (GillebaartR3 and GillebaartR3a) the required correction radius is generally lower, especially for the stricter tolerances. Especially WendlandC0 behaves significantly worse over the whole range of tolerances used, while WendlandC2 behaves slightly worse than the Gillebaart functions. However, all of the functions do converge to the same mesh quality when a high enough radius is chosen.

In the proposed Equation 3.36 a single parameter is still undetermined: γ. To assess its influence on the mesh quality the selection tolerance (ε_{tol}) is varied together with the value γ. The resulting mesh quality can be found in Figure 3.12 for the 4 different correction functions. The four results for four different correction functions indicate that the choice for γ is correction function independent, as intended by means of the limits derived in section 3.2.5. Only WendlandC0 performs worse at higher selection tolerances. Based on the results above, we propose to use Equation (3.36) with $\gamma = 10$ in combination with either WendlandC2 or GillebaartR3a and a tolerance of approximately 10^{-2}, resulting in a robust and user-independent boundary correction method due to

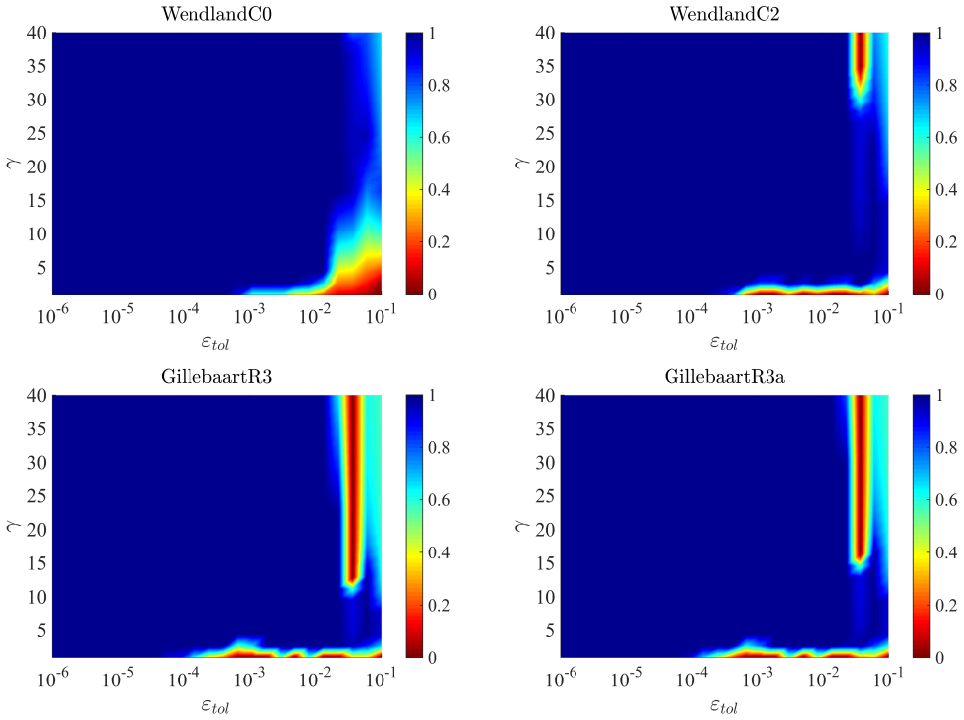

Figure 3.12: Normalized non-orthogonality for airfoil-flap motion for varying selection tolerance with boundary correction. Results for 4 functions are shown: WendlandC0, WendlandC2, GillebaartR3 and GillebaartR3a.

the robust choice of the correction radius.

3.3.2. ADAPTIVE SELECTION VERSUS UNITY GREEDY

Three different methods for selecting the subset of boundary points are discussed: unit displacement, prescribed displacement and actual (adaptive) displacement. First the 2D airfoil with a double-hinged trailing edge flap (TEF) is considered to assess the differences between the three methods, after which a 3D FSI tube case will be used to illustrate the behavior in a 3D internal flow case.

2D AIRFOIL WITH DOUBLE-HINGE FLAP

The motion of the airfoil and flap is the same as presented in the previous section, but are now simulated for a full period. In Figure 3.13 the maximum normalized boundary error is shown over time for the first period, for the three different approaches. From these

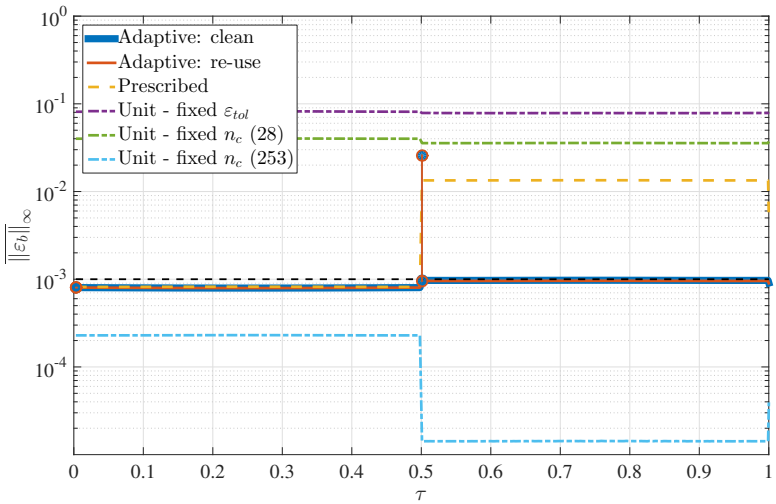

Figure 3.13: Normalized maximum boundary error of 2D airfoil with double-hinge trailing edge flap during a single period of motion for the adaptive algorithm, prescribed displacement selection (based upon maximum displacement in first half period) and unit displacement. For the unit displacement three cases are shown: the case where ε_{tol} is set as for the adaptive and prescribed cases and two cases where a fixed number of control points n_c is set. The n_c for these cases are equal to the n_c of the prescribed case or the n_c needed to obtain an actual boundary error below the given tolerance. The dots indicate the locations of re-selection and their corresponding old and new boundary error.

results it becomes clear that when the actual (i.e. real) displacement does not resemble a rigid body motion, the actual boundary error based on the unit displacement becomes significantly higher for the same ε_{tol}, which makes the method less robust as it requires an experienced user to make an educated guess for the tolerance. To clarify this further, the actual error versus the selection tolerance for the initial selection is shown in Figure 3.14 for both the unit displacement and the adaptive displacement. To illustrate how the unit displacement could result in a different error behavior, two extra simulations are performed: one where the number of control points selected by the unit displacement

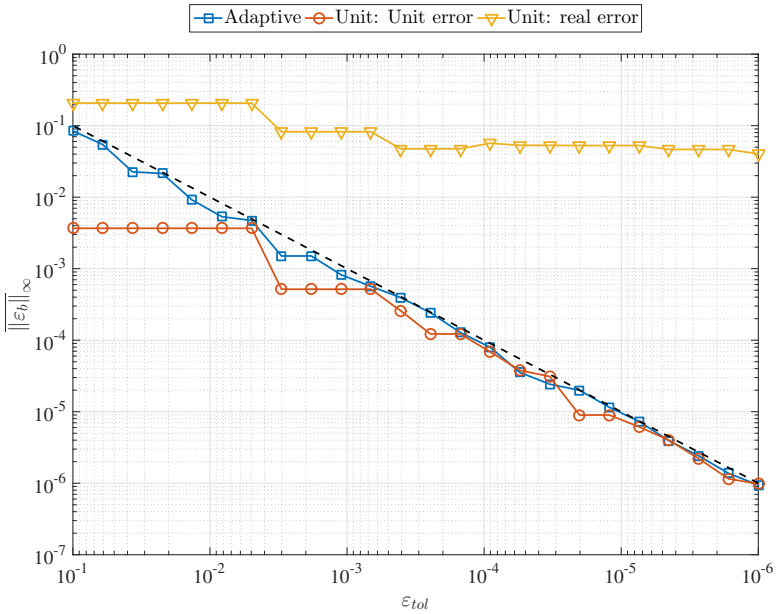

Figure 3.14: Normalized maximum boundary error of 2D airfoil with TEF versus the selection tolerance (ε_{tol}). The actual maximum boundary error is shown for the adaptive method and the unit displacement. For unit displacement also the unit displacement error (on which the selection is based upon) is shown.

is equal to the number of points selected by the prescribed displacement ($n_c = 28$), and one where the number of control points is chosen such that the unit displacement has an actual boundary error below the chosen tolerance ($n_c = 253$). Even though the error goes down when more points are selected, to get an similar boundary error as the adaptive method, one needs almost 10 times more points. This is mainly caused by the local deformation around the hinge, which the unit displacement does not capture until almost all points are used. Off course this behavior is highly case dependent and could be lower for different deformation types. This immediately illustrates the problem of the unit displacement: it is only robust for a limited amount of problems, while not resulting in the most efficient selection method.

In literature the proposed solution for this is to use a pre-processing step, where a typical deformation is prescribed to make a selection. However, this requires additional undesirable effort/user input (creating the typical displacement), while it still does not ensure that the boundary error remains below a certain criteria throughout the simulation. Even though the robustness is increased compared to the unit displacement, it is not as robust as the adaptive method. From the results in Figure 3.13 it becomes clear that the prescribed displacement works well for the first half of the motion, since it has a similar selection as the adaptive method. However, the adaptive method changes it selection based on the boundary error, which is required after the flap changes hinge. Here the increased robustness of the adaptive method becomes apparent. When periodic motions are considered, re-selection is best done by adding the newly selected points to the

previous selected control points. After the first period of motion, the control points are chosen such, that they do reflect a subset of points, which is able to represent all motions/deformations present in the structure. This is indicated by the line *Adaptive: re-use*. In the results shown in Figure 3.13, this results in a final number of control points of 35 opposed to the 28 control points used for the prescribed and adaptive method without re-use. In addition, notice how equally well the adaptive method and the prescribed method perform in the first part of the motion. Due to the choice of boundary error normalization, the set of control points selected by the adaptive method in the first step is very close to the set of points of the prescribed displacement used. This supports the approach of using the actual displacement both for selection and shape and normalization of the tolerance for a robust and adapting method.

One additional case is used to illustrate the robustness and efficiency of the adaptive method: the same airfoil with flap is used, but with the hinge at a fixed location half-way between the airfoil surface at $x/c = 0.8$. As show in Figure 3.15 the adaptive method and prescribed method works equally well, since no re-selection is needed throughout the period of motion. Additionally, the unit displacement performs similar as before: one needs to manually indicate the number of points to get to a low enough tolerance, which is significantly higher than with the prescribed or adaptive method. From these results, it is concluded that the adaptive method is equally expensive as the prescribed method when only a single selection is needed, while the adaptive method does add/re-selects points when required increasing its robustness significantly.

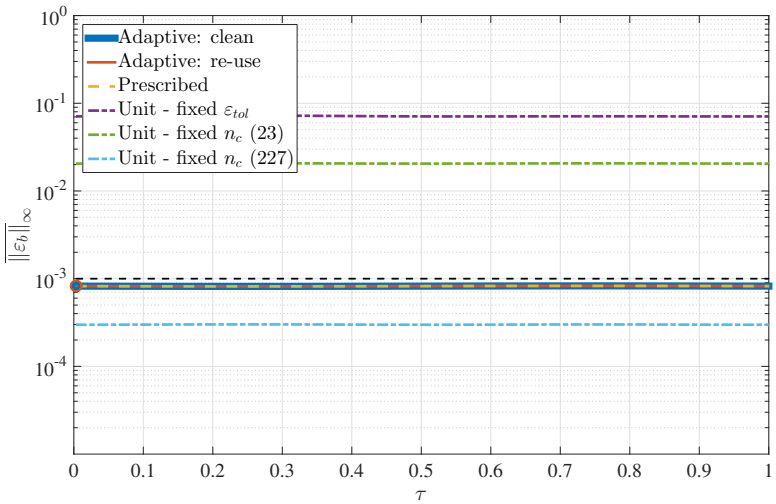

Figure 3.15: Normalized maximum boundary error of 2D airfoil with single hinge TEF during a single period of motion for the adaptive algorithm, prescribed displacement selection (based upon maximum displacement in first half period) and unit displacement. For the unit displacement three cases are shown: the case where ε_{tol} is set as for the adaptive and prescribed cases and two cases where a fixed number of control points n_c is set. The n_c for these cases are equal to the n_c of the prescribed case or the n_c needed to obtain an actual boundary error below the given tolerance.

3D FLEXIBLE TUBE

To show the applicability of the adaptive method to a 3D FSI simulation, a 3D flexible tube is considered, as described by Degroote et al. [22]. This case is chosen, because of the traveling local deformation along the length of the tube, as can be seen in Figure 3.16. The boundary error for the 3D tube case is shown in Figure 3.17. The number of control

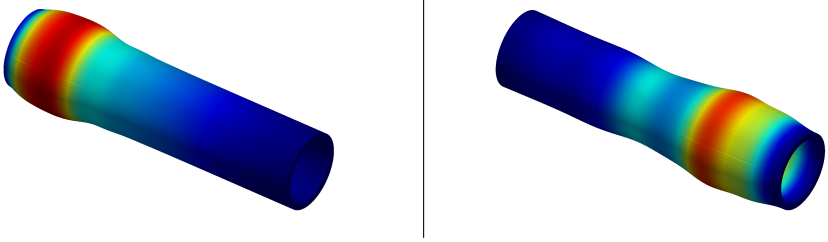

Figure 3.16: Displacement of the 3D flexible tube case scaled by a factor of 10. Left figure indicates the displacement at the 30% of the simulation and the right figure shows the scaled displacement at the end (100%) of the simulation.

points varies for the different methods and different selection tolerances, as shown in Figure 3.18. Here it can be seen using the unit displacement can result in similar boundary error as the adaptive boundary error, but with a higher number of control points. Using a prescribed motion here is more difficult, since the location of displacement is time varying. Using the adaptive approach ensures the number of control points to be smaller and at the same time ensures that the boundary error is always below a certain tolerance.

Results from both cases show that the adaptive method is both more robust as well as more efficient than the unit displacement and prescribed method. Robustness is inherently ensured, since the boundary error is not allowed to exceed a set threshold throughout the simulation. Efficiency is obtained by selecting a lower number of control points, while having the same accuracy as the other methods. Finally, the user dependency/input is limited to two tolerances, representing the boundary error and the (re-)selection tolerance, which is an improvement over the prescribed method, which required a complete deformation shape from the user, requiring significant more a priori knowledge.

3.3.3. PARALLEL SCALING: MEMORY AND CPU IMPLEMENTATION

With the mesh independency, boundary correction and the adaptive method, the number of control points needed is reduced significantly, while ensuring a high quality mesh deformation. However, the parallel scaling of the method compared to the unit and prescribed displacement remains to be assessed for the two cases: 2D airfoil with flap, and the 3D tube. At the same time, the difference between the CPU formulation and memory formulation has not been reported before in literature, but the memory formulation is potentially faster, especially in combination with the boundary correction and adaptive selection. To study both effects (parallelization and formulation) the above two cases are performed parallel up to 8 processors on a single node.

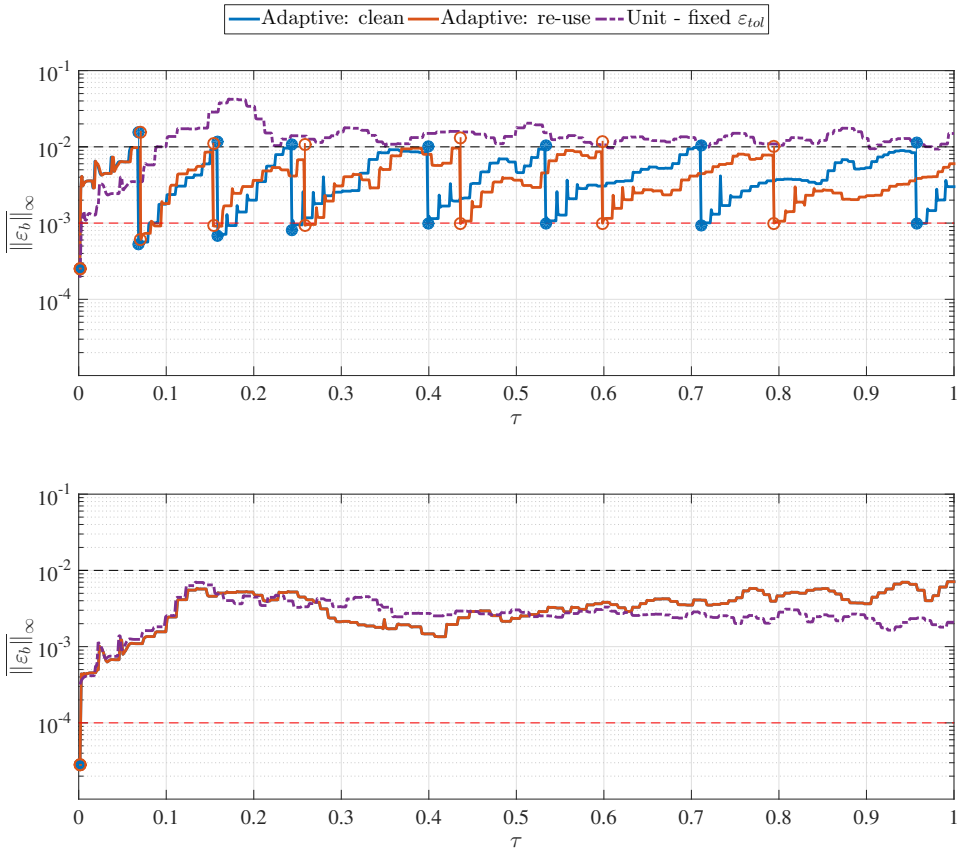

Figure 3.17: Maximum boundary error of 3D tube for two selection tolerances and three different methods: unit displacement, adaptive displacement and adaptive displacement with re-use. The top figure shows the results for a selection tolerance $\varepsilon_{tol} = 1 \cdot 10^{-3}$, while for the bottom figure $\varepsilon_{tol} = 1 \cdot 10^{-4}$. The adaptive methods use a re-selection tolerance of $1 \cdot 10^{-2}$ in both cases. Circles indicate a (re-)selection.

2D AIRFOIL WITH DOUBLE-HINGE FLAP

Four parts are identified in the proposed method: boundary displacement to internal displacement interpolation, boundary correction, (re-) selection (i.e. greedy selection) and boundary error calculation. Each of these four parts are timed during each of the simulations, which is performed for a single period with 1000 timesteps. For the 2D airfoil + flap case, the timings for different methods are shown in Figure 3.19. Each of the times are normalized by the total mesh deformation time of the CPU formulation on a single processor. As expected, the boundary error calculation and the (re-)selection do not scale, since they are implemented in serial. Where the boundary error calculation could be implemented in parallel (interesting for very large cases), the greedy algorithm is rather difficult because of the small full system of equations which need to be solved sequentially. However, the interpolation and boundary correction are scaling correctly for the CPU formulation, as shown more clearly by the parallel efficiency in Figure 3.20.

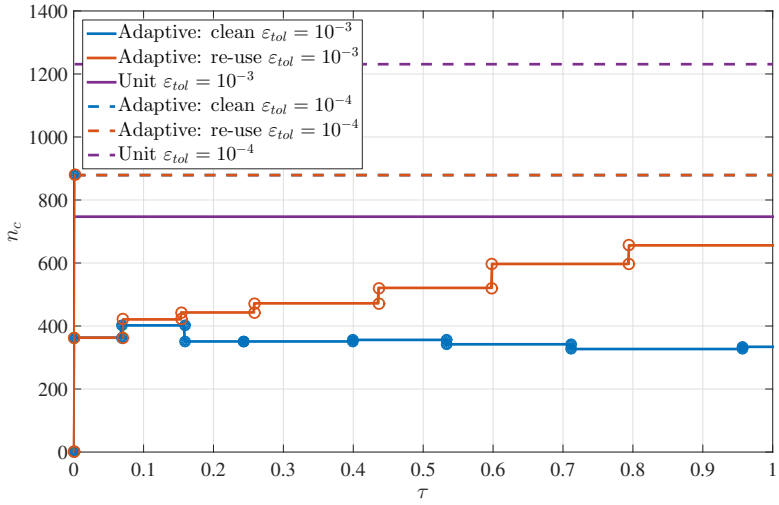

Figure 3.18: Number of control points n_c for two selection tolerances and three different methods: unit displacement, adaptive displacement and adaptive displacement with re-use. The adaptive methods use a re-selection tolerance of $1 \cdot 10^{-2}$ in both cases. Circles indicate a (re-)selection.

These are the main contributors to the computational work, since they are performed each time step. From this comparison it becomes clear that the unit displacement selection is significantly slower, due to the large amount of points required. The adaptive method performs equally well in terms of computational work as the prescribed method, with the benefit of being more robust. A slight increase can be found for the adaptive re-use method, because of the additional points used in the second stage of the simulation. However, for a periodic problem for which several periods need to be simulated, the method becomes more efficient.

Comparing the memory formulation with the CPU formulation, three conclusions can be drawn: 1) the major difference is found in the interpolation step, since the boundary error calculation, boundary correction and most of the selection is pure CPU work. 2) the memory formulation is significantly faster (up to 5 times). 3) the memory formulation does not scale as good, which is expected, since there is a limited bandwidth for communication between the CPU and shared memory, which is used by all cores on the CPU. However, this effect is only limited to a single node. When using multiple nodes, this effect will not further affect the performance, since the total bandwidth will increase with each node added to the computation. Depending on the hardware (especially number of cores per CPU and bandwidth) will determine when the CPU formulation becomes more efficient. In terms of efficiency, the results of this case seem promising. However, when the greedy algorithm becomes a larger part of the computational time, this will affect the efficiency significantly, as will be shown in the next section.

3D FLEXIBLE TUBE

For the 3D flexible tube, the work for three different methods (adaptive clean, adaptive re-use and unit) are shown in Figure 3.21. Results are obtained for a simulation of 200

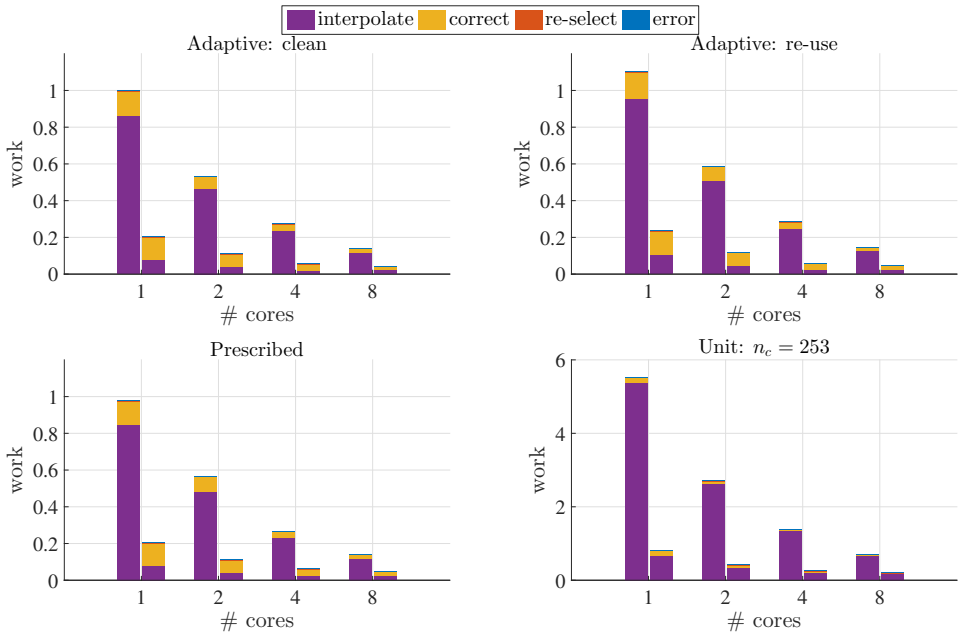

Figure 3.19: Work (normalized timings) versus number of cores of the 4 parts of the RBF mesh deformation algorithm for the 2D oscillating and translating airfoil with oscillating flap. Four different methods are shown: adaptive selection, adaptive re-use selection, prescribed selection, unit selection with similar boundary error. Timings are normalized by the total time of the CPU formulation results for the adaptive clean (re-)selection. Left bars of the pairs are for the CPU formulation and the right bars for the memory formulation.

time steps using a partitioned FSI approach. Again the adaptive method outperforms the unit displacement method, this time by almost a factor of two. Here again, the direct influence of the number of control points on the computational time becomes clear. With more than twice as much control points for the unit displacement, the *adaptive: clean* method is more efficient by only using approximately 400 points, which significantly reduces the cost of the interpolation step as well as the selection step. The *adaptive: re-use* method is slightly less efficient, because it has selected almost 800 points at the end of the simulation, causing both the interpolation and selection part to become more expensive. For such an extreme case as the 3D tube case, the *adaptive: clean* method performs the best both in robustness and in computational work.

Similar behavior is found between the CPU formulation and the memory formulation compared to the 2D airfoil with flap case: memory formulation is significantly faster (due to the interpolation step), but scales less favorable due to the bandwidth limitations. Due to the use of the TPS in this case, a larger difference is found between the computational work of the interpolation step, since the TPS is more expensive to compute. However, the calculations of the H_{red} becomes also more expensive for the TPS, resulting in a larger selection contribution for the memory formulation. For parallel scaling an important difference is found in the contribution of the (re-)selection to the total work. In this extreme case (with up to 7 re-selections) the greedy algorithm becomes the dom-

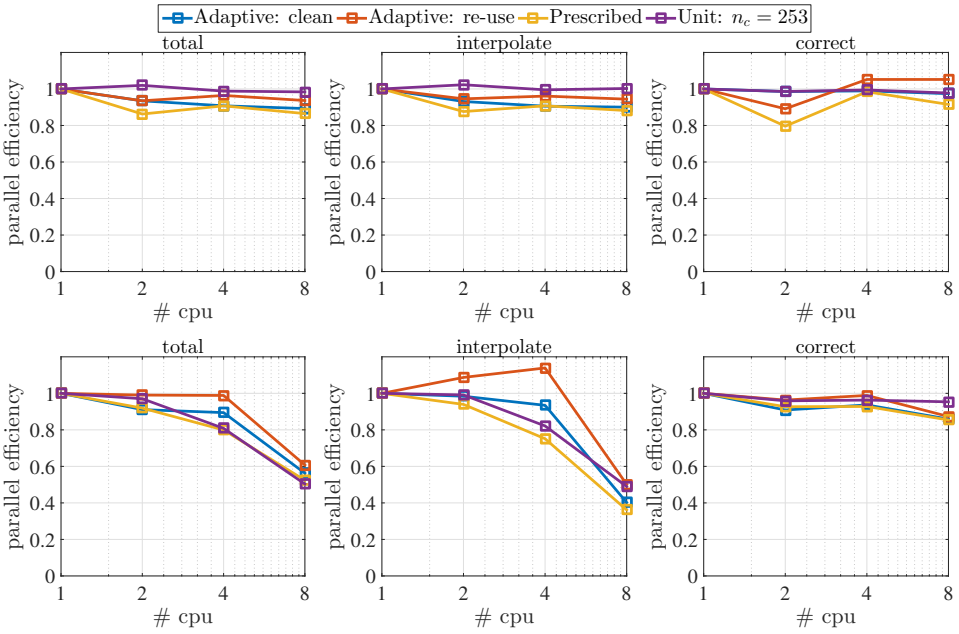

Figure 3.20: Efficiency versus number of cores for the 2 parallel implemented parts of the RBF mesh deformation algorithm for the 2D oscillating and translating airfoil with oscillating flap. Four different methods are shown: adaptive selection, adaptive re-use selection, prescribed selection, unit selection with similar boundary error. Efficiency is defined as the one over the time times the number of cores. Top row shows the results for the CPU formulation and the bottom row of graphs for the memory formulation.

inating part when scaling up, which is most clearly seen in the total parallel efficiency for 8 cores, shown in Figure 3.22. Purely looking at the two parallel implemented parts of the adaptive algorithm a good efficiency is found for the CPU formulation, while again the efficiency of the memory formulation is limited to the bandwidth of the communication channels between the CPU and memory. Compared to the 2D airfoil, the selection algorithm becomes a more dominating factor in the computational work, causing the parallel efficiency to drop. This is due to the large amount of small full systems to be solved, which are hard to parallelize. However, the unit displacement method does not perform better in terms of parallel efficiency, due to the larger number of control points required, increasing the cost of the single selection at the start of the simulation.

From both results, it becomes clear that the adaptive method is more efficient than the prescribed or unit displacement methods, since the adaptive method reduces the number of control points used. This directly decreases the computational work of the interpolation and correction parts, which are executed each time step. For periodic simulations the *adaptive: re-use* will be the most efficient method, since the final set of points needed will be obtained after the first period, after which the selection algorithm will not be used anymore for the remaining part and only the parallel scalable interpolation and boundary correction are executed.. In case of a more locally varying case, the *adaptive: clean* method will be the most efficient, because of its more optimal usage of the

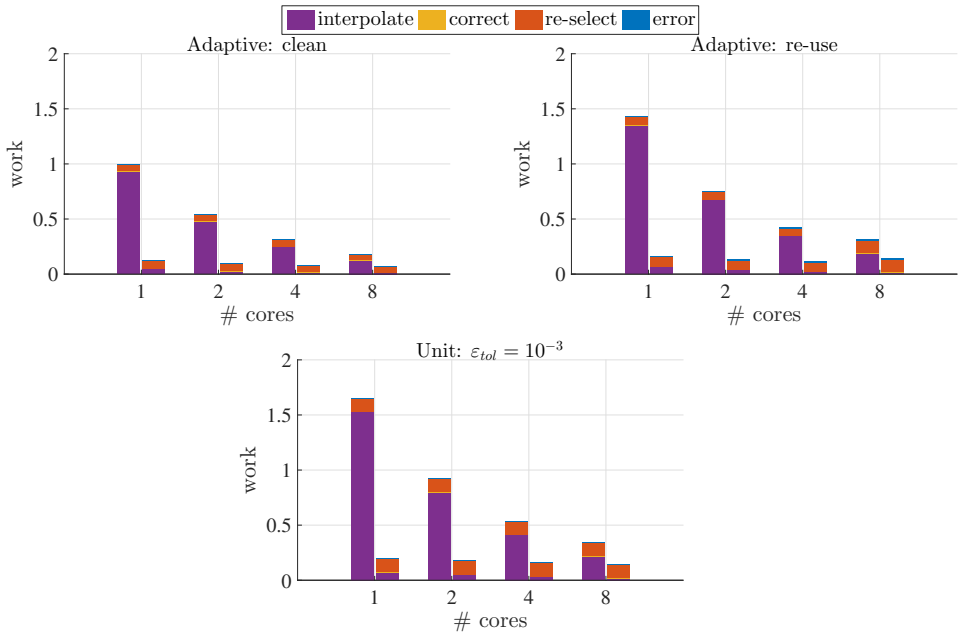

Figure 3.21: Work (normalized timings) versus number of cores of the 4 parts of the RBF mesh deformation algorithm for the 3D flexible tube. Three different methods are shown: adaptive selection, adaptive re-use selection and unit selection with the same selection tolerance. Timings are normalized by the total time of the CPU formulation results for the adaptive clean (re-)selection. Left bars of the pairs are for the CPU formulation and the right bars for the memory formulation.

selection algorithm, ensuring a significantly lower number of control points used over the simulation.

3.4. CONCLUSION & DISCUSSION

An adaptive RBF mesh deformation method which is robust, (parallel) efficient and user-independent (limited a priori knowledge needed) has been presented. The algorithm consists of the greedy algorithm in combination with an explicit boundary correction step. Based on a single cell analysis it is shown that the correction radius depends on the aspect ratio, first cell height, boundary error and the properties of the RBF function used. The order of $1-\phi(r)$ for $r << 1$ determines the effectiveness of the correction function. Therefore, two alternative functions are proposed that perform better within the domain where the properties of the correction function are dominant.

The proposed correction radius function is validated using a 2D oscillating and translating airfoil with oscillating flap for a range of Reynolds numbers (varying aspect ratio and first cell height), boundary error and correction radius. Good correspondence is found between the analytical derived function and the results for four functions: WendlandC0, WendlandC2, GillebaartR3 and GillebaartR3a. From these functions, WendlandC2 and GillebaartR3a perform best and are proposed as correction functions to be

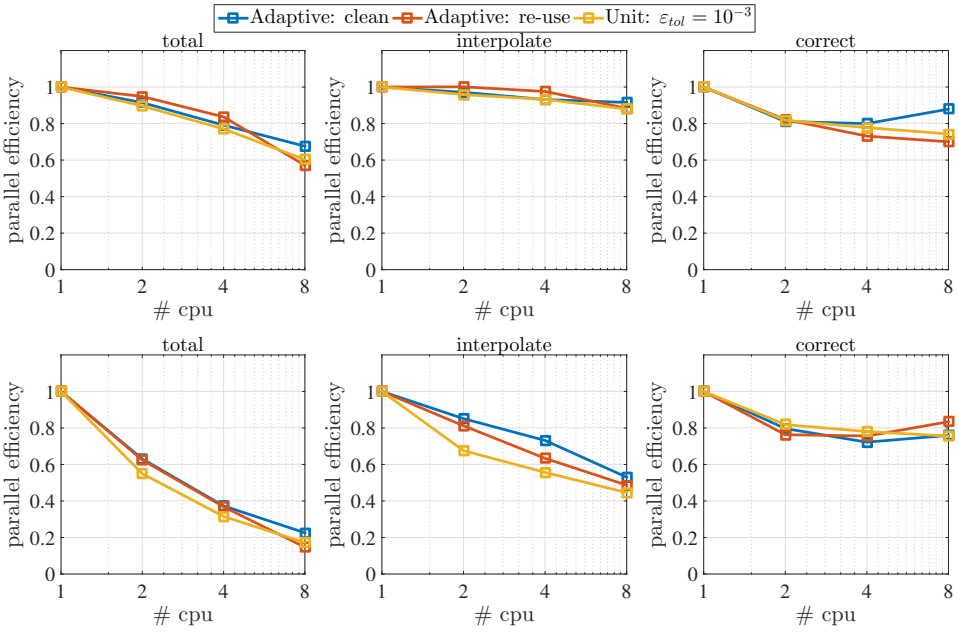

Figure 3.22: Efficiency versus number of cores for the 2 parallel implemented parts of the RBF mesh deformation algorithm for for the 3D flexible tube. Three different methods are shown: adaptive selection, adaptive re-use selection and unit selection with the same selection tolerance. Efficiency is defined as the one over the time times the number of cores. Top row shows the results for the CPU formulation and the bottom row of graphs for the memory formulation.

used within the adaptive method.

In combination with the mesh independency of the number of control points, an efficient method is obtained. This is demonstrated on a 2D oscillating and translating airfoil with oscillating flap and a 3D flexible tube problem. Results show that the adaptive method outperforms the unit displacement and prescribed displacement methods, both in terms of robustness and efficiency. All methods show a significant decrease in computational work when the memory formulation is used instead of the CPU formulation. The memory formulation is faster by a factor of 5 to 7 for the cases considered when performed on a single CPU. However, the parallel efficiency of the CPU formulation is better, due to the bottleneck of limited bandwidth between memory and CPU for the memory formulation.

In literature, different and potentially more efficient methods have been suggested for data reduction, such as the multi-level subspace approach [23], or a different approach by splitting up the problem in smaller problems by means of partition of unity [24, 25]. Both methods can be introduced in the presented framework, without affecting the proposed method in terms of robustness and user dependency. In terms of (parallel) efficiency, this will be the first step in future studies, to improve its parallel efficiency, while ensuring that the number of control points remains small. In short, the proposed adaptive radial basis function method, including the verified automated boundary cor-

rection, results in a robust, efficient and almost user-independent method, while still obtaining the high mesh quality known from the original/full radial basis function mesh deformation method.

REFERENCES

[1] M.-C. Hsu and Y. Bazilevs, *Fluid–structure interaction modeling of wind turbines: simulating the full machine,* Computational Mechanics **50**, 821 (2012).

[2] T. Gillebaart, A. V. Zuijlen, and H. Bijl, *Aerodynamic analysis of the wing flexibility and the clap-and-peel motion of the hovering DelFly II,* Proceedings of the International Micro Air Vehicles conference 2011 summer edition , 92 (2011).

[3] P. Lamata, I. Roy, B. Blazevic, A. Crozier, S. Land, S. A. Niederer, D. Rod Hose, and N. P. Smith, *Quality metrics for high order meshes: Analysis of the mechanical simulation of the heart beat,* IEEE Transactions on Medical Imaging **32**, 130 (2013).

[4] C. Farhat, *CFD on moving grids: from theory to realistic flutter, maneuvering, and multidisciplinary optimization,* International Journal of Computational Fluid Dynamics **19**, 595 (2005).

[5] J. T. Batina, *Unsteady Euler algorithm with unstructured dynamic mesh for complex-aircraft aerodynamic analysis,* AIAA Journal **29**, 327 (1991).

[6] K. Kovalev, *Unstructured Hexahedral Non-conformal Mesh Generation*, Ph.D. thesis, Vruie Universiteit Brussel, Belgium (2005).

[7] B. T. Helenbrook, *Mesh deformation using the biharmonic operator,* International Journal for Numerical Methods in Engineering **56**, 1007 (2003).

[8] S. Huo, F. Wang, W. Yan, and Z. Yue, *Layered elastic solid method for the generation of unstructured dynamic mesh,* Finite Elements in Analysis and Design **46**, 949 (2010).

[9] A. Gaitonde, *A three-dimensional moving mesh method for the calculation of unsteady transonic flows* (University of Bristol Department of Aerospace Engineering, 1993).

[10] A. de Boer, M. S. van der Schoot, and H. Bijl, *Mesh deformation based on radial basis function interpolation,* Computers and Structures **85**, 784 (2007).

[11] T. C. S. Rendall and C. B. Allen, *Efficient mesh motion using radial basis functions with data reduction algorithms,* Journal of Computational Physics **228**, 6231 (2009).

[12] T. C. S. Rendall and C. B. Allen, *Parallel efficient mesh motion using radial basis functions with application to multi-bladed rotors,* International Journal for Numerical Methods in Engineering **81**, 89 (2010), arXiv:1010.1724 .

[13] T. C. S. Rendall and C. B. Allen, *Reduced surface point selection options for efficient mesh deformation using radial basis functions,* Journal of Computational Physics **229**, 2810 (2010).

[14] O. Estruch, O. Lehmkuhl, R. Borrell, C. D. P. Segarra, and A. Oliva, *A parallel radial basis function interpolation method for unstructured dynamic meshes,* Computers and Fluids **80**, 44 (2013).

[15] S. Jakobsson and O. Amoignon, *Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization,* Computers and Fluids **36**, 1119 (2007).

[16] G. Wang, H. H. Mian, Z.-Y. Ye, and J.-D. Lee, *Improved point selection method for hybrid-unstructured mesh deformation using radial basis functions,* AIAA Journal **53**, 1016 (2015).

[17] Y. Wang, N. Qin, and N. Zhao, *Delaunay graph and radial basis function for fast quality mesh deformation,* Journal of Computational Physics **294**, 149 (2015).

[18] P. M. Hartwich and A. Shreekant, *Method for perturbing multiblock patched grids in aeroelastic and design,* in *13th Computational Fluid Dynamics Conference* (1997) pp. 984–993.

[19] S. Turek and J. Hron, *Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow,* in *Fluid-Structure Interaction,* edited by H.-J. Bungartz and M. Schafer (Springer Berlin Heidelberg, 2006) pp. 371–385.

[20] T. Richter, *Goal-oriented error estimation for fluid-structure interaction problems,* Computer Methods in Applied Mechanics and Engineering **223-224**, 28 (2012).

[21] K.-J. Bathe and G. A. Ledezma, *Benchmark problems for incompressible fluid flows with structural interactions,* Computers & Structures **85**, 628 (2007).

[22] J. Degroote, P. Bruggeman, R. Haelterman, and J. Vierendeels, *Stability of a coupling technique for partitioned solvers in FSI applications,* Computers & Structures **86**, 2224 (2008).

[23] H. H. Mian, G. Wang, Z.-y. Ye, and M. A. Raza, *RBF interpolation with improved data reduction algorithm – A Meshfree method for fluid- structure coupling and mesh deformation,* in *Proceedings of 2014 11th International Bhurban Conference on Applied Sciences & Technology (IBCAST)* (Islamabad, Pakistan, 2014) pp. 234–242.

[24] H. Wendland, *Fast evaluation of Radial Basis Functions : Methods based on partition of unity,* in *Approximation Theory X: Wavelets, Splines, and Applications* (Vanderbilt University Press, 2002) pp. 473—-483.

[25] G. Wang, H. H. Mian, Z.-Y. Ye, and J.-D. Lee, *Parallel implementation of localized radial basis function interpolation for computational aeroelastic predictions,* in *Parallel Computational Fluid Dynamics* (Springer Berlin Heidelberg, 2014) pp. 11–24.

4

MODEL VALIDATION FOR AN OSCILLATING TRAILING EDGE FLAP

With consistent time integration and adaptive RBF mesh deformation, the URANS model is validated with experimental data. To create a controlled environment for validation, experimental data from a rigid 2D airfoil with oscillating flap are used. In addition to the high fidelity URANS model, a typical engineering model (based on the Beddoes-Leishman dynamic stall model) is applied in this validation study. A variation in angle of attack, reduced frequency, flap deflection amplitude and transition (fixed or free) creates a set with varying conditions encountered by TEF in wind turbines. Detailed comparisons are made between the lift, drag and moment coefficients, for both steady as unsteady results.

4.1. INTRODUCTION

In the past years smart rotors have gained significant interest, with the active Trailing Edge Flaps as one of the key developments [1]. Feasibility studies have shown the potential of the TEF for reducing fatigue and extreme loads [2, 3]. These initial studies employed Blade Element Momentum (BEM) methods, which ensure a low computational cost. However, in the more recent years an increase in the use of high fidelity models is observed [4–6], since a more detailed analysis of the concept is required to push it towards acceptance within the industry.

However, the question remains, what is the accuracy of the different models for simulating active TEF on wind turbine blades? Bergami et al. performed model comparison, by comparing an engineering model (dynamic stall model), vortex panel code with free wake and an incompressible URANS model for a variety of flow conditions and flap angles [7]. They observed that especially the steady-state solutions have a large influence on the unsteady results and difference between the models. However, experimental data are missing for a thorough validation study. Therefore, experimental data are needed to validate the various models under different conditions. In the past years several experiments have been performed. In 2010, Bak et al. studied the trailing edge flap concept experimentally to demonstrate, test and determine the potential of the concept [8]. They used 9% of the chord as flap controlled by a piezoelectric bender at a Reynolds number of $1.66 \cdot 10^6$. They varied the reduced frequency from the quasi steady regime (0.003) to the unsteady regime (0.181) up to the heavy unsteady regime (0.518). However, the flap amplitude was between -3^o and 1.8^o for the lower frequencies and only -2^o to 0.7^o for the highest frequency, while in most numerical studies flap deflections of -10^o to 10^o are considered. Despite of the small flap angles they did demonstrate the concept and also achieved significant load reduction when a 30^o phase difference between the pitching and flap motion was chosen. A more thorough experimental study on the influence of the flap motion on the forces has been performed by Baek et al [9]. In this study the flap was 10% of the chord and the flap deflection angle varied from $+/-10^o$ to $+/-20^o$ depending on the reduced frequency. Besides the flap angle amplitude, they varied the angle of attack from attached flow to stalled flow and the reduced frequency from 0 to 0.2. Only a selection of these combinations have been presented in the paper. Additionally the drag has only been presented in the steady cases, while the moment coefficient is not present for any of the cases. The UPWIND project also considered trailing edge flaps on wind turbine airfoils [10]. Besides the experiments from Baek et al., also steady flap experiments where performed at two different airfoils for different flap angles, angles of attack, and fixed and free transition. In the report C_l and C_d are given for all combinations. Very recently, Simão Ferreira et al. conducted an extensive experimental campaign within the AVATAR project [11, 12]. Herein, the angle of attack varied from 0° to 18°, flap angle amplitude varied from 5° to 10°, the reduced frequency from 0.01 to 0.1 and transition from tripped to free. During the experiments the unsteady pressure distributions are measured, from which the pressure lift, drag and moment are calculated. With this newly available experimental data, models could potentially be validated.

In this chapter, two models are validated with the AVATAR experimental data [11, 12]: an engineering model based on the Beddoes-Leishman dynamic stall model [13] and an incompressible URANS model. First the steady-state results are validated, after

which the unsteady results are assessed. In the first section the two models and the interpolation of the experimental flap deflections is discussed. After this the steady-state lift, drag and moment coefficients are compared for a range of flap angles. Next the unsteady data is compared. Finally, based on the comparison between the models and the experimental data, conclusions are made regarding the validity of the models for simulating oscillating trailing edge flaps on a wind turbine airfoil.

4.2. MODELING APPROACH

Two aerodynamic models are used: 1) an unsteady engineering aerodynamic model (based on Beddoes-Leishman dynamic stall model) and 2) an Unsteady Reynolds Averaged Navier-Stokes (URANS) model. Both models use the prescribed motion of an oscillating flap based on the experimental data from Simão Ferreira et al. [11]. Each of the two models is described below. In addition, the method of interpolating the flap deflection input data based on the available experimental data is shortly discussed.

4.2.1. ENGINEERING MODEL

The engineering model is the DU-SWAT model, described by Bernhammer in detail [14] and is partly based on the description by Bergami and Gaunaa [15]. The first part of this section will discuss the equations governing this model. The starting point of the model is the steady lift surface versus angle of attack (α) and the flap angle (β). This data can be obtained either experimentally or numerically. In the second part of this section, the input data and how it is obtained is discussed. Finally, a time-step study is performed to ensure that a proper time step is used to limit the time discretization error in the results.

GOVERNING EQUATIONS

Based on the angle of attack and the flap angle, the model predicts the unsteady lift, drag and moment coefficient. The lift coefficient (C_l^{st}) is separated into a linear, attached flow region over the airfoil (C_l^{att}) and a fully separated flow (C_l^{fs}) for the dynamic modeling of the lift. The relation for the lift coefficient is as given in Equation 4.1:

$$C_l^{st} = C_l^{att} f^{st} + C_l^{fs}\left(1 - f^{st}\right). \tag{4.1}$$

The attached lift coefficients are simply obtained by computing the lift curve slope as a function of α and β, as well as the zero lift angle of attack (α_0). The separation coefficient f^{st} is obtained by comparing the linear, attached lift coefficient C_l^{att} to the actual lift curve:

$$f^{st} = \left(2\sqrt{\frac{C_l^{st}}{C_l^{att}}}\right)^2. \tag{4.2}$$

This allows computing the fully separated contribution to the lift:

$$C_l^{fs} = \frac{C_l^{st} - C_l^{att} f^{st}}{1 - f^{st}}. \tag{4.3}$$

To move to a dynamic domain, the lift is split into a circulatory term ($C_{l,circ}^{dyn}$) and non-circulatory terms:

$$C_l^{dyn} = C_{l,circ}^{dyn} + \pi \frac{b_{hc}}{U_0}\dot{\alpha} + \frac{F_{dydx}}{\pi}\frac{b_{hc}}{U_0}\dot{\beta}, \tag{4.4}$$

where F_{dydx} is a term depending on the flap geometry as described by Gaunaa et al. [16]. The the circulatory term is determined by the attached flow lift coefficient and separated flow coefficient:

$$C_{l,circ}^{dyn} = C_{l,att}\left(\alpha_{eff}, \beta_{eff}\right) f^{dyn} + C_l^{fs}\left(\alpha_{eff}, \beta_{eff}\right)\left(1 - f^{dyn}\right). \tag{4.5}$$

To obtain the dynamic separation coefficient f^{dyn}, both potential lift and the steady separation coefficient are passed through a first order filter. Details on this procedure can be found in [14]. The effective angle of attack (α_{eff}) is obtained via the quasi-steady angle of attack (α_{qs}):

$$\begin{aligned} \alpha_{qs} = \alpha_{3/4} &= \alpha_{st} - \frac{\dot{y}}{U_0} + \frac{(0.5 - \epsilon_{EA})\, b_{hc}\dot{\alpha}}{U_0} \\ \alpha_{eff} &= \alpha_{qs}\Phi(0) + \sum^{N_{lag}} z_i^{\alpha}, \end{aligned} \tag{4.6}$$

where $\alpha_{3/4}$ is the angle of attack at 3/4 of the chord, y the vertical displacement, ϵ_{EA} is the distance from the aerodynamic center to the elastic axis, $U_0$is the free-stream velocity, b_{hc} the half chord, Φ is a lag function with experimentally obtained coefficients and z is a state variable depending on the downwash time history. For the flap deflection angle an analogue procedure is followed, which can be found in [14].

The drag and moment coefficients are also based on the steady data. They are the sum of individual components that can be obtained in a straightforward manner as shown by Bernhammer [14]. The drag coefficient is given as:

$$C_d^{dyn} = C_d^{eff} + C_{d,ind}^{\alpha} + C_{d,ind}^{\beta} + C_{d,ind}^{f} \tag{4.7}$$

$$C_d^{eff} = C_d\left(\alpha_{eff}, \beta_{eff}\right) \tag{4.8}$$

$$C_{d,ind}^{\alpha} = C_{l,circ}^{dyn}\left(\alpha_{qs} - \alpha_{eff}\right) \tag{4.9}$$

$$C_{d,ind}^{\beta} = C_{l,circ}^{dyn}\frac{\partial C_l \backslash \partial\beta}{\partial C_l \backslash \partial\alpha}\left(\beta_{qs} - \beta_{eff}\right) f^{dyn} \tag{4.10}$$

$$C_{d,ind}^{f} = \left(C_d^{eff} - C_{d,\alpha_0,0}\right)\left[\left(\frac{1 - \sqrt{f^{dyn}}}{2}\right)^2 - \left(\frac{1 - \sqrt{f^{C_{l,lag}}}}{2}\right)^2\right], \tag{4.11}$$

and the moment coefficient is given by:

$$C_m = C_m^{qs} + C_m^{nc,\dot{\alpha}} + C_m^{nc,\dot{\beta}} \tag{4.12}$$

$$C_m^{qs} = C_m\left(\alpha_{eff}, \beta_{eff}\right) \tag{4.13}$$

$$C_m^{nc,\dot{\alpha}} = -\frac{1}{2}\pi\frac{b_{hc}}{V_{local}}\dot{\alpha}^{str} \tag{4.14}$$

$$C_m^{nc,\dot{\beta}} = \dot{\beta}\left[-\frac{1}{2}\frac{b_{hc}}{V_{local}\pi}\left(G_{dydx} + 0.5F_{dydx}\right) + \frac{1}{2}\frac{1}{V_{local}}\left(\frac{F_y}{\pi} + \frac{H_y}{2}\right)\right] \tag{4.15}$$

where, F_y and H_y are coefficients depending on the flap shape. More detail on the governing equations can be found in the thesis of Bernhammer [14].

INPUT DATA

As seen in the description above, the steady-state lift, drag and moment are required as input for the engineering model. Currently, the steady-state experimental data from Simão Ferreira et al. [11] are used to generate the lift, drag and moment surface for angles of attack of $-20°$ to $20°$ and flap angles from $-10°$ to $10°$. From the known steady-state experimental data the surfaces are obtained by interpolation using Radial Basis Functions. When the steady-state results are presented, the lift, drag and moment coefficients including the viscous contributions (Cl, Cd and Cm) are used as input data. However, the unsteady experimental results only contain pressure lift, drag and moment. Therefore, for the unsteady simulations the steady-state pressure lift, drag and moment (Cl_p, Cd_p and Cm_p) are used as input data.

However, these data are not always readily available, being a big drawback of such an engineering model. Other methods to obtain the lift, drag and moment coefficients could be used, such as using other simulations tools (e.g. XFoil, CFD). Since for this validation study, the experimental data is used to create the input data, it is expected that the engineering model will perform well, especially in (near) steady-state conditions.

In addition to the lift, drag and moment input data, also flap shape dependent variables are required:

$$\begin{aligned} F_{dydx} &= -0.0245 \\ G_{dydx} &= -0.0093 \\ F_y &= -0.0020 \\ H_y &= -0.0079 \end{aligned}$$

These are obtained by the integration rules presented by Gaunaa et al. [16].

TIME STEP STUDY

To ensure the simulations are time-step independent a time-step study is performed for both attached and separated flow conditions. Here the time-step is coupled directly to the flap frequency and is set by specifying the number of time-steps per flap cycle. In Figures 4.1 and 4.2 the resulting lift and drag coefficients are shown for the different number of time-steps per cycle, respectively.

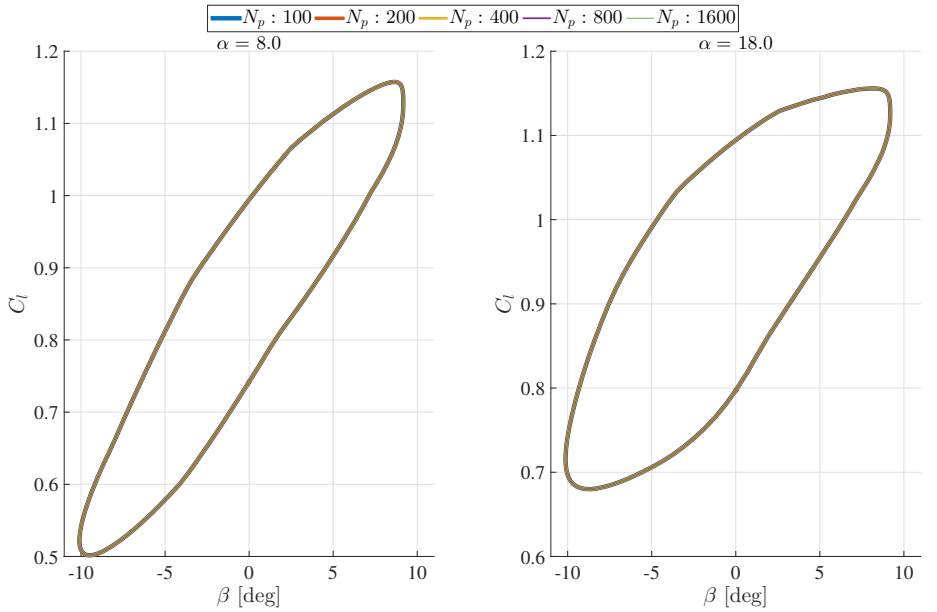

Figure 4.1: Periodic lift coefficient for 5 different number of time-steps per period (N_p) obtained with the unsteady aerodynamic model at two flow conditions: attached flow ($\alpha = 8.0°$) and separated flow ($\alpha = 18.0°$).

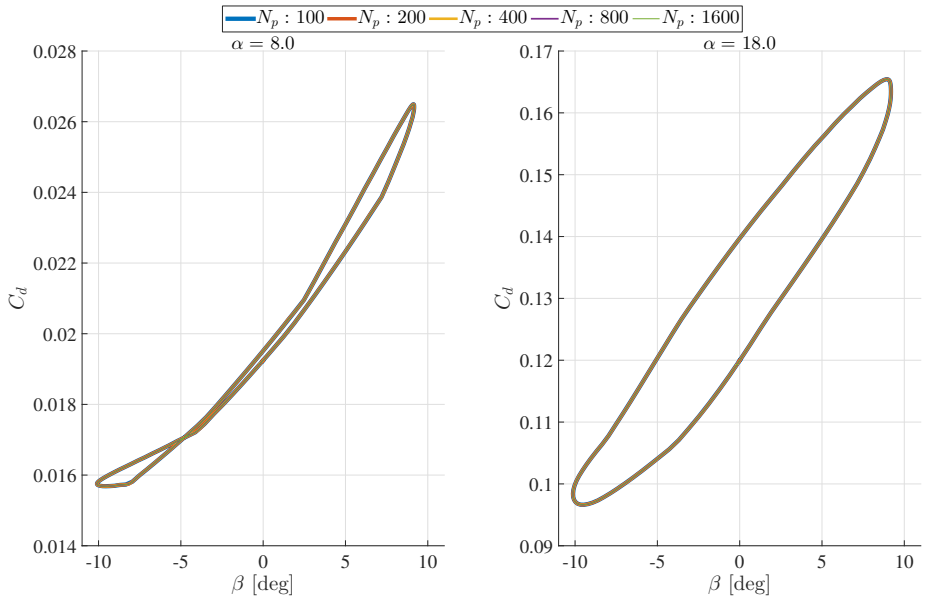

Figure 4.2: Periodic drag coefficient for 5 different number of time-steps per period (N_p) obtained with the unsteady aerodynamic model at two flow conditions: attached flow ($\alpha = 8.0°$) and separated flow ($\alpha = 18.0°$).

All lines coincide in Figure 4.1 and 4.2. Therefore, 200 time-steps per cycle are used for all simulations. This ensures the results to be time-step independent and calculations remain stable. With a lower number of time-steps per cycle instabilities did arise for some of the cases considered in this Chapter. This is most probably caused by the explicit time stepping method used (3^{rd} order Runge-Kutta), known to become unstable for larger time-steps.

4.2.2. URANS MODEL

To model the fluid around the airfoil and flap the incompressible Unsteady Reynolds Averaged Navier Stokes (URANS) equations together with the $k\omega$-SST turbulence model or Spalart-Allmaras turbulence model are used. The equations are written in Arbitrary Lagrangian Eulerian (ALE) form and discretized using the finite volume method as implemented in OpenFOAM [17]. The equations are solved using the consistent time integration for moving grids as described in Chapter 2. The URANS continuity and momentum equation in integral ALE formulation are as stated in Equations (2.1) and (2.2):

$$\begin{aligned} \int_{V_C} (\nabla \cdot \mathbf{u})\, dV &= 0, \\ \frac{\partial}{\partial t}\int_{V_C} \mathbf{u} dV + \oint_{S_C} \mathbf{n}\cdot(\mathbf{u}-\mathbf{u}_m)\mathbf{u} dS - \int_{V_C} \nabla\cdot(\nu\nabla\mathbf{u}) dV &= \int_{V_C} \frac{\nabla p}{\rho} dV. \end{aligned}$$

The mesh velocity is determined using the Discrete Geometric Conservation Law (DGCL) and the mesh deformation. The mesh is deformed based on the adaptive Radial Basis Functions as presented in Chapter 3, resulting in a robust and efficient way of deforming the fluid mesh.

As turbulence model $k\omega - SST$ model is used [18] with he updated presented in [19]. This model is often referred to as the SST-2003 turbulence model. This model is used, since it is considered to be one of the best for predicting airfoil aerodynamics. The model is governed by two transport equations for the turbulent kinetic energy k and the specific dissipation rate ω:

$$\frac{\partial \rho k}{\partial t} + \frac{\partial \rho u_j k}{\partial x_j} = P - \beta^* \rho\omega k + \partial x_j \left[(\mu + \sigma_k \mu_t) \frac{\partial k}{\partial x_j} \right], \tag{4.16}$$

$$\frac{\partial \rho \omega}{\partial t} + \frac{\partial \rho u_j \omega}{\partial x_j} = \frac{\gamma}{\nu_t} P - \beta\rho\omega^2 + \frac{\partial}{\partial x_j}\left[(\mu + \sigma_\omega \mu_t) \frac{\partial \omega}{\partial x_j} \right] \tag{4.17}$$

$$+ \quad 2(1-F_1)\frac{\rho\sigma_{\omega_2}}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j}, \tag{4.18}$$

from which the turbulent eddy viscosity is determined by:

$$\mu_t = \frac{\rho\alpha_1 k}{\max(\alpha_1\omega, SF_2)}, \tag{4.19}$$

$$S = \sqrt{2S_{ij}S_{ij}}. \tag{4.20}$$

For both production terms, a limiter is used:

$$P = \tau_{ij}\frac{\partial u_i}{\partial x_j}, \tag{4.21}$$

$$P = \min\left(P, 10\beta^* k\omega\right), \tag{4.22}$$

and the blending functions are given by:

$$F1 = \tanh\left(\min\left[\max\left(\frac{\sqrt{k}}{\beta^*\omega d}, \frac{500\nu}{\sqrt{d}\omega}\right), \frac{4\sigma_{\omega 2}k}{CD_{k\omega}d^2}\right]^4\right) \tag{4.23}$$

$$F2 = \tanh\left(\max\left[2\frac{\sqrt{k}}{\beta^*\omega d}, \frac{500\nu}{\sqrt{d}\omega}\right]^2\right) \tag{4.24}$$

with the modified (based on [19]) constants $\gamma_1 = 5/9$ and $\gamma_2 = 0.44$. A more detailed description of the turbulence model and its coefficients can be found in the original papers by Menter [18, 19]. A turbulence intensity of 0.01% is used, resulting in a fully turbulent flow. The Spalart-Allmaras model is used as well, to compare the influence of the turbulence model on the unsteady force predictions.

For the URANS model a time-step study is needed to ensure that the results are not influenced by the temporal discretization error. In addition, the URANS model also requires the spatial discretization error to be acceptable. Therefore both the time-step dependency and the mesh dependency is studied for the URANS model and the time-step dependency is studied for the engineering model.

MESH STUDY

The mesh independency is checked by using 4 meshes, each increasing by approximately a factor of 4 in number of cells. The first cell height is kept constant with a value of 2×10^{-5}, ensuring a y^+ near 1.0 all over the airfoil. In Table 4.1 the characteristics of the four meshes are summarized. For the mesh study, steady-state results are used at 8°

Table 4.1: Mesh characteristics of the four meshes used for the mesh study. Distances/lengths are given in chord lengths.

	mesh 1	mesh 2	mesh 3	mesh 4
$l_{farfield}$	90	110	105	100
$n_{points}^{farfield}$	64	128	256	512
$n_{points}^{airfoil}$	64	128	256	512
n_{points}^{wake}	5	10	20	40
n_{cells}	6784	27388	108094	429658

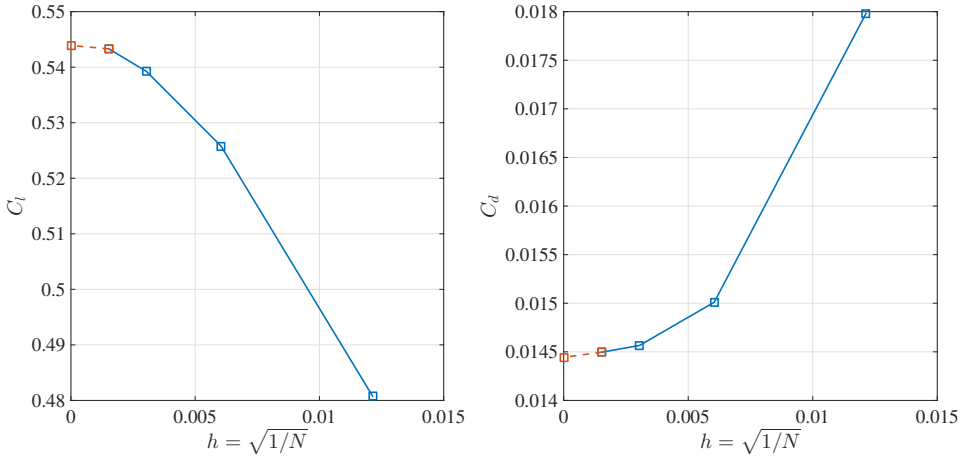

Figure 4.3: Lift and drag coefficient for 4 different meshes. Horizontal axis is average mesh spacing at the airfoil surface for the different meshes used, as summarized in Table 4.1. Estimated final force coefficients are obtained by Richardson extrapolation.

angle of attack with a fixed flap deflection of $-10°$. These conditions represent typical conditions encountered in the simulations performed for validation. Central differencing is used for all terms except the convective terms, for which linear upwind is applied. Sub-iterations are performed until a residual below 10^{-6} is reached. In Figure 4.3 the resulting lift and drag coefficients are shown for the four meshes. Based on these results, mesh 3 is chosen as mesh, having a difference below 1% compared to the extrapolated value.

TIME STEP STUDY

For the oscillating flap unsteady simulations are needed, for which the temporal discretization error should be minimized. Therefore, the time-step dependency will be assessed, ensuring a proper time-step is chosen, such that the results are time-step independent. Two time-step studies are performed, since two flow types will be encountered during the simulation of the different cases: attached flow and separated flow. For the attached flow the time-step influence will be dominated by the flap frequency, while for the separated flow it will be driven by the separation frequency encountered. In both cases the highest reduced frequency is chosen to assess the time-step dependency: 0.1. For the attached flow an angle of attack of 0° is used, while for the separated flow an angle of attack of 18° is used. Both cases will be simulated with four different normalized time steps ($\Delta\tau = \frac{\Delta t U}{c}$): 6×10^{-2}, 3×10^{-2}, 1.5×10^{-2}, 0.75×10^{-2}. In Figures 4.4 and 4.5 the resulting lift and drag coefficients for two different flow conditions are shown. From the results it is concluded that a normalized time-step of 3×10^{-2} is sufficiently small for the attached flow cases. Both the periodic lift and drag show insignificant differences for the time-steps considered. Therefore, the largest time-step is chosen. More optimal time step (i.e. larger time step) for the simulations of the attached flow are possible, but are not applied in this study. However, for the separated flow cases a normalized time-step

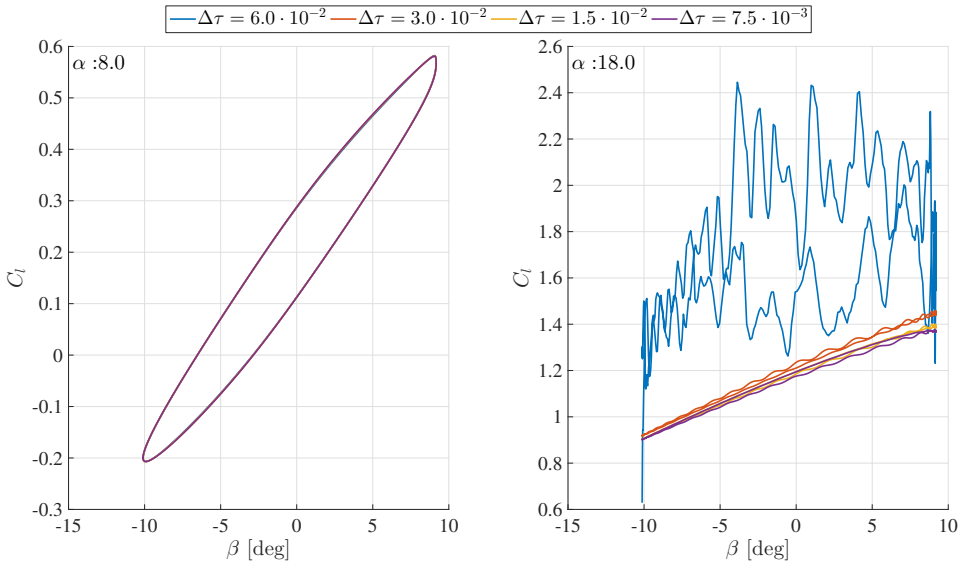

Figure 4.4: Periodic lift coefficient for 4 different time-steps obtained with the URANS model at two angles of attack: 0.0° (left) and 18.0° (right)

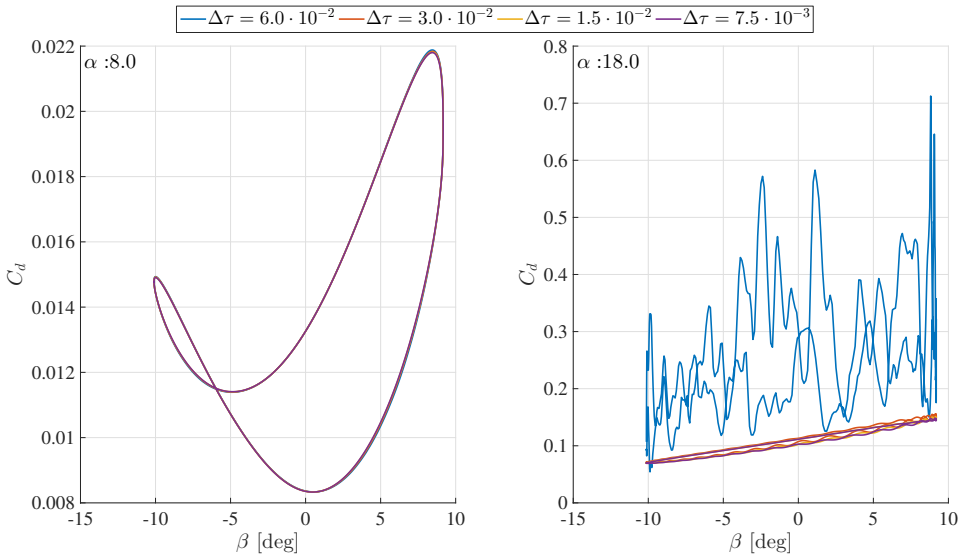

Figure 4.5: Periodic drag coefficient for 4 different time-steps obtained with the URANS model at two angles of attack: 0.0° (left) and 18.0° (right)

of 0.75×10^{-2} is used, since the lift and drag coefficient at the larger time-steps show significant differences. Especially, the results at the time-step of 6×10^{-2}, show unexpected and almost unstable behavior. This is due to bad convergence within each time-step, resulting in a solution which does not resolve the vortex shedding properly. From a time-step 3×10^{-2} a converged solution is achieved, but smaller differences are still present. Therefore, the smallest time-step is used for the cases of 18.0° angle of attack.

4.2.3. FLAP DEFLECTION INPUT

From the experimental data presented in Simão Ferreira et al. the flap deflection in time is available at discrete times during a single cycle. However, the numerical models used have a different time-step, requiring interpolation of the experimental data to ensure a consistent comparison/validation study. Since the motion is periodic, a Fourier Series interpolation is used:

$$\beta = a_0 + \sum_{n=1}^{N_{fourier}} a_n \cos(nt) + \sum_{n=1}^{N_{fourier}} b_n \cos(nt) \tag{4.25}$$

This interpolation is performed with $N_{fourier} = 5$, ensuring an accurate representation of the flap angle within 2 % of the experimental data, without introducing higher order oscillations within the numerical codes. Using less than 5 modes results in a too large deviation from the experimental data, as shown in Figure 4.6. Here, the angular displacement (on top left), its corresponding error (on top right) and the angular acceleration (on the bottom) of the flap is shown for varying number of Fourier modes used for interpolation. As can be seen, a higher number of modes decreases the error, but introduces dominantly higher frequencies in the acceleration. Since the pressure of an incompressible flow model directly depends on the acceleration, having these high frequency accelerations, might obscure the pressure distribution and thus the forces. Therefore, $N_{fourier} = 5$ seems a good trade-off between accuracy and preventing high frequency acceleration.

4.3. RESULTS

Both steady as well as unsteady experimental data are available. Both data sets are used to differentiate between the prediction capabilities of both models in a steady and unsteady regime. First, the steady-state responses are compared to the experimental steady-state values. Secondly, the unsteady experimental data is used to determine how well the aerodynamic models predict the unsteady behavior (without the steady-state influence).

4.3.1. STEADY-STATE RESULTS

First, the steady-state lift, drag and moment response of the URANS and the engineering model are compared to the experimental values. Five polars are available for the five flap deflections: −10.1°, −5.7°, −0.8°, 4.0° and 9.2°.

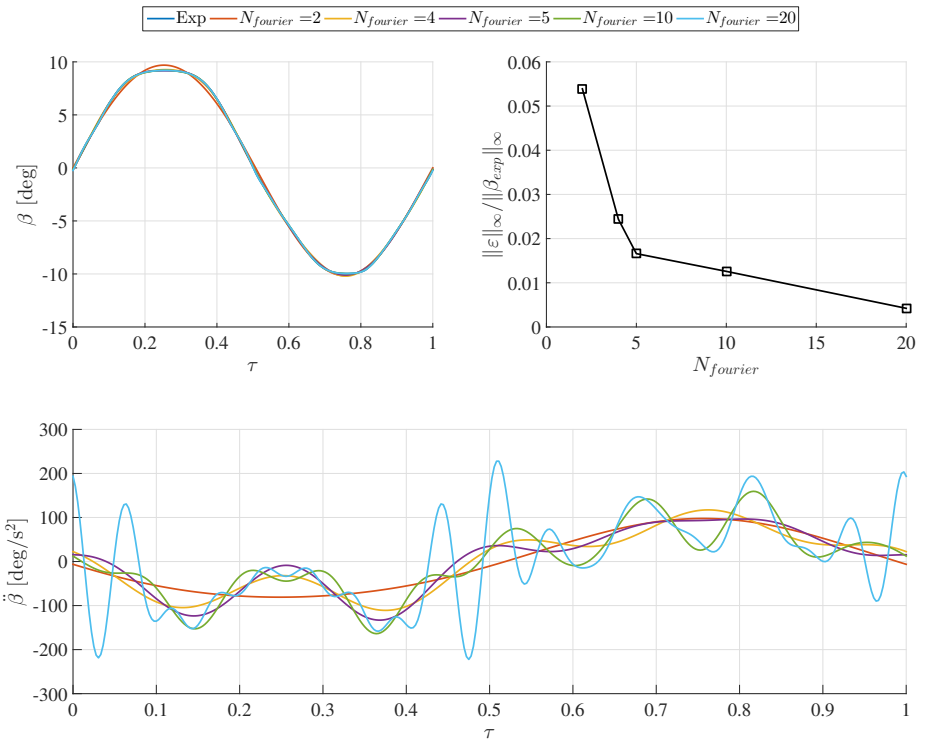

Figure 4.6: Displacement, displacement error and acceleration for the flap angle using varying $N_{fourier}$ for the Fourier series interpolation.

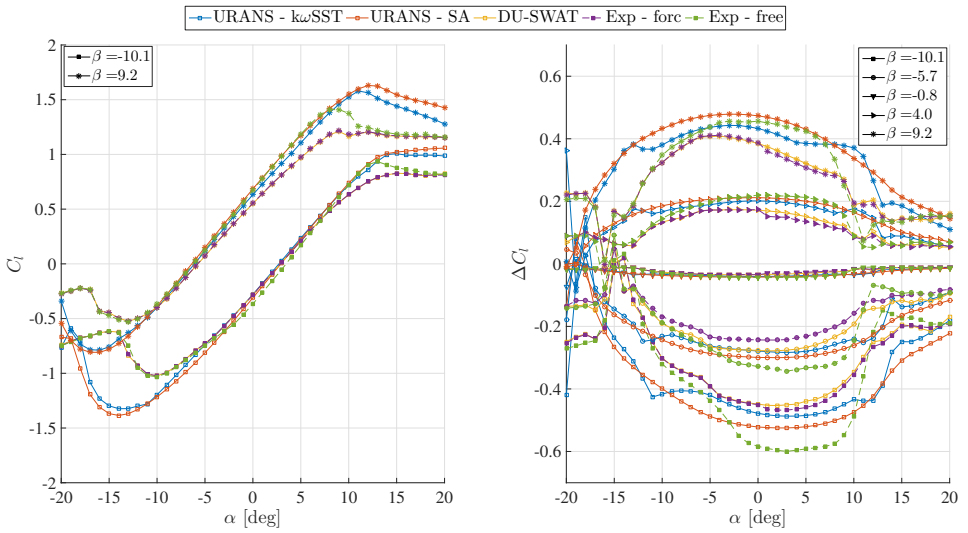

Figure 4.7: Lift coefficient versus angle of attack (α) for the two extreme flap deflections: $-10.1°$ and $9.2°$. The left graph shows the two extreme polars, while the right graph show all five polars as the difference with the (linearly interpolated) zero flap deflection lift coefficient polar.

LIFT COEFFICIENT

In Figure 4.7 the lift coefficient is shown for the URANS model with both the kω-SST turbulence model and the Spalart-Allmaras turbulence model. In addition, the interpolated steady-state data used within the engineering model is shown besides the experimental data. These data are obtained from the forced transition steady-state experimental data and thus good agreement is expected, while the URANS (high fidelity) model does not use any of the experimental results.

As expected, the engineering model corresponds well with the experimental values, since the input data of the model are based upon the experimental data. For the URANS model both turbulence models over-predict the maximum lift at all flap angles. In addition, the lift curve slope at the highest flap deflection ($\beta = 9.2°$) deviates from the experimental value. The experimental data show an early non-linear behavior, while the URANS data continues to show linear behavior up until an angle of attack around 7 to 8 degrees. Comparing the two turbulence models, results are as expected: the Spalart-Allmaras model over-predicts the maximum lift more than the kω-SST model. However, the free transition results from the same experiments show a better correspondence with the URANS results, especially near $\alpha = 8.0°$. In Figure 4.8 the pressure distributions are shown for the two URANS results and the experimental data for both forced (i.e. tripped) transition and free (i.e. natural) transition at $\alpha = 8.0°$ at a flap angle of $9.2°$. The forced transition pressure distributions shows a different behavior, especially on the suction side near the leading edge. First, the minimum pressure is significantly lower, after which this difference remains approximately constant up to the flap. Secondly, a dip in pressure is found between 5% and 10% x/c, which is where the trip zig-zag strip was located. This result suggests that at higher positive flap angles, the tripping of the boundary layer

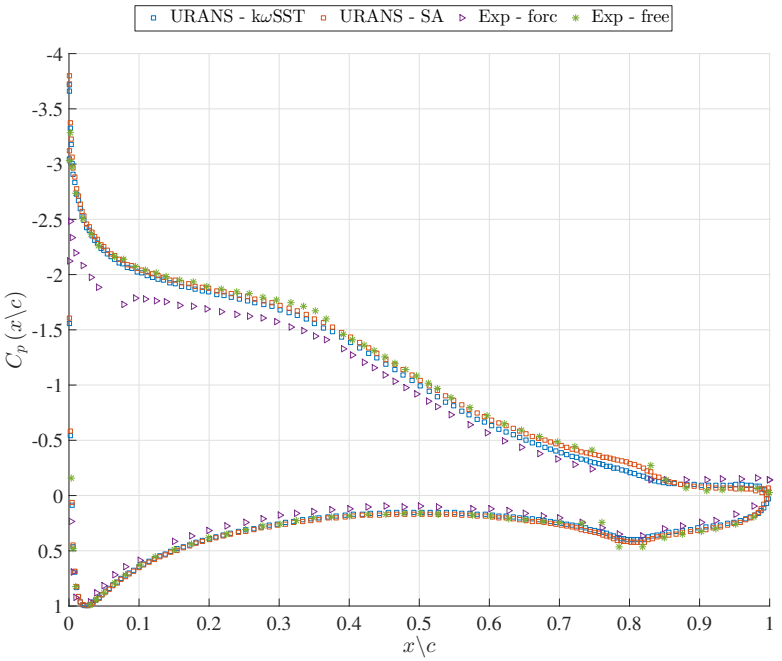

Figure 4.8: Pressure distributions for URANS model (kω-SST and Spalart-Allmars turbulence model) and experimental data (forced and free transition) at $\alpha = 0.0°$ and $\beta = 9.2°$.

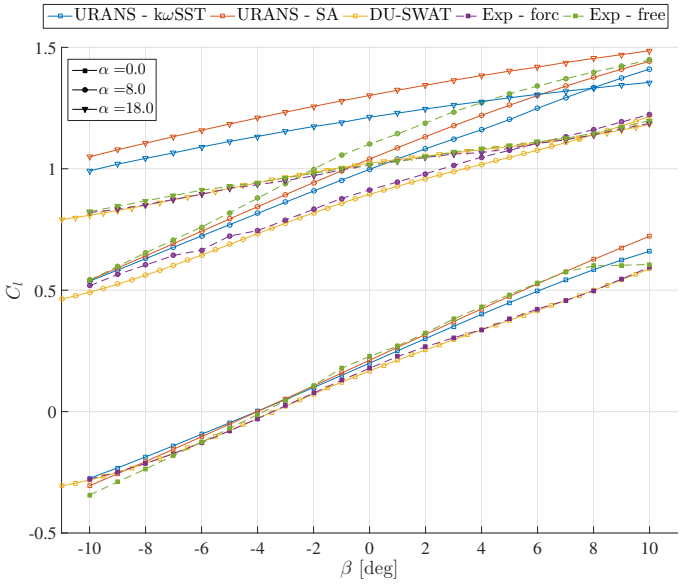

Figure 4.9: Lift coefficient versus flap angle (β) for three angles of attack: 0.0°, 8.0° and 18.0°.

might have caused significant differences in the pressure distribution and thus the integral forces. Further analysis of the experimental data and especially the setup should be performed to determine the cause of these differences. Finally, the URANS results do correspond very well with the free transition experimental data within the linear regime at higher positive flap angles.

A similar behavior can be found for the C_l versus β polars for three angles of attack: 0.0°, 8.0° and 18.0°, as shown in Figure 4.9. Where at lower angles of attack a good correspondence is found between the URANS results and the experimental values, at higher angles of attack both the effect of maximum lift over-prediction as well as the difference in lift slope is found. Also the URANS results at higher positive flap angles has a better correspondence with the free transition experimental data. However, at an angle of attack of 18.0° the differences remain large, which is expected for URANS models. The non-linear behavior in the fixed-transition experimental data for the angle of attacks of 0.0° and 8.0° is not as reported by other studies, and should be studied in more detail.

DRAG COEFFICIENT

Secondly, the drag coefficient is considered. In Figure 4.10 the drag coefficient versus angle of attack is shown for the engineering model, the URANS model with both the kω-SST turbulence model and the Spalart-Allmaras turbulence model and the experimental data for both the free and tripped transition.

First, it is clear that the engineering model, which uses the experimental data as input does show a good correspondence with the fixed transition data. It has a smoother response, especially near the minimum drag region, where the experimental data shows less smooth results. From the global overview figure it can be seen that the URANS model

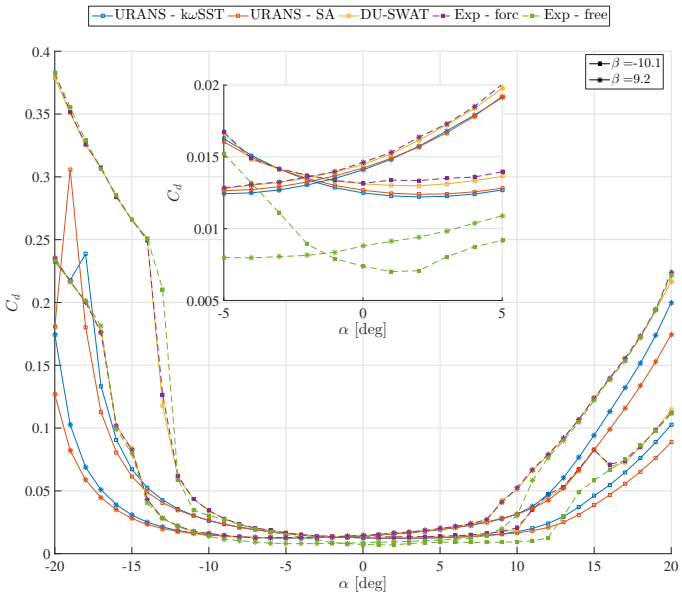

Figure 4.10: Drag coefficient versus angle of attack (α) for the two extreme flap deflections: $-10.1°$ and $9.2°$.

sharply increases at a higher angle of attack than the experimental values do. Within the zoomed in graph it can be seen that the trend corresponds well for both the turbulence models as well as the engineering model. The variation of the drag coefficient due to the flap deflection is shown in Figure 4.11, by showing the variation of the drag versus the flap angle (x-axis). Here it becomes evident that the minimum drag coefficients are under-estimated by approximately 10% for both the URANS model for both turbulence models when compared to the fixed transition experimental results. In addition, the drag is more significantly under-predicted at the angle of attack of 18.0°, although the slope is fairly well estimated at the higher angle of attack. The under-prediction is expected, because it is related to the delay in flow separation at the higher angles of attack, as seen in the lift coefficients results (see Figure 4.7).

A larger difference is found between the URANS results and the free-transition experimental results, which is caused by the laminar boundary layer present at the first part of the airfoil within those experiments, while the URANS results predict a fully turbulent boundary layer (causing a higher skin friction) over the complete airfoil length. This is strengthened by the results at higher angles of attack: here the free-transitions and fixed-transitions results correspond, since they are dominated by the pressure drag and an earlier transition in the free-transitions experiments.

MOMENT COEFFICIENT

Finally, the steady-state moment coefficients for the two models are compared with the experimental values. In Figure 4.12 the moment coefficient is shown for the two extreme flap deflections. Also the difference in moment coefficient with respect to the zero flap angle polar is shown at the right hand side.

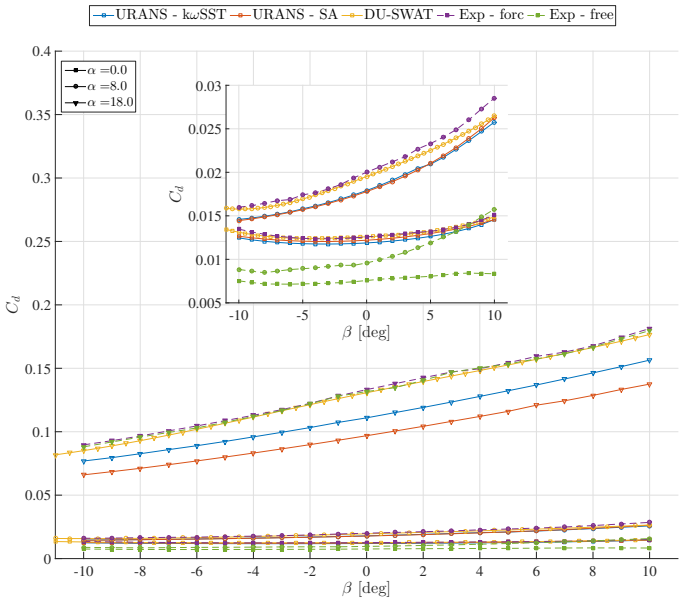

Figure 4.11: Drag coefficient versus flap angle (β) for three angles of attack: 0.0°, 8.0° and 18.0°.

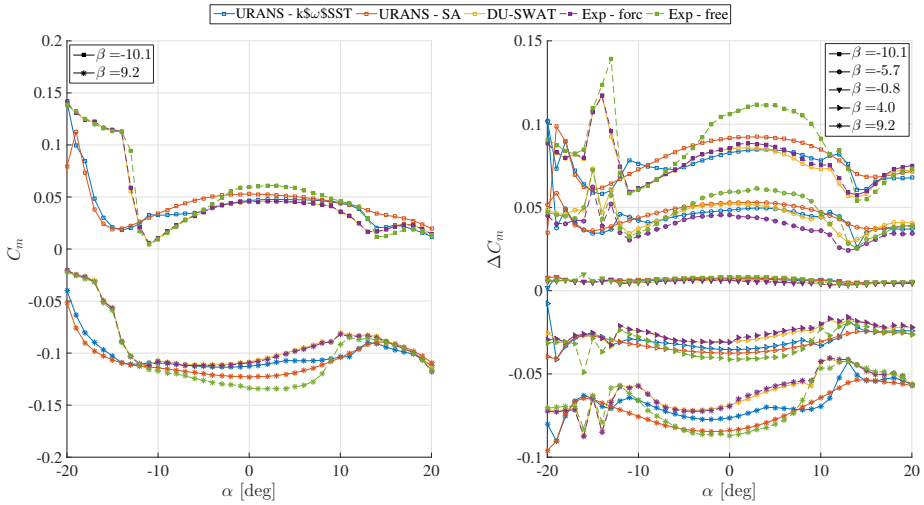

Figure 4.12: Moment coefficient versus angle of attack (α) for the two extreme flap deflections: −10.1° and 9.2°. The left graph shows the two extreme polars, while the right graph show all five polars as the difference with the (linearly interpolated) zero flap deflection moment coefficient polar.

In the attached flow regime a good correspondence is found, while for higher angles of attack the deviation increases. This is mainly caused by the difference in pressure distribution, which also lead to the difference in lift coefficient. The moment coefficient is closely related to the lift coefficient, especially at lower angles of attack. The kω-SST is better in predicting the steep change in moment coefficient around an angle of attack of $+/-10°$. The Spalart-Allmaras model over- or under-predicts the influence of the flap angle for negative and positive flap angles, respectively. This behavior is related to predicting the partial separation of the flow near the trailing edge.

To illustrate the differences for the variation in flap angle, two moment coefficient distributions are shown in Figure 4.13 and 4.14. Using the pressure distributions, the moment coefficient distributions are determined and plotted for the two experiments (free and forced transition) and the two URANS turbulence models. At the negative flap angle,

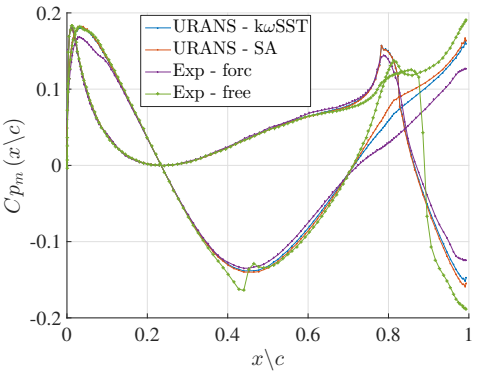

Figure 4.13: Moment distribution over the airfoil for the two experimental data sets and the results of the URANS model with two turbulence models. The angle of attack is 8.0° and the flap angle is −10.0°.

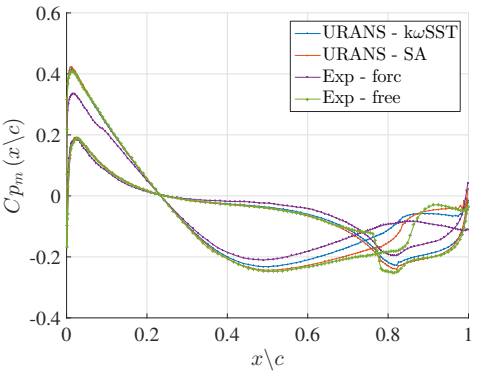

Figure 4.14: Moment distribution over the airfoil for the two experimental data sets and the results of the URANS model with two turbulence models. The angle of attack is 8.0° and the flap angle is 10.0°.

the correspondence of the moment coefficient distribution is good between the turbulence models and the experiments. The forced transitions shows a slightly smaller value near the LE, caused by the earlier discussed differences in the pressure distributions near the zig-zag strip. However, near the flap the increases are larger, mainly caused by the increase in arm. URANS results are in between the two experimental data sets. Again this leads to questions about the forced transitions experiments.

Finally, the moment coefficient distribution for the positive flap angle (10.0°) shows a larger difference between the forced transition experimental data and the turbulence model. As shown in Figure 4.8, this is caused by the large difference in pressure distributions prediction. Overall, the turbulence models, especially Spalart-Allmaras, corresponds well with the free-transition moment coefficient distribution. Also near the flap region the distributions only show small differences with the free transition data, while showing larger differences with the forced transition data.

4.3.2. UNSTEADY RESULTS

Within this section the unsteady lift, drag and moment coefficients from the two models are presented and discussed. To eliminate the steady-state offset shown in the previous section, the steady-state values at the specific angle of attack and flap angle are deducted from the unsteady results. With this approach, the unsteady force and moment variations due to the flap deflection are studied separately from the steady-state results, allowing to focus on differences in unsteady loads affecting the performance of the smart rotor. To obtain a measure of unsteadiness these polars are integrated over the flap deflection angle, for all variations in the angle of attack, reduced frequency and flap deflection amplitude ($\hat{\beta}$).

Based on the integral value of the different models, specific cases are discussed in detail below. To discuss the unsteady cases, the relative normalized unsteady force and moment coefficients are used. In Figure 4.15 both the original unsteady polars (left column) and the relative normalized unsteady polars (right column) are shown for the angle of attack of 8.0°, a reduced frequency of 0.1 and a flap deflection amplitude of 10.0°. All other original polars can be found in Appendix C, while the relative normalized unsteady polars can be found in Appendix D. These are obtained by subtracting the corresponding steady-state value from the graphs at the left side and normalizing the force and moment coefficient by the band of the corresponding experimental steady-state result. Consequently, the y-axis of the right side graphs indicates the unsteady variation relative to the steady-state variation. From the left column, the steady-state differences between the models and the experiments become apparent. However, when a closer look is taken at the unsteady results (right column) differences become more clear. For the lift the size of the loops correspond well, however, there is a clear difference in average slope. For the drag the lift graph already indicates the difference in size of loops, which is more clearly illustrated in the right graph. Here it becomes clear that the URANS models over-predict the unsteady drag the most, but also the engineering model (fed with steady-state experimental data) over-predicts the unsteady drag significantly. In addition, a similar change in average slope between the experiments and models is found. For the moment, the unsteady variation is relatively small, both seen in the left graph as well in the right graph by means of the vertical axis values: approximately 10% variation from the steady-state values. The engineering model fails to predict this variation. The URANS models predict a larger variation at positive flap angles and a smaller variation at negative flap angles, while the experiments show the opposite behavior.

In the remainder of this section, the unsteady lift, drag and moment will be discussed by assessing the integral values of the loops for varying reduced frequency, angle of attack and flap deflection amplitude. To assess the origin of the differences between URANS results and the experimental results, pressure and force distributions are used.

LIFT COEFFICIENT

To assess the accuracy of the two models for unsteady flap aerodynamics, the integral lift value is used and shown in Figure 4.16. In this figure, the integral of lift over the flap angle is shown for varying reduced frequency (x-axis). The angle of attack varies from top to bottom while the flap deflection amplitude varies from left to right.

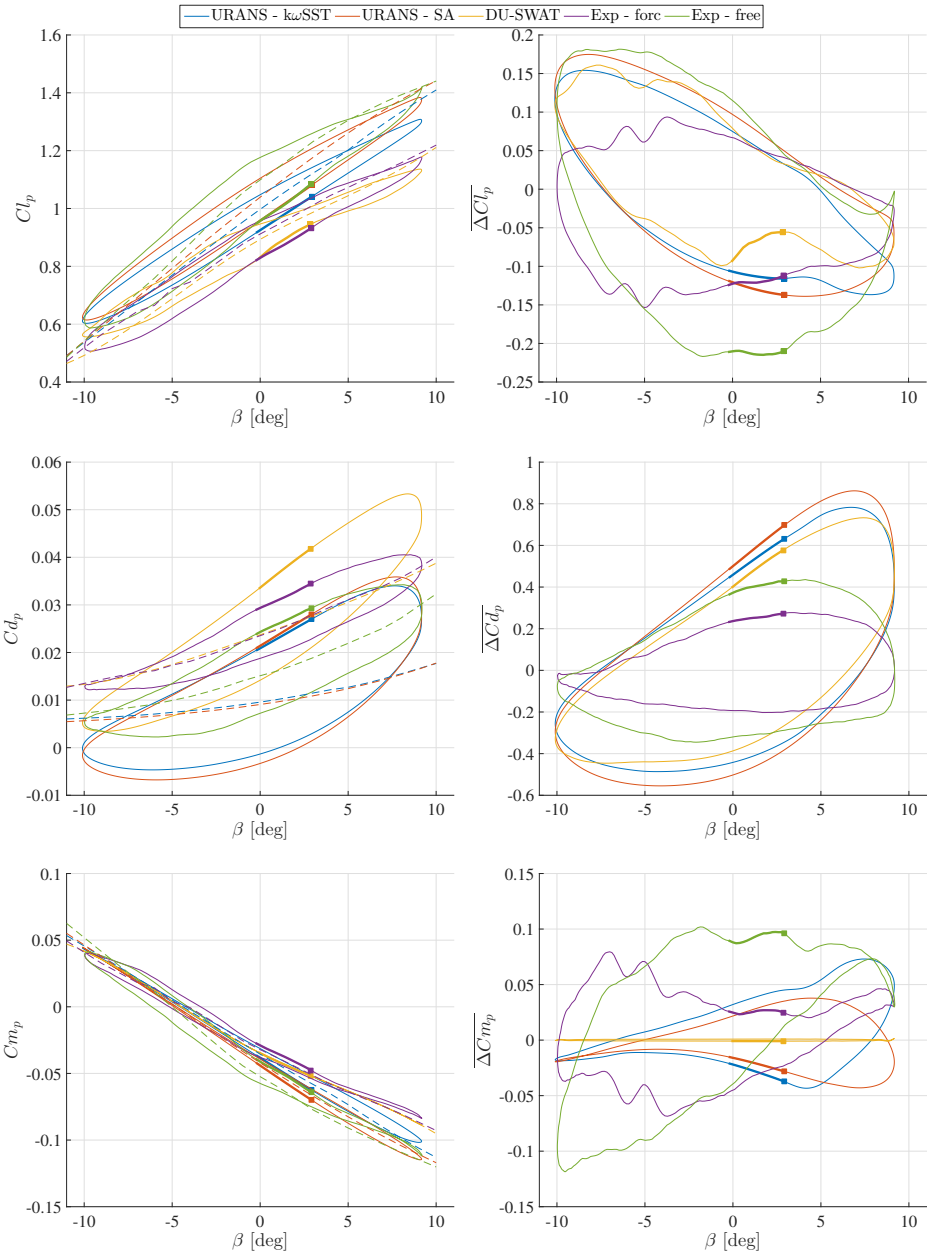

Figure 4.15: Unsteady absolute polars (Cl_p, Cd_p, Cm_p) at the left side and normalized unsteady polars ($\overline{\Delta Cl_p}$, $\overline{\Delta Cd_p}$ and $\overline{\Delta Cm_p}$) for $\alpha = 8.0°$, $k = 0.1$ and $\hat{\beta} = 10.0°$. Dashed lines at the left side represent the steady-state values for the corresponding model or experiment. The thicker line with square end indicates the direction of the loop, going from the beginning of the thicker line towards the square end.

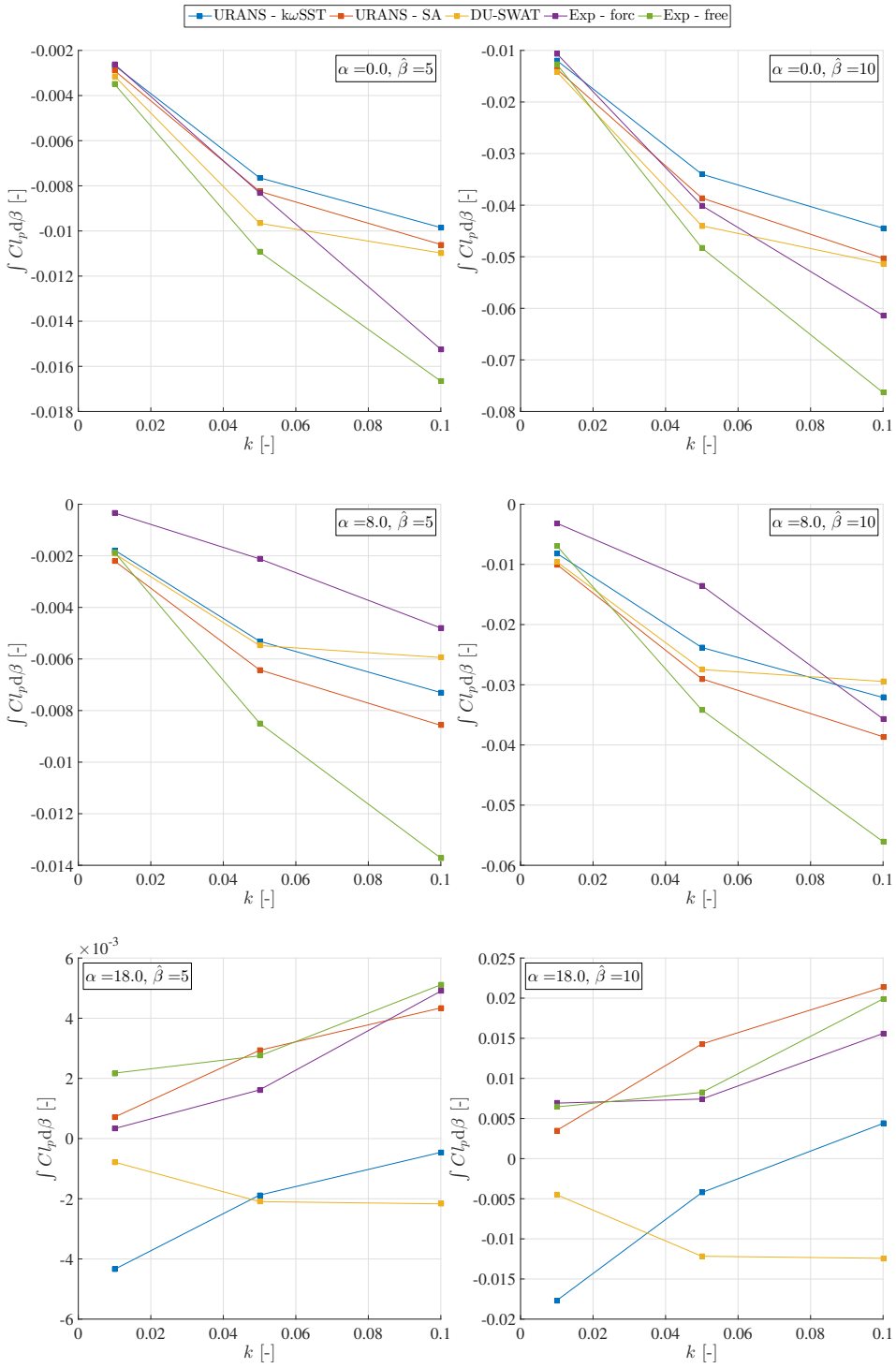

Figure 4.16: Integrated unsteady Cl_p at three angles of attack and two flap deflection amplitudes. $\int Cl_p \mathrm{d}\beta$ is obtained by integrating the unsteady lift coefficient for a single period over β.

From the results, it becomes clear that the models perform well compared to the experimental values at the two lower angles of attack, especially for the lower reduced frequency. However, for the zero angle of attack the models start to deviate for the highest reduced frequency. To study where these differences come from, Figure 4.17 shows the relative normalized unsteady lift polars for an angle of attack of 0.0°, $k = 0.1$ and $\hat{\beta} = 10.0°$. Here it can be seen that there are two main differences between the models

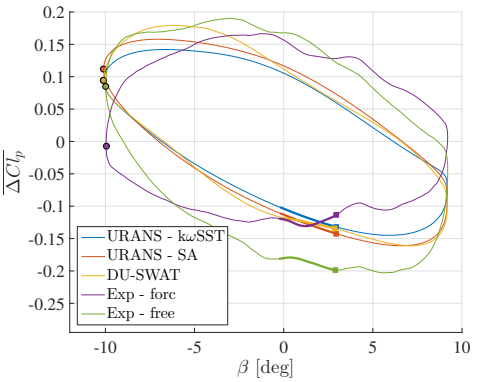

Figure 4.17: Unsteady normalized ΔCl_p for $\alpha = 0.0°$, $k = 0.1$ and $\hat{\beta} = 10.0°$. $\overline{\Delta Cl_p}$ is obtained by deducting the corresponding steady-state values at the corresponding angle of attack and flap angle for each model separately and normalizing this with the band of the experimental steady-state results. Dots indicate with black lining indicate the positions for which the lift and pressure distributions are shown in Figure 4.19 and 4.20.

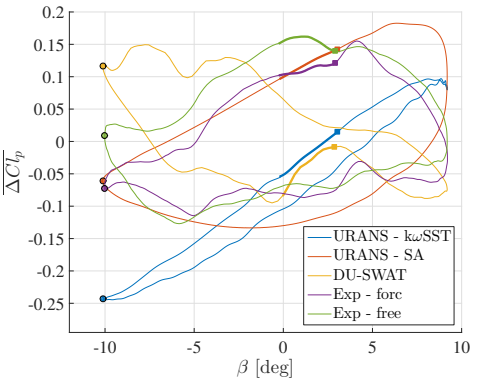

Figure 4.18: Unsteady normalized ΔCl_p for $\alpha = 18.0°$, $k = 0.1$ and $\hat{\beta} = 10.0°$. $\overline{\Delta Cl_p}$ is obtained by deducting the corresponding steady-state values at the corresponding angle of attack and flap angle for each model separately and normalizing this with the band of the experimental steady-state results.

and the experimental data. First of all the loop of the experiments are thicker, causing the magnitude of the integral value to be higher (i.e. integral value correspond to the area within the loop). Secondly, the average slope of the loop is negative for models, while the tripped experiments show a more neutral average slope. The free transition experimental data agrees better with the numerical model results considering the average slope, but shows a more pronounced unsteady influence (thickness of the loop).

The negative slope indicates a reduced flap effectiveness ($\frac{\partial C_l}{\partial \beta}$) w.r.t. to the steady-state results, since the average variation in lift due to the varying flap angle is reduced. Even though the maximum values are close to 0.2 (20% variation w.r.t. steady-state lift) the unsteady lift in the models flatten out the C_l versus β curve, as can be seen in Figure C.1. For a robust control strategy the flap effectiveness ($\frac{\partial C_l}{\partial \beta}$) is crucial, since it determine the amplitude of the flap deflection required to provide the desired force. The change in average slope can also be seen as a delay effect: the minimum and maximum deviations from the steady-state are not corresponding to the instance of maximum and minimum flap amplitude.

To further assess these differences, the lift distribution at the minimum flap deflection (−10.0°) of the Spalart-Allmaras results and the two experimental results are shown

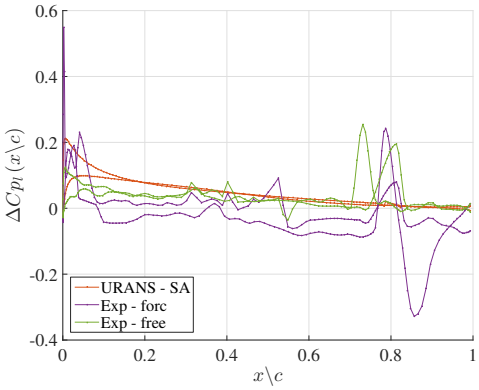

Figure 4.19: Delta lift distribution $\Delta Cp_l\,(x\backslash c)$ for $\alpha = 0.0°$, $k = 0.1$ and $\hat{\beta} = 10.0°$ at minimum flap deflection (indicated by black lined dot in Figure 4.17). $\Delta Cl_p\,(x\backslash c)$ is obtained by deducting the corresponding steady-state lift distribution from the unsteady lift distribution. The lift distributions themselves are obtained by decomposing the pressure distribution in lift and drag contribution.

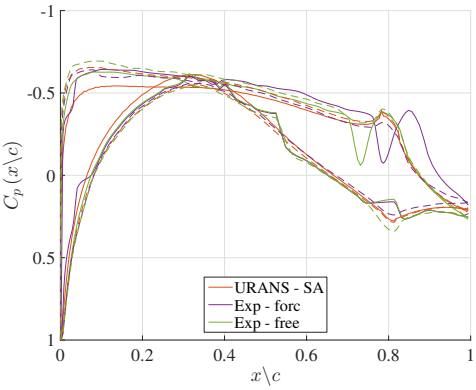

Figure 4.20: Pressure distributions $C_p\,(x\backslash c)$ for $\alpha = 0.0°$, $k = 0.1$ and $\hat{\beta} = 10.0°$ at minimum flap deflection. The full lines indicate the unsteady pressure distribution, while the dashed line show the corresponding steady-state pressure distribution.

in Figure 4.19. At the minimum flap deflection the numerical models and free-transition experiments predict a larger lift than the corresponding steady-state lift. URANS Spalart-Allmaras (and kω-SST, but omitted from the graph for clarity) lift distribution shows that the majority of the change in lift comes from the first half of the airfoil, while the increase in lift at the flap is minor. It could be said, that the lift disturbance due to the flap motion is directly felt by (and near) the flap, but requires time to travel towards the LE, causing a delay. The free-transition shows a similar pattern, although the distribution is obscured by other variations. For the forced-transition experimental results, this pattern is not found, which is inline with the observed lift variation.

However, the experimental variation in lift distribution shown in Figure 4.19 has irregularities near the leading edge (forced-transition), middle (free- and forced-transition) and mostly near the flap hinge ($x\backslash c = 0.8$). To assess these irregularities, the pressure distributions for the two experimental data sets and the Spalart-Allmaras results are shown in Figure 4.20. The dashed lines indicate the steady-state pressure distributions at the corresponding angle of attack and flap angle, while the full lines represent the unsteady pressure distributions. As expected from the lift distribution, the Spalart-Allmaras shows a smooth deviation from the steady-state values at the first half of the airfoil, while the steady-state and unsteady pressure distribution are similar near the flap. However, the two experimental data sets show significant disturbances near the flap hinge, especially the forced-transition results. In the experiments, a flexible seal was used to create a smooth transition from the airfoil to the flap. While the steady-state pressure distributions do show this smooth transition, the unsteady pressure distributions show a sharp change at the pressure side and an oscillatory behavior at the suction side. In addition, the forced-transition data shows sharp changes near the LE, near the zig-zag strip lo-

cation. Both these phenomena are unexpected and should be studied in more detail. Most likely, either the measurement setup (moving flexible seal close to pressure sensors) caused oscillating pressures or the post-processing of the unsteady data needs revision.

For $\alpha = 8.0°$ and flap deflection amplitude of 5.0° the differences originate from the same behavior. Both the average directions of the loop differ as well as the thickness of the loop differ. In Figure D.2 these results can be viewed in more detail. For the larger flap deflection amplitude (see top figures in Figure 4.15, also a similar trend is found. The most significant difference is found in the integral value of the forced-transition experiments, which shows a less pronounced difference with the models, indicating the thickness of the loop is similar. Concerning the flap effectiveness (average slope), the trend is the same as seen at zero angle of attack (see Figure 4.15).

For $\alpha = 18.0°$, $k = 0.1$ and $\hat{\beta} = 10.0°$ the models and experiments vary in integral value. For both the flap deflection amplitudes, the engineering model predicts a different behavior, since it shows a negative trend in integral value, opposed to the other models and experimental values. In Figure 4.18 the relative normalized unsteady lift coefficient is shown for this specific case, in which the differences become apparent. The engineering model both predicts a different average slope and a opposite loop direction, resulting in the negative integral value. Bergami et al. [7] found similar behavior w.r.t. the direction of the loop at stalled conditions, which was depended on the input data. However, their CFD results did not show a reversed direction. The two key differences between the results at $\alpha = 18.0°$ and at the lower angles of attack is: 1) the negative steady-state lift slope w.r.t. the angle of attack for $\alpha = 18.0°$, and 2) the separated suction side causing the flap deflection only to effect the pressure side.

The experiments and the Spalart-Allmars results correspond well, especially considering the separated aerodynamics. Both in average direction as loop thickness, a good correspondence is found. However, the kω-SST model shows a more slender loop and a pronounced offset from the zero relative lift. Since the steady-state RANS equations are not able to capture the vortex shedding causing inherently unsteady results at $\alpha = 18.0°$, the resulting steady-state lift is different from the average lift when computed by the URANS equations. This is most clearly seen at low reduced frequency, where the same offset is present (see Figure D.3). However, both URANS models are able to predict the reverse direction of the loops (from counter-clockwise to clockwise), but the engineering model fails to predict this, while it uses the steady-state experimental data.

From the results it can be concluded that the models predict similar behavior at the two lower angles of attack, but differ more on the highest angle of attack. Compared to the experimental values only a small over-prediction of the loops is found, although the direction/slope of the loops deviates from the forced-transition experimental results. In the experimental lift and pressure distributions, irregularities are found which required a more detailed study on the experimental data. Especially in the flap region unexplained oscillations are found. At the highest angle of attack, the engineering model fails to predict the reverse loop direction, while it uses the steady-state experimental data as input.

Overall, the URANS model with the Spalart-Allmaras turbulence model performs best in predicting the unsteady lift effects due to the flap motion, even-though the deviations from the experimental values are still significant.

DRAG COEFFICIENT

The integral value of the drag, presented in Figure 4.21, shows a different behavior than the lift integral values. At zero angle of attack, the models predict a positive integral value, while the experiments show a negative value (forced-transition) or a small positive value (free-transition). In addition, the magnitude of the integral is significantly larger for the models. In Figure 4.22 the relative normalized unsteady drag coefficient for $\alpha = 0.0°$, $k = 0.1$ and $\hat{\beta} = 10.0°$ is shown for the 2 models together with the experimental data. Even though the direction of the loops is the same, the crossing is located near zero flap angle for the experiments, while the models all predict a negative flap angle when the lines cross. This causes the integral value to be of different sign. Purely considering the area within the loops, one can conclude that all models over-predict the unsteady variation in the drag, especially the two URANS results. In addition, experimental results show an offset from the horizontal axis, which is caused by the difference between the average unsteady drag force and the steady-state drag force. Where this differences originates from is unclear and should be further investigated within the experimental study.

Increasing the angle of attack, the difference between the integral values of numerical models and the experimental values decreases. However, the over-prediction at an angle of attack of 8.0° is still high, with the two URANS results predicting more than two times higher integral values compared with the forced-transition results. However, compared with the free-transition experimental data, the over-prediction is in the order of 50% for the URANS results and 20% for the engineering model. In Figure 4.23 these over-predictions are related to the thicker loops of the numerical results. As for the unsteady normalized relative lift, the average slope of the loops differ between the experiments (neutral) and models (positive). In Figure 4.25 the drag distribution is shown for the Spalart-Allmaras model and the two experimental results. Again, the numerical predicts a large increase in unsteady drag near the leading edge, although the unsteady drag near the flap is more pronounced when compared with the lift distributions. However, the free-transition experimental data only shows variation near the flap and an almost flat unsteady drag at the remaining part of the airfoil. Even though the forced-transition models shows more variation along the chord, the trend cannot be related to the numerical results. As concluded within the analysis of the lift coefficient, the experimental data should be studied further for any further validation.

At the highest angle of attack the engineering model predicts a negative value of the integral value, opposite to the positive values found for the URANS and experiments. The engineering model predicts a counter-clockwise loop direction and a negative average slope, as can be seen in Figure 4.24. The experiments and URANS results do not predict the change in loop direction. A similar result was found in the lift results, where the engineering model failed to predict the change in loop direction. The main difference

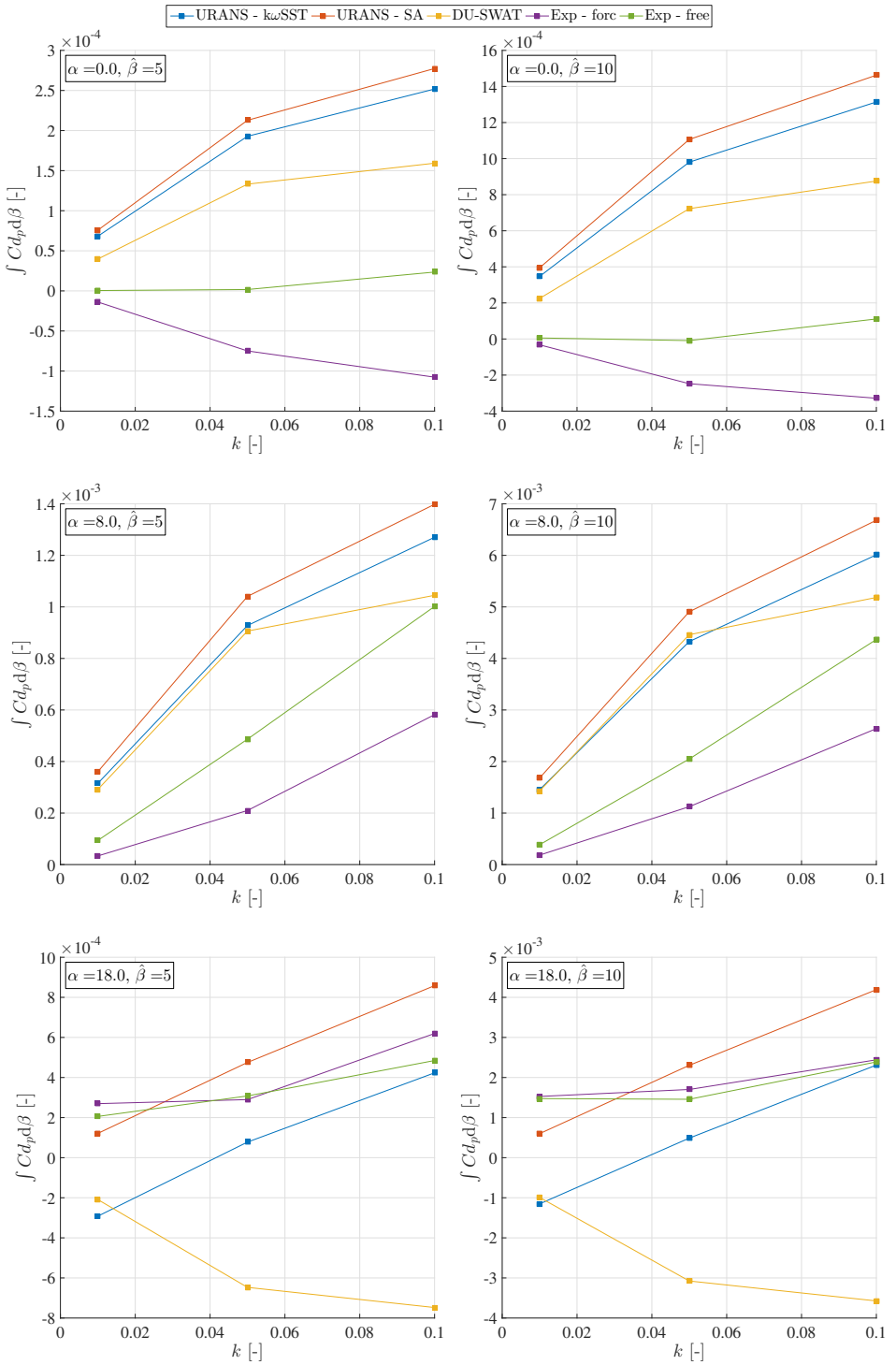

Figure 4.21: Integrated unsteady Cd_p at three angles of attack and two flap deflection amplitudes. $\int Cd_p \mathrm{d}\beta$ is obtained by integrating the unsteady drag coefficient for a single period over β.

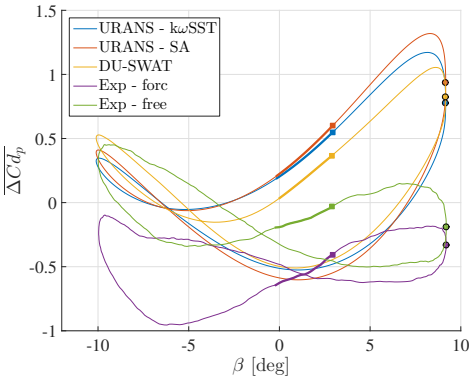

Figure 4.22: Unsteady normalized ΔCd_p for $\alpha = 0.0°$, $k = 0.1$ and $\hat{\beta} = 10.0°$. $\overline{\Delta Cd_p}$ is obtained by deducting the corresponding steady-state values at the corresponding angle of attack and flap angle for each model separately and normalizing this with the band of the experimental steady-state results.

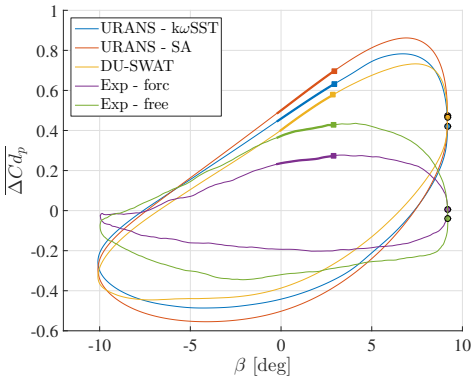

Figure 4.23: Unsteady normalized ΔCd_p for $\alpha = 8.0°$, $k = 0.1$ and $\hat{\beta} = 10.0°$. $\overline{\Delta Cd_p}$ is obtained by deducting the corresponding steady-state values at the corresponding angle of attack and flap angle for each model separately and normalizing this with the band of the experimental steady-state results. Dots indicate with black lining indicate the positions for which the drag distributions are shown in Figure 4.25.

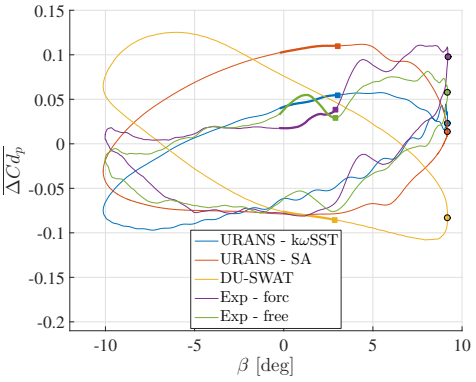

Figure 4.24: Unsteady normalized ΔCd_p for $\alpha = 18.0°$, $k = 0.1$ and $\hat{\beta} = 10.0°$. $\overline{\Delta Cd_p}$ is obtained by deducting the corresponding steady-state values at the corresponding angle of attack and flap angle for each model separately and normalizing this with the band of the experimental steady-state results.

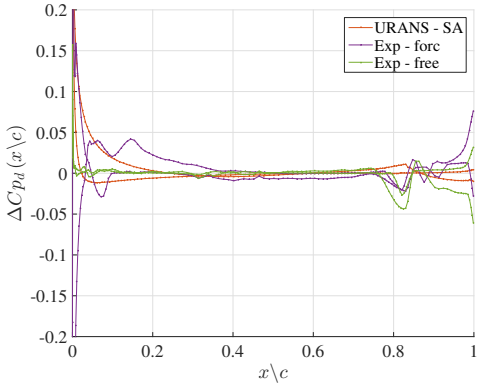

Figure 4.25: Delta drag distribution $\Delta Cp_d(x\backslash c)$ for $\alpha = 8.0°$, $k = 0.1$ and $\hat{\beta} = 10.0°$ at maximum flap deflection (indicated by black lined dot in Figure 4.23). $\Delta Cd_p(x\backslash c)$ is obtained by deducting the corresponding steady-state drag distribution from the unsteady drag distribution. The drag distributions themselves are obtained by decomposing the pressure distribution in lift and drag contribution.

between the kω-SST result and the Spalart-Allmaras result is the irregularity within kω-SST results, caused by the high frequency vortex shedding predicted by this model. This is best illustrated by Figure D.6, where the drag variations at lower reduced frequency show a more pronounced high frequency oscillation caused by the vortex shedding. At

these angles of attack vortex shedding is expected, however, URANS is known to over-predict drag caused by vortex shedding due to the incorrect prediction of large coherent vortical structures. However, the loop behavior is similar, corresponding to the small difference between the integral values. Focussing on the Spalart-Allmaras result, the thickness of the loop is largely over-predicted (almost 80%), but the direction of the loop correspond reasonably well.

It can be concluded that all models have significant difficulties predicting the unsteady drag coefficient due to the flap. Generally, the unsteady variation is over-predicted, mostly by the URANS models. This can be best seen in the thickness of the loop. Since the unsteady variation within the drag can be up to 80% off the steady-state variation (see Spalart-Allmaras in Figure 4.23) this prediction should be improved by further investigating where the differences come from. The engineering model shows a smaller over-prediction, but is also using the experimental data as input. In addition, the engineering model predicts a change in loop directions at $\alpha = 18.0°$, while this is not seen in both the experimental data and the URANS results.

MOMENT COEFFICIENT

Finally, the unsteady moment coefficient is considered. In Figure 4.26, the integral value of the unsteady moment coefficient can be found. Differences at zero angle of attack are relatively large, but this is mainly caused by the small variation (both unsteady and steady-state) in moment coefficient at this angle.

More interesting is the more pronounced opposite sign of the integral value at an angle of attack of 8.0°. This effect increases when increasing the reduced frequency. Therefore, Figure 4.27 shows the relative normalized unsteady moment coefficient for $\alpha = 8.0°$, $k = 0.1$ and $\hat{\beta} = 10.0°$, where the differences are the most pronounced. First, the loops of the models are in clockwise direction, while the experimental results show a anti-clockwise loop. Secondly, the experimental loops have the widest part at negative flap angles and the models at the positive flap angles. Finally, the engineering model only predicts a small variation compared to the other values. To determine where the opposite loop direction originates from the moment distribution at zero angle of attack is shown in Figure 4.29. The Spalart-Allmaras results are in good agreement with the free-transition experimental results near the leading edge. However, the largest contribution in the experimental data is found near the flap hinge, having a large arm to the quarter chord point. Due to the pressure distributions oscillations, observed before in the lift analysis, the experimental moment results are questionable. Any further analysis should be focussed on the assessment of the experimental results.

The sign of the experimental values changes when considering the highest angle of attack. In Figure 4.28 the relative normalized unsteady moment coefficient is shown for $\alpha = 18.0°$, $k = 0.05$ and $\hat{\beta} = 10.0°$, clearly showing the now counter-clockwise loop of both experimental and Spalart-Allmaras results. The kω-SST results show a different behavior, with a figure-of-eight loop and a largely counter-clockwise direction. As seen in the drag and lift results at higher angles of attack, the kω-SST turbulence models predicts vortex shedding, causing the large differences in results w.r.t. the Spalart-Allmaras

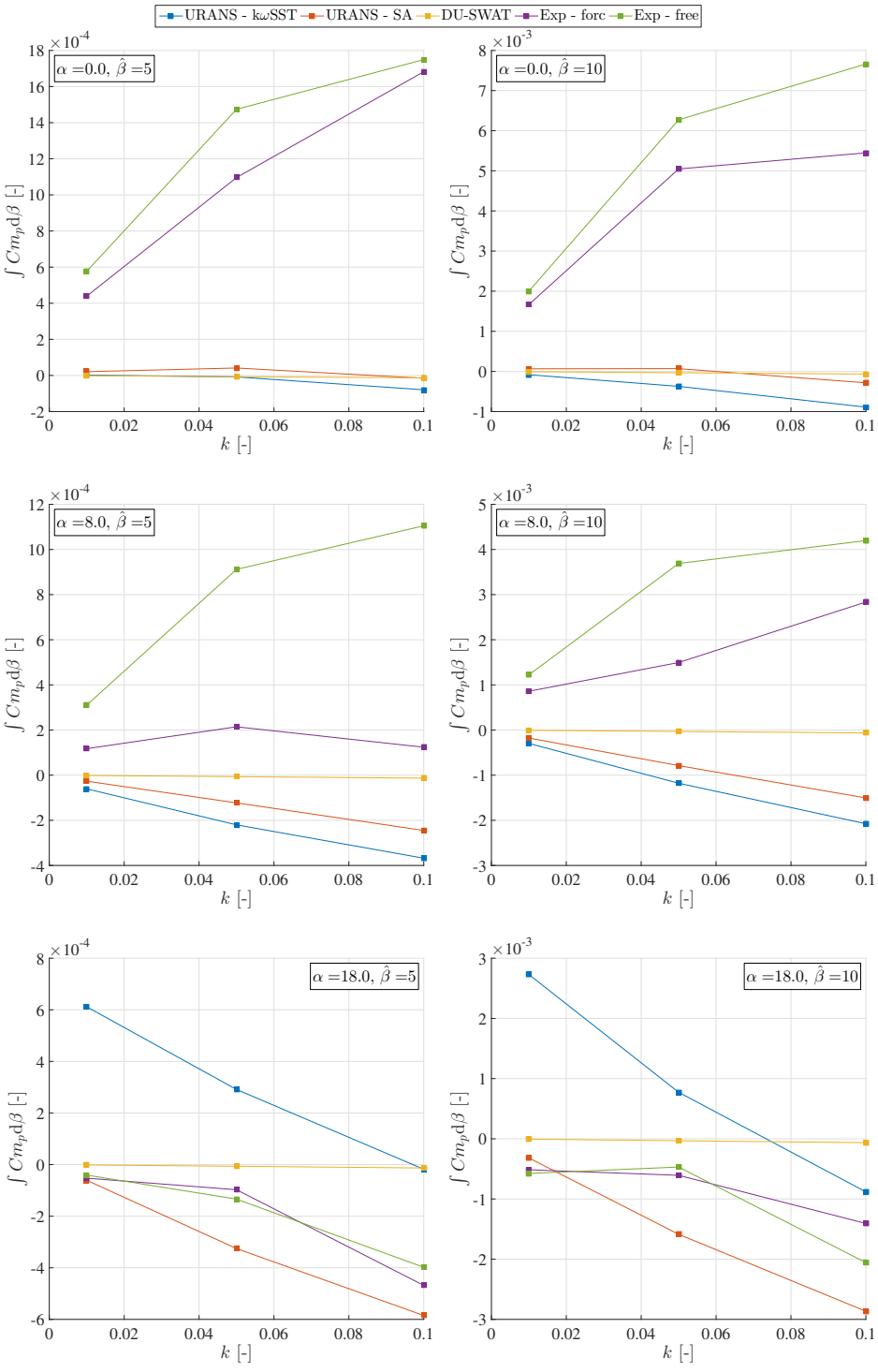

Figure 4.26: Integrated unsteady Cm_p at three angles of attack and two flap deflection amplitudes. $\int Cm_p \mathrm{d}\beta$ is obtained by integrating the unsteady moment coefficient for a single period over β.

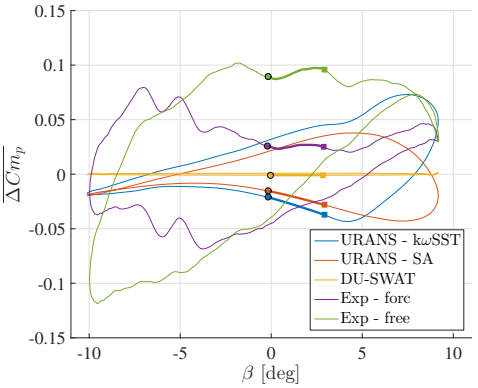

Figure 4.27: Unsteady normalized ΔCm_p for $\alpha = 8.0°$, $k = 0.1$ and $\hat{\beta} = 10.0°$. $\overline{\Delta Cm_p}$ is obtained by deducting the corresponding steady-state values at the corresponding angle of attack and flap angle for each model separately and normalizing this with the band of the experimental steady-state results. Dots indicate with black lining indicate the positions for which the moment distributions are shown in Figure 4.29

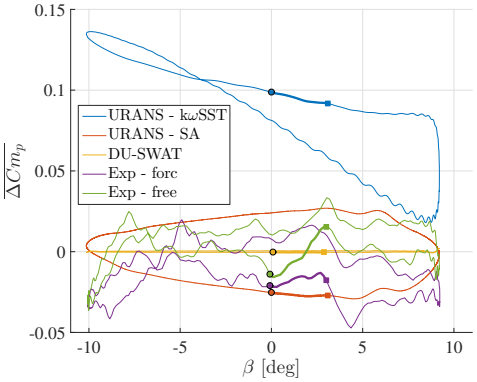

Figure 4.28: Unsteady normalized ΔCm_p for $\alpha = 18.0°$, $k = 0.05$ and $\hat{\beta} = 10.0°$. $\overline{\Delta Cm_p}$ is obtained by deducting the corresponding steady-state values at the corresponding angle of attack and flap angle for each model separately and normalizing this with the band of the experimental steady-state results.

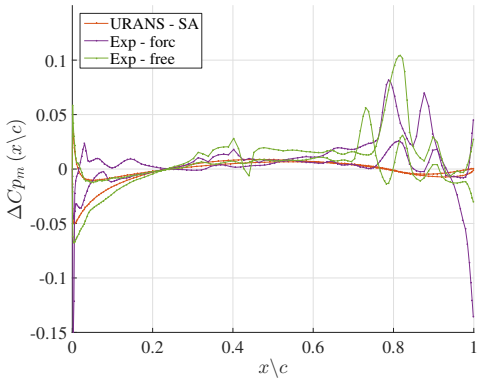

Figure 4.29: Delta moment distribution $\Delta Cp_m\,(x\backslash c)$ for $\alpha = 8.0°$, $k = 0.1$ and $\hat{\beta} = 10.0°$ at zero flap deflection (indicated by black lined dot in Figure 4.27). $\Delta Cm_p\,(x\backslash c)$ is obtained by deducting the corresponding steady-state moment distribution from the unsteady moment distribution. The moment distributions themselves are obtained by decomposing the pressure distribution in lift, drag and moment contribution.

results.

Summarizing, the Spalart-Allmaras model is able to predict most of the behavior of the moment coefficient correctly. The engineering model does predict a significantly

smaller variation. However, it should be noted that the unsteady variation is of the order of 10% compared to the steady-state variation, questioning the significance of the differences found, especially taking into account the experimental unsteady moment results near flap.

4.4. CONCLUSIONS

Results from the URANS model (with both the Spalart-Allmars and kω-SST turbulence model) and the engineering model are compared with the steady and unsteady experimental data to validate these models for modeling airfoils with active trailing edge flaps. Three parameters have been varied: angle of attack, reduced frequency and flap deflection amplitude.

From the steady-state results, it becomes immediately apparent that the URANS model predicts the linear region well, but over predicts the lift at the non-linear region. In addition, the lift slope is also over-predicted compared with the forced-transition results by both the turbulence models at high flap deflections. However, compared with the free-transition results, the linear region is captured well. Both the turbulence models under-predict the drag coefficient, both at low and high angles of attack. However, the increase in drag due to the flap deflection is relatively well captured. Finally, for the moment coefficient the attached flow regime a well predicted by both turbulence models, while for higher angles of attack the deviation increases. The kω-SST is better in predicting the steep change in moment coefficient around an angle of attack of $+/-10^{\circ}$. The Spalart-Allmaras model over- or under-predicts the influence of the flap angle for negative and positive flap angles, respectively.

The unsteady influence is analyzed by considering $\overline{\Delta Cl_p}$, $\overline{\Delta Cd_p}$ and $\overline{\Delta Cm_p}$, representing the normalized relative unsteady pressure lift, drag and moment coefficients. These values are relative to the steady-state results and normalized with the band from the corresponding experimental steady-state values. Both models predict similar behavior as the experiments at the two lower angles of attack, but differ more on the highest angle of attack. Differences occur both in thickness of the loops as well as the average slope and loop direction. From a more detailed analysis it becomes clear that irregularities are present in the experimental pressure distributions, affecting the lift distribution and thus the unsteady lift variation. Especially near the flap hinge, oscillations within the unsteady pressure distributions are found, where these are not present in the steady-state results. The negative average unsteady lift slope found for the numerical results is related to the delay in lift (and thus pressure) distribution, which is illustrated by the large offset from the steady-state values near the leading edge, while having almost no offset at near the flap. Overall, the URANS model with the Spalart-Allmaras turbulence model performs best in predicting the unsteady lift effects due to the flap motion.

All models have significant difficulties when predicting the unsteady drag coefficient due to the flap. Generally, the unsteady variation is over-predicted, mostly by the URANS models and especially by the Spalart-Allmaras model. Similar as seen in the unsteady lift coefficient, the engineering model predicts the reverse direction of the unsteady loop at the highest angle of attack.

The Spalart-Allmaras model is able to predict most of the behavior of the moment coefficient correctly. The engineering model does predict a significantly smaller variation. However, the unsteady variation is of the order of 10% compared to the steady-state variation, questioning the significance of the differences found. In addition, the experimental moment coefficients are largely affected by the irregularities near the flap hinge, due to the large arm to the quarter chord point.

From the validation it becomes clear that both the engineering model and the URANS model perform well in terms of unsteady lift prediction of a TEF, especially at attached flow conditions. Most important for the engineering model is the proper input data, since this is where most of differences can arise from. Even though the experimental data has been used, the engineering model over-predicts the drag and predicts a minor variation in moment coefficient. In addition, the focus for the URANS model should be on improving the stead-state predictions and unsteady drag predictions. Even though kω-SST predicts the steady lift, drag and moment the best, Spalart-Allmaras captures the unsteady behavior better. Finally, the experimental data should be further studied, especially the differences between the steady-state and unsteady pressure distributions near the flap hinge. The influence of the zig-zag strip on the pressure distribution should be initially investigated for the steady-state conditions at large positive flap deflections. Whenever the experimental data has been improved, the approach of coupling the integral quantities to the unsteady normalized relative polars should provide more insight in the predictions capabilities of the different models.

REFERENCES

[1] T. Barlas and G. van Kuik, *Review of state of the art in smart rotor control research for wind turbines,* Progress in Aerospace Sciences **46**, 1 (2010).

[2] P. B. Andersen, *Advanced load alleviation for wind turbines using adaptive trailing edge flaps : sensoring and control*, Ph.D. thesis, Riso DTU (2010).

[3] M. A. Lackner and G. A. M. van Kuik, *A comparison of smart rotor control approaches using trailing edge fl aps and individual pitch control,* Wind Energy **13**, 117 (2010).

[4] J. Heinz, N. N. Sørensen, and F. Zahle, *Investigation of the load reduction potential of two trailing edge flap controls using CFD,* Wind Energy **14**, 449 (2011).

[5] E. Jost, A. Fischer, T. Lutz, and E. Krämer, *CFD studies of a 10 MW wind turbine equipped with active trailing edge flaps,* in *Proceedings of the 10th PhD Seminar on Wind Energy in Europe*, October (Orleans, France, 2014) pp. 119–122.

[6] T. Wolff, J. R. Seume, and L. U. Hannover, *Numerical investigation of an airfoil with morphing trailing edge,* in *Proceedings of the 10th PhD Seminar on Wind Energy in Europe* (Orleans, France, 2014) pp. 114–118.

[7] L. Bergami, V. A. Riziotis, and M. Gaunaa, *Aerodynamic response of an airfoil section undergoing pitch motion and trailing edge flap deflection : a comparison of simulation methods,* Wind Energy (2014).

[8] C. Bak, M. Gaunaa, P. B. Andersen, T. Buhl, P. Hansen, and K. Clemmensen, *Wind tunnel test on airfoil Risø-B1-18 with an Active Trailing Edge Flap,* Wind Energy **13**, 207 (2010).

[9] P. Baek, J.-G. Jérémiaz, P. Kramer, and M. Gaunaa, *Experimental comparison of a miniflap to a Trailing Edge Flap on a wind turbine airfoil,* EWEA conference , 1 (2011).

[10] T. Lutz, A. Wolf, W. Wurz, and J.-G. Jeremiasz, *Design and verification of an airfoil with trailing-edge flap and unsteady wind-tunnel tests: UPWIND WP1B3,* Tech. Rep. (2011).

[11] C. J. Simão Ferreira, A. G. Salcedo, and T. Gillebaart, *Unsteady measurements of the DU95W180 airfoil with oscillating flap,* Wind Energy **to be subm** (2016).

[12] Avatar, *AVATAR (AdVanced Aerodynamic Tools of lArge Rotors),* (2015).

[13] J. Leishman and T. Beddoes, *A Generalised Model for Airfoil Unsteady Aerodynamic Behaviour and Dynamic Stall Using the Indicial Method - AHS Online Store,* in *Proceedings of the 42nd Annual Forum of the American Helicopter Society"* (Washington D.C., USA, 1986).

[14] L. O. Bernhammer, *Smart Wind Turbine: Analysis and Autonomous Flap,* Ph.D. thesis, Delft University of Technology (2015).

[15] L. Bergami and M. Gaunaa, *ATEFlap Aerodynamic Model, a dynamic stall model including the effects of trailing edge flap deflection,* Ris{ø}-R-1792(EN) (2012).

[16] M. Gaunaa, *Unsteady two-dimensional potential-flow model for thin variable geometry airfoils,* Wind Energy **13**, 167 (2010).

[17] H. Jasak, *Department of Mechanical Engineering Imperial College of Science, Technology and Medicine,* Ph.D. thesis, Imperial College of Science, Technology and Medicine (1996).

[18] F. R. Menter, *Two-equation eddy-viscosity turbulence models for engineering applications,* AIAA Journal **32**, 1598 (1994).

[19] F. R. Menter, M. Kuntz, and R. Langtry, *Ten Years of Industrial Experience with the SST Turbulence Model,* Turbulence, Heat and Mass Transfer **4**, 625 (2003).

5

ACTIVE FLAP CONTROL ON AN AEROELASTIC WIND TURBINE AIRFOIL IN GUST CONDITIONS

Within this Chapter the Fluid-Structure-Control Interaction model is completed by coupling the URANS model to a 3 Degrees of Freedom structural model, flap controller and varying inflow conditions. Confidence in aero-servo-elastic tools used to assess and predict the behavior of rotors using trailing edge devices like flaps is still limited. Ideally, experimental data is used as validation, however, experimental validation data for a aero-servo-elastic problems is not available. A comparison of different models could provide confidence or at least insight in these models and identify possible areas for improvements. In this chapter an unsteady aerodynamic model (Beddoes-Leishman type) and the CFD model (URANS) are used to analyze the aero-servo-elastic (i.e. FSCI) response of a 2D three degree of freedom rigid body wind turbine airfoil with a deforming trailing edge flap encountering deterministic gusts. Both uncontrolled and controlled simulations are used to assess the differences between the two models for 2D aero-servo-elastic simulations. A parameter study is performed by varying the inflow conditions and gust conditions while assessing the differences between the results of the two models.

Parts of this chapter have been published in Gillebaart et al., Active flap control on an aeroelastic wind turbine airfoil in gust conditions using both a CFD and an engineering model, Journal of Physics: Conference Series (2014)

5.1. INTRODUCTION

Smart rotor models consist out of four components: aerodynamic modeling, structural modeling, inflow variations and controller. In the previous Chapter (Chapter 4) the aerodynamics have been validated for both the engineering model as well as for the URANS model. The final step for a complete 2D FSCI model is to include a structural model, inflow variations and a controller. With such a complete model three things can be achieved: quantify the accuracy of engineering models for aero-servo-elastic problems compared to URANS models, detailed investigation of the time-varying force/pressure distribution and increase the confidence of such a high fidelity model as a step towards 3D smart rotor simulations.

A way to quantify the accuracy of the URANS and engineering models, besides validating with experiments, is by comparing the results of the two models. First efforts in using CFD for 2D controlled flap motion have been made by Heinz et al. [1]. These efforts will be followed up in this chapter by giving a thorough quantification of the aero-servo-elastic response of a 3 degree of freedom structural model combined with two different aerodynamic models, namely the unsteady aerodynamic model [2] and the URANS model.

As a first step, 2D simulations are used to assess the differences between the aero-servo-elastic response of the aerodynamic models to deterministic gusts. To assess this the DU91-W2-250 airfoil is used [3], where the last 10% is used as a deforming flap, which is connected to a PD controller. To limit the complexity of the controller in this research, noise, delay maximum rotational speed and maximum hinge force (known influences in the controller system) are currently not modeled. To have controlled inflow conditions, limiting the scope of this study, but ensuring interpretable results, two deterministic gust types are considered: 1-cosine and Mexican hat gust. Both uncontrolled and controlled responses are given, after which the differences between the model responses are analyzed.

Finally, to assess the limits of applicability of the engineering model a small parameter study is performed using the Mexican hat gust and varying two parameters: angle of attack to vary the type of flow from linear to separated flow and the gust frequency to vary the deviation from quasi-steady behavior. Trends in the differences between the results from the two models will be analyzed and presented as final part of this chapter.

First the methods applied in the URANS model to introduce gusts is discussed, after which the structural model, flap deflection, controller and gust shapes are presented. After this, the results of both the detailed comparison of a typical aero-servo-elastic response and the parameter study are discussed.

5.2. MODELING APPROACH

Two aerodynamic models are used as input to an aero-servo-elastic system: 1) the unsteady aerodynamic model and 2) the high fidelity Unsteady Reynolds Averaged Navier-Stokes (URANS) model. Both models are coupled with a three degree of freedom rigid body structural model (horizontal displacement, vertical displacement and pitching) in-

cluding a deformable trailing edge flap. To limit the structural response due to gusts a PD controller is used to control the flap. Both aerodynamic models, the structural model, the controller and the flow conditions are discussed in this section.

5.2.1. UNSTEADY AERODYNAMIC MODEL

For the unsteady aerodynamic model, the same approach is taken as presented in Section 4.2.1, except for the steady-state input data. Steady-state CFD simulations are used to generate steady-state input data for the engineering model to ensure a fair comparison between the models on their unsteady predictions. For consistency, the same mesh and turbulence model as for the unsteady CFD simulations are used to generate the engineering input data. Results sweeping over angles of attack from -20 to 20 degrees and flap angles from -10 to 10 degrees are used to obtain the steady input data for the unsteady aerodynamic model. To include gusts in the unsteady aerodynamic model a temporal variation in the angle of attack and inflow velocity magnitude are used.

5.2.2. URANS MODEL

For the URANS model, the same approach is taken as presented in Section 4.2.2. Again the $k\omega - SST$ turbulence model is used [4]. However, since a higher Reynolds number is used, a new mesh is generated to ensure the y^+ remains approximately 1.

MESH STUDY

In order to assess the spatial discretization error a mesh study is performed. Three meshes have been used for this mesh-study, as shown in Table 5.1

Table 5.1: Mesh characteristics of the four meshes used for the mesh study. Distances/lengths are given in chord lengths.

	mesh 1	mesh 2	mesh 3
$l_{farfield}$	93	92	93
$n_{points}^{farfield}$	120	240	480
$n_{points}^{airfoil}$	145	290	579
n_{cells}	17400	69600	277920

For the mesh study, a steady-state case is used at 9.5° angle of attack with a zero flap deflection. In Figure 5.1 the resulting vertical and horizontal force coefficients are shown for the three meshes. Based on these results, mesh 2 is chosen as mesh, having a difference below 0.5% in both force coefficients compared to the extrapolated value. A 2^{nd} order time integration scheme is used (BDF2), while the spatial discretization is second order central, except for the convective terms which are discretized by 2^{nd} order linear upwind. Each time step the PISO loop is iterated until a residual of $5 \cdot 10^{-4}$ is obtained.

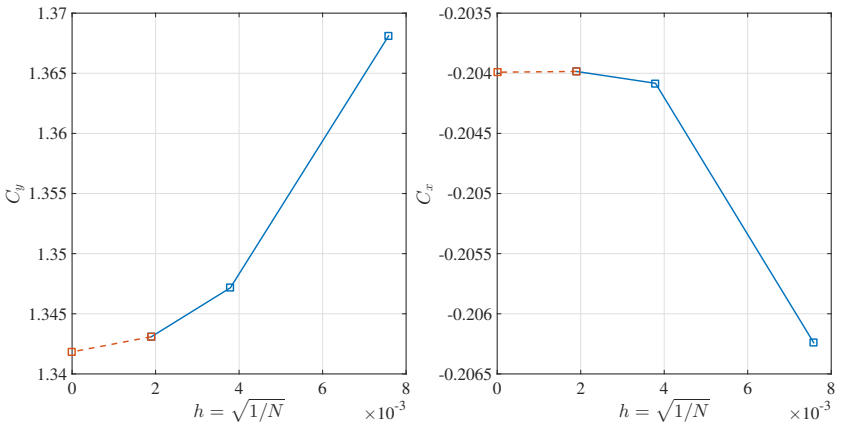

Figure 5.1: Vertical and horizontal force coefficient for 3 different meshes. Horizontal axis is average mesh spacing at the airfoil surface for the different meshes used, as summarized in Table 5.1. Estimated final force coefficients are obtained by Richardson extrapolation.

5

GUSTS

Gusts are incorporated in both the URANS model and the unsteady aerodynamic model. For the unsteady aerodynamic model gusts are introduced by a change in angle of attack and velocity. For the URANS model the mesh velocity technique is used [5]. The method is based on using the mesh velocities to incorporate the gusts as shown in the Equations 5.1 and 5.2:

$$V = \mathbf{u} - \mathbf{u}_m + \mathbf{u}_g = \left(u - u_m + u_g\right)\mathbf{i} + \left(v - v_m + v_g\right)\mathbf{j} + \left(w - w_m + w_g\right)\mathbf{k} \tag{5.1}$$

$$\tilde{\mathbf{u}}_m = \tilde{u}_m\mathbf{i} + \tilde{v}_m\mathbf{j} + \tilde{w}_m\mathbf{k} = \left(u_m - u_g\right)\mathbf{i} + \left(v_m - v_g\right)\mathbf{j} + \left(w_m - w_g\right)\mathbf{k}. \tag{5.2}$$

Here u, v and w are the velocity components, u_m, v_m and w_m are the geometric grid velocities caused by mesh deformations and u_g, v_g and w_g are imposed gust velocities. By modifying the grid velocities the gust velocities are incorporated in the simulation. The modified grid velocities are represented by $\tilde{u}_m$, $\tilde{v}_m$ and $\tilde{w}_m$. For global gusts, with equal velocity in whole computational domain, this is the same as moving the mesh accordingly. However, this method also gives the possibility to simulate traveling (local) gusts, which vary spatially across the domain.

5.2.3. STRUCTURAL MODEL

Both the URANS model and the unsteady aerodynamic model are coupled to a 3 degrees of freedom structural model without structural damping. This model is governed by the following equations:

$$m\ddot{x}_s + k_x x_s = F_x + ml\dot{\theta}_s^2 \cos\left(\theta_s + \theta_g\right) + ml\ddot{\theta}_s \sin\left(\theta_s + \theta_g\right), \tag{5.3}$$

$$m\ddot{y}_s + k_y y_s = F_y + ml\dot{\theta}_s^2 \sin\left(\theta_s + \theta_g\right) - ml\ddot{\theta}_s \cos\left(\theta_s + \theta_g\right), \tag{5.4}$$

$$\left(I_{CG} + ml^2\right)\ddot{\theta}_s + k_\theta \theta_s = F_\theta + ml\ddot{x}_s \sin\left(\theta_s + \theta_g\right) - ml\ddot{y}_s \cos\left(\theta_s + \theta_g\right). \tag{5.5}$$

Where, m is the mass per unit depth, k is the spring stiffness in the respective degree of freedom, F is the aerodynamic force in the respective degree of freedom, F_θ is the counter clockwise positive aerodynamic moment around the rotational center, l is the distance between the centre of gravity and the rotational centre positive in the direction from leading edge to trailing edge, θ_s is pitch angle deformation, θ_g the geometric installed pitch angle and I_{CG} is the moment of inertia around the centre of gravity. The equations are solved in time using an explicit four stage Runge-Kutta time integration. Figure 5.2 is an illustration of the aerodynamic forces and relevant geometric variables. The model is equal to the one used in [1]. During this study the parameters stated in

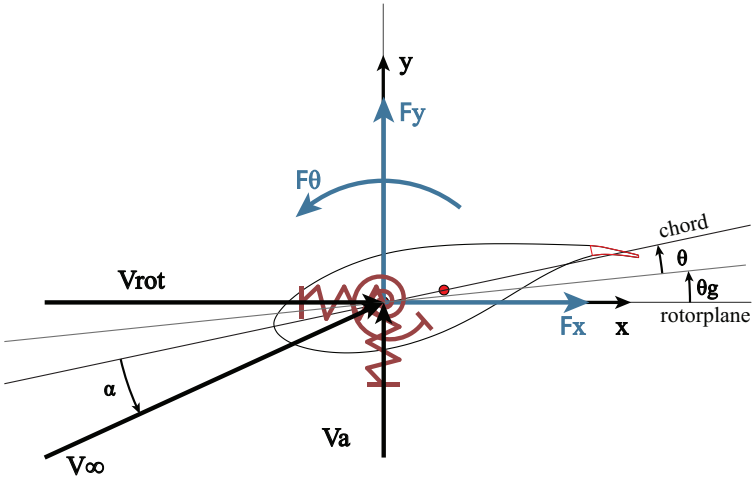

Figure 5.2: Illustration of structural model including a flap together with forces, inflow directions and angle definitions.

Table 5.2 are used in the structural model. The values stated result in a realistic 2D representation of an airfoil section in a wind turbine blade [1]. Next, the fluid and structural model need to be coupled. Coupling between the structure and aerodynamics is done in a strong way using sub iterations combined with Aitken's under relaxation [6], such that the partitioned approach has no influence on the results.

Table 5.2: Structural properties for three degree of freedom rigid body model

c	1 m	k_x	6316 N/m
RC from LE	0.3 m	k_y	1579 N/m
CG (from LE)	0.35 m	k_θ	8290 N m/rad
m (per unit depth)	40 kg/m	θ_g	5^o
I_{CG}	2 kg m^2		

5.2.4. CONTROLLER AND FLAP

Both models use a PD controller on the body velocity to describe the rotational velocity of the flap. As input the vertical velocity and the acceleration of the body (i.e. wing) are used, with the goal of minimizing the displacement in vertical direction (y). Consequently, the rotational velocity is integrated using a second order backward differencing scheme in the URANS model and a Runge-Kutta scheme in the unsteady aerodynamics model. The controller is described by Equation 5.6. Noise and delay are not considered in this controller, which will be present in the actual system. However, to compare the two models these influences are considered to be less important in the current scope of the study, even though they are of importance for designing appropriate controllers.

$$\frac{d\beta}{dt} = K_v \frac{dy}{dt} + K_a \frac{d^2y}{dt^2} \tag{5.6}$$

The gains are tuned to: $K_v = -100$ and $K_a = -20$ by hand using the unsteady aerodynamic model, which does not necessarily ensure an optimal controller.

From the integration of the rotational velocity the flap angle is obtained, which is used to deform the flap accordingly. To prevent sharp edges on the airfoil surface a smooth deformation is chosen described in Equations 5.7:

$$y_{flap} = y^0_{flap} - \left(x_{flap} - x_{hinge}\right)^2 \beta \frac{1}{l_{flap}}. \tag{5.7}$$

Here y_{flap} and x_{flap} are the coordinates describing the flap shape with there origin at the leading edge. Additionally, y^0_{flap} is the original y position of the flap, x_{hinge} is the hinge location of the flap, β is the flap angle in radians, and l_{flap} is the flap length. The flap length is chosen to be 10% of the chord. No deformation in x direction is assumed, which is valid for small angles. The maximum allowed angle in the controller is 7 degrees (0.12 rad), which makes this assumption reasonable for the intended purpose. In Figure 5.3 the flap deformation is shown.

5.2.5. FLOW CONDITIONS AND GUSTS

Typical flow conditions for a wing turbine section close to the tip are taken from [1]. A rotational velocity of 60 m/s is chosen, while as unperturbed axial velocity a speed of 10 m/s is used. With a chord length of 1 m and the dynamic viscosity equal to $1.4531 \cdot 10^{-5}$ m^2/s a Reynolds number of 4.19 million is obtained. For the turbulence model a turbulence intensity of 0.01% is used and a fully turbulent flow is assumed. The DU91-W2-250 airfoil is used as typical wind turbine airfoil [3], where the last 10% of the wing are used as flap.

For a clear response a specific deterministic shape of the gust is used. Interaction between the gust and the structure, controller and aerodynamics is limited to a one-way coupling: the gust does interact with the aerodynamics, structure and controller, but the gust is not influenced by either of those components. Two types of gusts are considered: a 1-cos gust and a Mexican hat gust. Equations 5.8 and 5.9 describe the 1-cos and Mexican hat gusts, respectively.

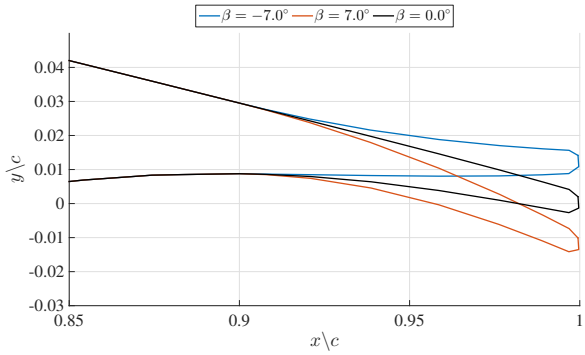

Figure 5.3: Deformation of the trailing edge flap for three flap angles: $\beta = -7.0°$, $\beta = 0.0°$ and $\beta = 7.0°$

$$V^g_{cos}(t) = \frac{A_g}{2}\left(1 - \cos\left(2\pi f_g \xi\right)\right) \text{ if } 0 < \xi < 1/f_g \tag{5.8}$$

$$V^g_{mex}(t) = \frac{A_g}{2}\left(1 - \cos\left(2\pi f_g \xi\right)\right)\sin\left(3\pi f_g \xi\right) \text{ if } 0 < \xi < 1/f_g \tag{5.9}$$

Here V_g is the gust velocity in time and space, A_g the gust amplitude, f_g the gust frequency and $\xi = t - t^g_0$, where t^g_0 is the starting time of the gust.

5.3. RESULTS

For both models the airfoil response to two types of gusts is simulated and the results are compared. First the response of the two models is analyzed in detail for a case within the attached flow regime and a small gust amplitude. Here differences are expected to be small. Consecutively, two parameters of the inflow conditions are varied: angle of attack and gust frequency. From those results trends are derived in the observed differences between the two models. The computational times (CPU time) for one of these simulations are in the order of 200 CPU seconds for the unsteady aerodynamic model and 50 CPU hours for the URANS model.

All simulations are started from a steady-state solution. Due to the coupling between structure and flow model, the steady-state results involve an aeroelastic computation to determine the deflections and forces for which the airfoil is in balance for the specified angle of attack. The resulting steady-state for both models is shown in Table 5.3. Here C_{ms} is the moment coefficient around the rotational center, counterclockwise positive. Differences (up to 8%) are already present in the steady-state solution. Two causes are identified: 1) steady-state CFD input data is interpolated within the engineering model to obtain a lift, drag and moment coefficient at the specific angle of attack and flap angle. This interpolation only ensures the forces to be equal to the CFD data at the angle of attack and flap angle where the CFD data is available, leading to (small) interpolation errors at different angles of attack and flap angles. 2) The URANS simulations involve

Table 5.3: Steady state forces and deflections for both the URANS and UA model.

	URANS	UA	difference
x [m]	-0.036	-0.038	5.6%
C_x [-]	-0.101	-0.105	4.0%
y [m]	0.992	1.026	3.4%
C_y [-]	0.691	0.715	3.5%
θ [deg]	1.124	1.214	0%
C_{ms} [-]	0.072	0.078	8.3%

mesh deformation, resulting in small changes in the computed forces. These two effects combined result in a slightly larger moment coefficient, resulting in a larger pitch angle for the engineering model. As a consequence the engineering model has a larger effective angle of attack, leading to a larger vertical force and thus displacement. For the unsteady results the initial steady-state values are deducted, to eliminate the small offset in results due to the different steady-state values.

A cosine and Mexican hat gust with a frequency of 1.2 Hz and an amplitude of 1 m/s are chosen, which act in the axial direction, depicted as the y-axis in Figure 5.2. To determine the effectiveness of the controller and TEF in decreasing the vertical displacement, also the uncontrolled cases are simulated. In Figure 5.4 and 5.5 the dynamic response of the airfoil to a cosine gust and Mexican hat gust is shown for both models with and without controller. On a global level both models perform similar: they reduce the maximum deflection of the airfoil significantly. However, differences can be found in both the level of reduction, flap deflection and the consequences of this deflection, on which the next sections elaborate.

5.3.1. UNCONTROLLED RESPONSE

First the response without controller is compared for both models. The force and displacement in y-direction, and the pitching moment and angle, are the most sensitive to the flap deflection and controller. Therefore, these parameters will be discussed in detail. In Tables 5.4 and 5.5 the absolute and relative differences between the response of the two models is given for both the gusts in controlled and uncontrolled conditions. The relative difference is the (absolute) maximum amplitude differences of the two model responses normalized by the band of the URANS response, stated in percentage.

For both the cosine and Mexican hat gust the lateral displacements and forces correspond well for the two models, with maximum relative differences around 5%. However, the pitch angle has a larger relative difference with values near 8%. The moment coefficients shows the largest relative differences, up to 62.9 %. However, it should be noted that the absolute differences for the moment coefficient and pitch angle are small, since the amplitudes themselves are small. Besides the large relative (but small absolute differences) differences in the pitch angle and moment both models behave very similar in amplitude response.

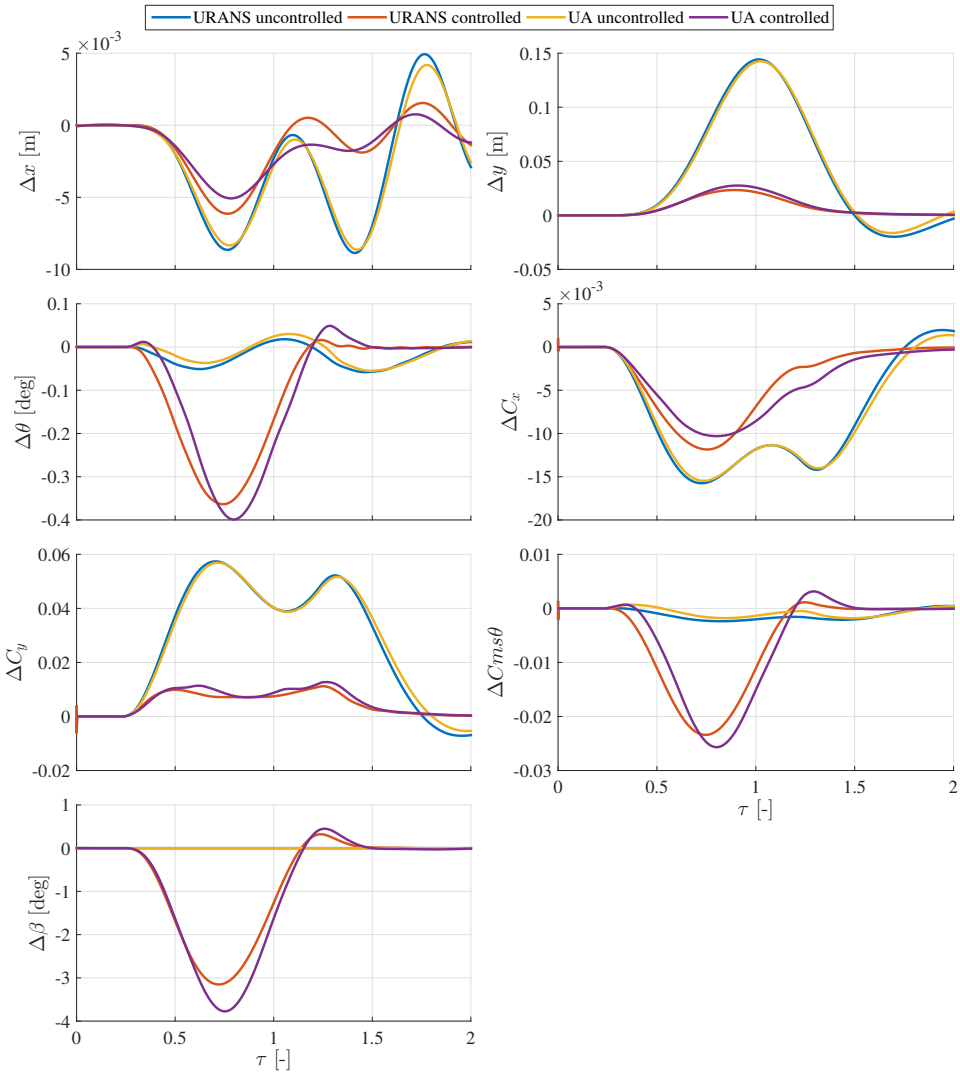

Figure 5.4: Response in time to a 1-cos gust with amplitude 1 m/s and frequency of 1.2 Hz for the unsteady aerodynamic (*UA*) model and the URANS model (*URANS*) for both the uncontrolled and controlled case. Controller gains are $K_v = -100$ and $K_a = -20$. The x-axis shows the normalized time: $\tau = t \times f_{gust}$.

5.3.2. CONTROLLED RESPONSE

Adding a controller, and thus a moving trailing edge flap, increases complexity, which both models handle differently. First, it is clear from both models that a significant reduction in vertical displacement can be achieved: 83% for the URANS model and 82% for the unsteady aerodynamic model in case of the cosine gust. For the Mexican hat gust reductions of 78% and 76% are found, respectively. The consequence of the flap actuation is an increase in both the moment and the related pitch angle. A deflection of the

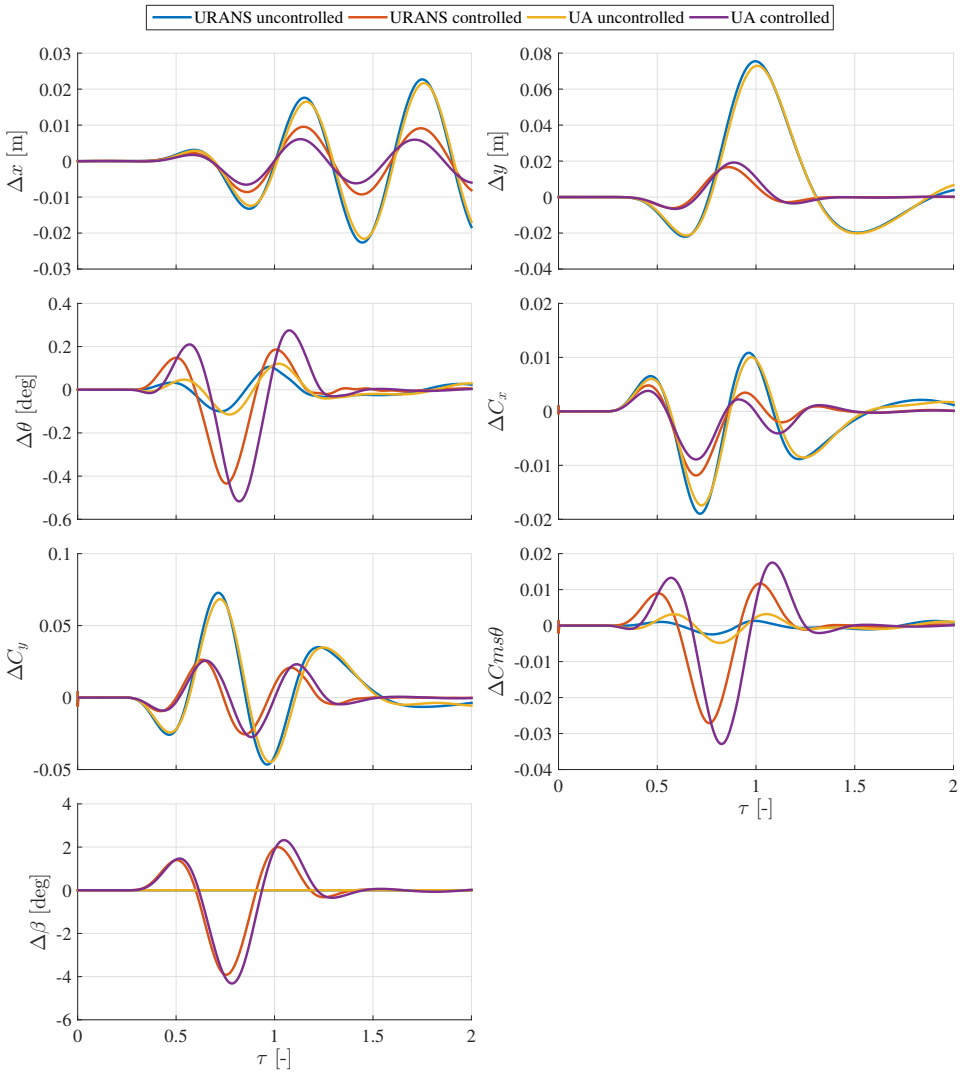

Figure 5.5: Response in time to a Mexican hat gust with amplitude 1 m/s and frequency of 1.2 Hz for the unsteady aerodynamic (*UA*) model and the URANS model (*URANS*) for both the uncontrolled and controlled case. Controller gains are $K_v = -100$ and $K_a = -20$. The x-axis shows the normalized time: $\tau = t \times f_{gust}$.

flap causes an additional force far away from the rotational centre, causing a steep increase in the moment. Here, the unsteady aerodynamic model predicts a larger change in pitch moment and angle when compared to the URANS model, partly caused by the larger flap deflection used.

As for the uncontrolled response, the responses of the two models are compared to assess the magnitude of the differences. For both the gusts, differences in the lateral displacements and forces and the pitch angle increase, with values near 17%. However,

Table 5.4: Absolute and relative differences between the unsteady aerodynamic model and URANS model in the cosine gust responses of a smart airfoil. Relative Differences are given in normalized maximum amplitude differences relative to the URANS results. Normalization is done by the band of the URANS response of the respective variable.

	Uncontrolled		Controlled	
variable	**Absolute**	**Relative**	**Absolute**	**Relative**
x [m]	0.00023	2.6%	0.0011	16.0%
C_x [-]	0.00027	1.7%	0.0015	13.1%
y [m]	0.0017	1.1%	0.0041	17.5%
C_y [-]	0.00048	0.8%	0.0015	13.8%
θ [deg]	0.0064	8.5%	0.0354	9.3%
C_{ms} [-]	0.00058	23.9%	0.0022	9.2%
β [deg]	-	-	0.62	17.8%

Table 5.5: Absolute and relative differences between the unsteady aerodynamic model and URANS model in the Mexican-hat gust responses of a smart airfoil. Relative differences are given in normalized maximum amplitude differences relative to the URANS results. Normalization is done by the band of the URANS response of the respective variable.

	Uncontrolled		Controlled	
variable	**Absolute**	**Relative**	**Absolute**	**Relative**
x [m]	0.0015	3.8%	0.0027	14.4%
C_x	0.0015	5.2%	0.0030	17.8%
y [m]	0.0025	2.6%	0.0025	11.1%
C_y	0.0044	3.7%	0.0019	3.7%
θ [deg]	0.014	7.0%	0.083	13.3%
C_{ms}	0.0024	62.9%	0.0058	14.9%
β [deg]	-	-	0.40	6.8%

the difference for the pitching moment (C_{ms}) decreased compared to the uncontrolled case, although they remain relative high with values up to 23.9%. These differences could be caused by the differences in flap deflection by the two models, which have a large influence on the moment coefficient. Adding an actively controlled flap to the system results in larger differences between the models, although the absolute differences are considered to be minor.

Considering the absolute differences the two models perform similar in unsteady response. The most significant difference is the flap deflection, with differences of approximately 0.5°. With maximum a flap deflection of 4.0° this difference is significant for designing the flap and controller.

5.3.3. PARAMETER STUDY

In the above analysis only a single frequency and angle of attack are considered. However, varying these two parameters influences the differences between the two models. Therefore, a (small) parameter study is performed by varying the angle of attack and gust frequency. This is done to illustrate how the differences between the two models vary for different flow types and different levels of unsteadiness.

Three angle of attacks are considered: $\alpha = 0°$, $\alpha = 8.5°$ and $\alpha = 15.0°$ corresponding to the angles in the linear region, start of separation and near maximum lift. The frequency of the gust is defined by varying the reduced frequency of the gust (based on the far-field velocity and semi-chord length) from 0.05 to 0.2 to 0.5, where previously only a reduced frequency of 0.062 was used.

Only, the Mexican hat gust is used for this parameter study with an amplitude equal to a 2.5° change in angle of attack, resulting in an amplitude of 2.66 m/s in the chord orthogonal direction. The controller gains are adjusted to $K_v = -15$ and $K_a = -8$ to ensure an stable response for all cases.

The amplitude and phase differences between the models, respectively ϵ^A and ϵ^ϕ, are shown in Figure 5.6. Emphasis in the comparison is on the vertical force, pitching

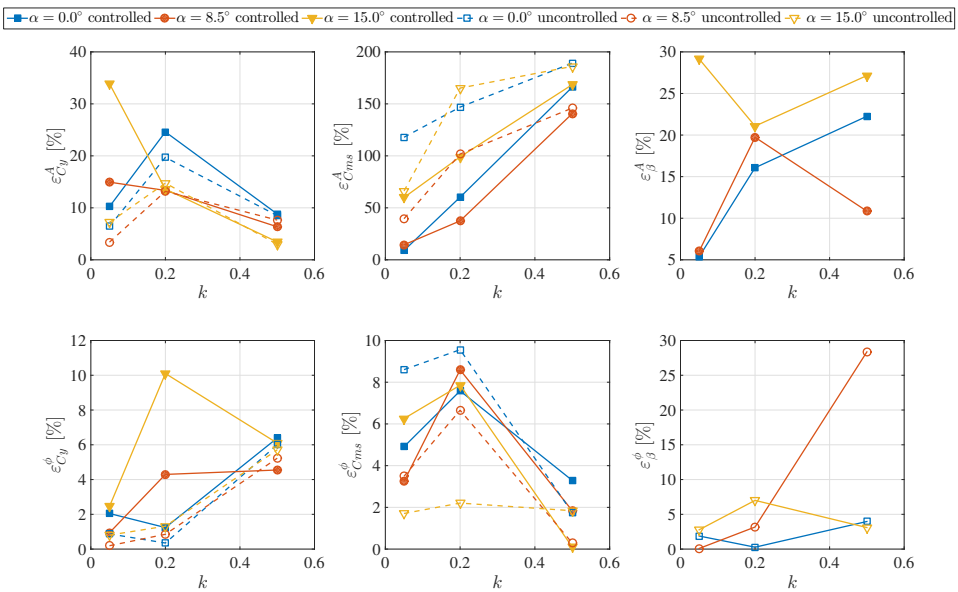

Figure 5.6: Relative differences between URANS and UA model for C_y, C_{ms} and β for varying angle of attack and reduced frequency. Differences are w.r.t. the URANS results. Both the controlled and uncontrolled response are shown with full and dashed lines, respectively.

moment and the flap deflections. Deflections become relatively small for higher reduced gust frequencies, due to the smaller amount of time the gust has to displace the airfoil: i.e. the structure acts as a low-pass filter. However, forces do respond more directly to the gusts and are, therefore, of greater interest for comparison.

At first, the largest relative differences are found within the moment coefficient and especially in the moment coefficient amplitude. However, both the uncontrolled and controlled responses in the moment coefficient are small in absolute value (up to 0.02). Phase differences in the moment coefficient remain relatively small.

At the lower angles of attack and reduced frequencies relative good correspondence is found in both amplitude and phase difference. However, the largest differences are present at an angle of attack in the non-linear regime at a relatively high reduced frequency (near 0.2). At this reduced frequency the gust still has significant energy to trigger a significant response, while at higher frequencies the differences decreases due to the limited time the gust influences the airfoil. In all presented variables significant differences can be found for the reduced frequency of 0.2 and higher angles of attack. One of the causes could be the large relative difference in pitch moment, causing a difference in angle of attack. This change in angle of attack would cause an increase/decrease in vertical force, triggering the vertical displacement to change. Since the controller is tuned to react upon such a displacement, the flap deflections become different resulting in a different moment. What the origin of the differences is, cannot be concluded from the data presented above. Therefore, future studies should be performed by eliminating parts of the model (degrees of freedom in the structure, controller, gust types).

To illustrate the different behavior found opposed to the first case presented with only a single angle of attack and frequency, the controlled and uncontrolled response of the airfoil for an angle of attack of 8.5° and a reduced gust frequency of 0.2 is shown in Figure 5.7. Here the differences shown in Figure 5.6 become more apparent: significant differences can be found between the results, especially in the moment and pitch angle, as well as in the vertical force and displacement. Initially the vertical displacement response is very similar, causing the flap deflection to be similar. However, as soon as the vertical displacement starts to deviate, also the flap deflections starts to deviate. This is expected, since the controller responds to the vertical velocity and acceleration.

5.4. CONCLUSION

In this chapter two aerodynamic models are used to simulate the aero-servo-elastic response of a smart wind turbine airfoil: the unsteady aerodynamic model and the URANS model. The goal of this study is to give a thorough quantification of the aero-servo-elastic response of a 3 degree of freedom structural model, while comparing the two different aerodynamic models. This is done by assessing the controlled and uncontrolled response to gusts.

Comparing the structural and aerodynamic response for the uncontrolled case indicates that the relative difference are small for the lateral components, while for the pitching angle and moment the relative difference are significantly larger (up to 65.1%). However, it must be noted that the absolute differences in the dynamic response for these parameters are small. For the controlled case an the differences in the lateral displacements and forces increase up to 17%. Also the flap angle shows similar differences, up to 17%. The pitching angle and moment do have a significant smaller relative difference for the cosine gust (between 9% and 14%). From this first part, it can be concluded that for the investigated cases the differences increase for the controlled direction (y)

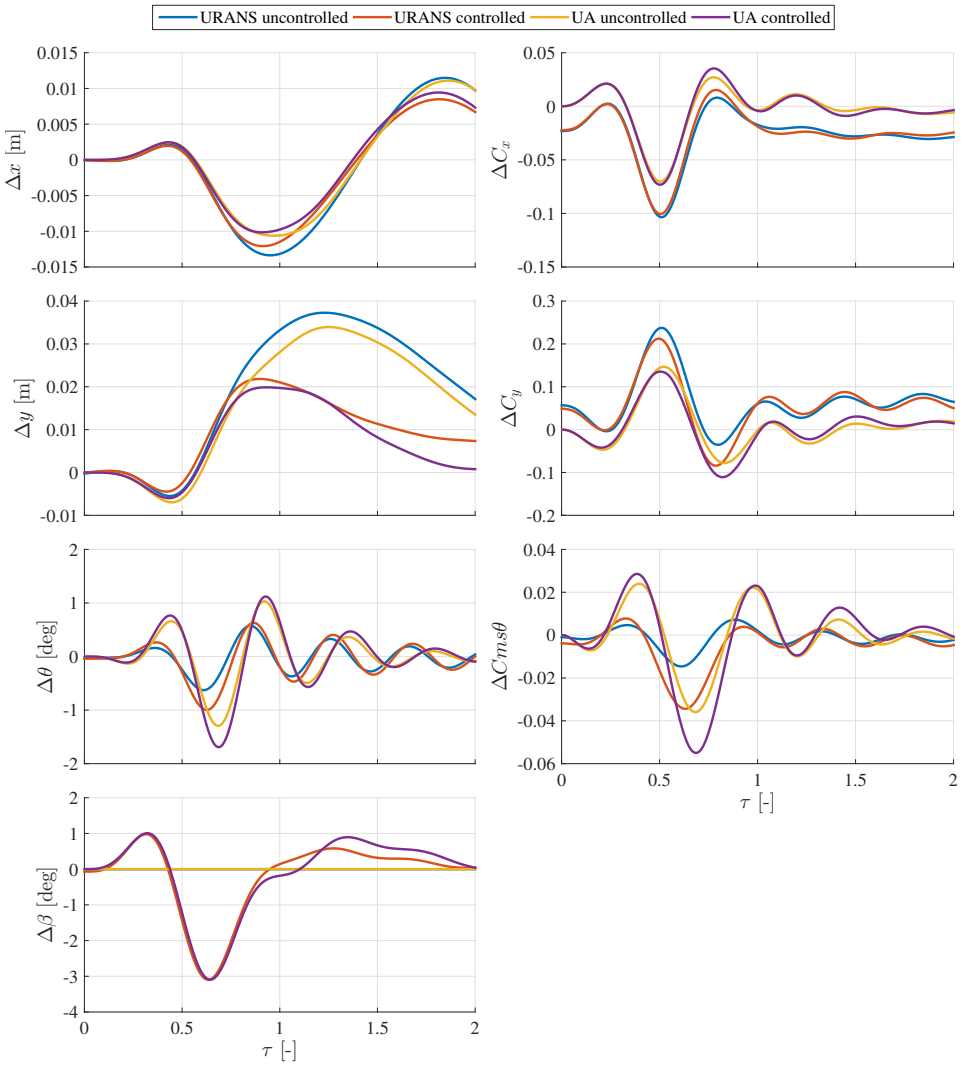

Figure 5.7: Response in time to a Mexican hat gust with amplitude equivalent to a 2.5° change in angle of attack and reduced frequency of 0.2 at an angle of attack of 8.5°. Results of both the unsteady aerodynamic (*UA*) model and the URANS model (*URANS*) are shown for both the uncontrolled and controlled case. Controller gains are $K_v = -15$ and $K_a = -8$. The x-axis shows the normalized time: $\tau = t \times f_{gust}$.

when a flap and controller are added.

In addition, both models do predict a similar reduction in vertical displacement: 83 and 82% for the cosine gust and 78 and 76% for the Mexican hat gust. However, due to the flap deflection a significant increase in pitching moment and angle is observed. This has also been observed in other studies.

Finally, a small parameter study has been performed as preliminary study to further

indicate the sensitivity of the two models to varying angle of attack and gust frequency. At the lower angles of attack and reduced frequencies relative good correspondence is found in both amplitude and phase difference. However, at intermediate reduced frequencies ($k = 0.2$) the difference increase. It is postulated that the change in moment coefficient causes an increase in pitch angle, resulting in a different angle of attack. Due to the difference in angle of attack the forces becomes different causing the displacements to change. Finally, this has an influence on the flap deflection and therefore changing the moment again. Future studies could focus on reducing the difference in moment coefficient prediction to possibly reduce the overall differences. With the goal of increasing the length (and therefore flexibility) of the blades, the torsional moment becomes more important in predictions due to the influence on the effective angle of attack and thus the other forces.

Two deterministic gusts are considered in this study, which already show a wide range of differences. From these inflow conditions it can be seen that in uncontrolled cases the observed behavior in amplitude is similar for both models, but that adding the controlled flap motion increases these differences. Therefore a more detailed comparison should be done in the future, using a wider range of gusts types, amplitudes and frequencies. Additionally, more attention must be given to the flap and its controller by using a more realistic controller design and including maximum rotational speed, delays, signal noise and maximum allowed (achievable) forces in the system. A first step would be to eliminate the structural model and purely looking at the force and moment response due to the gust with and without a controller. After this the lateral degrees of freedom could be added, eliminating the influence of the varying pitch angle.

REFERENCES

[1] J. Heinz, N. Sørensen, and F. Zahle, *Investigation of the load reduction potential of two trailing edge flap controls using CFD,* Wind Energy **14**, 449 (2011).

[2] L. Bergami and M. Gaunaa, *ATEFlap aerodynamic Model, a dynamic stall model including the effects of trailing edge flap deflection*, Risø-R-1792(EN) (2012).

[3] W. A. Timmer and R. P. J. O. M. van Rooij, *Summary of Delft University wind turbine dedicated airfoils,* Journal of Solar Energy Engineering **125**, 488 (2003).

[4] F. R. Menter, *Two-equation eddy-viscosity turbulence models for engineering applications,* AIAA Journal **32**, 1598 (1994).

[5] R. Singh and J. D. Baeder, *Direct calculation of three-dimensional indicial lift response using computational fluid dynamics,* Journal of Aircraft **34**, 465 (1997).

[6] D. P. Mol and W. A. Wall, *Partitioned analysis schemes for the transient interaction of incompressible flows and nonlinear flexible structures,* in *Trends in Computational Structural Mechanics* (International Center Numerical Methods Engineering, Schloss Hofen, Austria, 2001).

6

CONCLUSION

In both academia and industry, there is a desire to have high fidelity models of smart rotors, resulting in an increase in research projects and studies revolving around method development and validation of high fidelity models. The goal of this thesis is to increase the efficiency and confidence level of incompressible CFD based models for smart airfoils and rotors. Both on efficiency as well as on confidence of the model progress has been made within the research presented in this thesis.

To increase the efficiency, two parts of the FSI model contributing significantly to the computational work are considered: time integration for partitioned FSI models and mesh deformation. In Chapter 2 and 3 the results have been presented on the proposed methods, which both contribute to the first goal of the thesis: increasing efficiency. After this the aerodynamic model (including the methods presented in Chapter 2 and 3) is validated with experimental data, while also compared to an widely used engineering model (see Chapter 4). Finally, a typical application for the high fidelity model is shown by analyzing the response of an elastically suspended airfoil to a gust with and without controller connected to the TEF. Hereby, an efficient FSCI model for the 2D high fidelity model is presented in Chapter 5. Each of these parts are treated separately in the coming sections, after which conclusions on the main goal of the thesis will follow. Finally, an outlook is given for future studies on high fidelity smart rotor research.

6.1. TIME CONSISTENT FLUID-STRUCTURE INTERACTION

Because the FSCI model is intrinsically unsteady, time integration is needed within the model. For incompressible flow on collocated grids, non-linear momentum interpolation is needed, which makes time integration non-trivial. To ensure consistent time integration on a moving grid (needed when the ALE formulation of the incompressible Navier-Stokes is used) the time change in face area and face normal should be accounted for. The solution can either be formulated as two additional terms (for the face area and face normal change) in the face flux calculation, or as an evaluation of the consistent face velocities (both normal and tangential components) at their corresponding

grid from which the face fluxes will follow. Both formulations are presented in a general formula for backward differencing schemes and are shown for 1^{st}, 2^{nd} and 3^{rd} order backward differencing time integration. With the appropriate boundary condition, force coupling and structural time integration 3^{rd} order time integration is shown for the Fluid-Structure Interaction model. Finally, efficiency is increased with the consistent time integration by using a larger time step combined with a higher order time integration scheme. This is especially true for weakly interacting FSI problems, which a smart rotor model generally is.

6.2. ADAPTIVE RADIAL BASIS FUNCTION MESH DEFORMATION

By using the ALE formulation of the incompressible Navier-Stokes, the mesh needs to be deformed accordingly based on the structural deformation (including flap deflection). In Chapter 3 the Adaptive Radial Basis Function Mesh Deformation has been presented. Using a known data reduction method, to decrease to problem size adaptively, in combination with an automated approach of the explicit boundary correction, results in an robust and efficient mesh deformation method.

Robustness is achieved by two means: using the accuracy of the full RBF mesh deformation method, and limiting the user-dependency to only a small set of parameters. This last point is partly achieved by adaptively selecting the control points during a simulation, which eliminates the need of detailed a priori knowledge of the structural deformation. Secondly, an analytical analysis of a single deforming high aspect ratio cell, resulted in a general formula for the correction radius needed and two new correction functions. The proposed correction radius formula is validated by simulating a pitching airfoil with oscillating flap at four different Reynolds numbers (from $5 \cdot 10^5$ to 10^7). Good correspondence between the analytical function and results from the smart airfoil simulations is found, showing that the proposed automated explicit boundary correction approach is a robust method.

Efficiency is mainly achieved by the adaptive selection procedure, ensuring a small set of points is chosen as control points. Since the remaining part of the method has proper parallel scalability, the computational cost of the proposed method are lower when compared to the RBF methods presented in literature. In addition, with the new method ensuring a small set of points are required, the memory formulation of the method is within the possibilities of the current state of computers. Results for different cases have shown that the memory formulation shows a significantly lower computational time. However, parallel scalability is worse, since the bandwidth of the CPU to the memory is limited and does not scale. Depending on the hardware available the CPU formulation might become more efficient when a high number of cores per CPU are used.

With the proposed adaptive RBF mesh deformation method, a robust, efficient, accurate and almost user-independent mesh deformation method has been presented. The method has been applied to a 2D oscillating and translating airfoil with oscillating flap and performs well for all Reynolds numbers considered (up to 1×10^7). In combination with the consistent time integration a more efficient method has been developed for the FSCI of smart rotors/airfoils.

6.3. MODEL VALIDATION

The high fidelity model has been compared with experimental results and an widely used engineering model. A prescribed oscillating flap at different angles of attack, reduced frequencies and flap amplitudes is used for validation purposes of both the high fidelity model and engineering model.

One of the biggest contributors to the differences between experiments and high fidelity model results, are the steady-state results. Known to RANS calculations is the over-predictions of the lift coefficient in the non-linear region, which is also observed in this study for both the Spalart-Allmaras and kω-SST turbulence model. However, irregularities near the zig-zag strip are found within the forced-transition experimental data, which might lead to a reduction of the lift in the linear region. This is supported by the results from the free-transition, which show a very similar behavior for the lift as the RANS results. For the engineering model the input steady-state values are obtained from an interpolation from the experimental steady-state results, ensuring good correspondence between the steady-state results from the engineering model and the experimental results. However, the method of the interpolation is crucial for proper predictions of both the unsteady and steady results.

From the validation study, it becomes clear that the varying lift coefficient is reasonably well predicted by both models. However, the models predict a reduction in flap effectiveness which is not seen in the forced-transition results, but are found in the free-transition results. Having a closer look at pressure distributions, irregularities are found near the flap hinge, which might be caused by the seal, measurement equipment (time delay) or post-processing. The unsteady contribution of the drag is significantly over predicted by both models. For the moment coefficient, the high fidelity model predicts the unsteady behavior reasonably well, but the engineering model fails to predict the unsteady contribution properly. However, variations within the moment are relatively small compared to the steady-state variations.

From the validation it becomes clear that both the engineering model and the URANS model perform well in terms of unsteady lift prediction of a TEF. Most important for the engineering model is the proper input data, since this is where most of differences can arise from. In addition, the focus for the URANS model should be on improving the stead-state predictions and unsteady drag predictions. Finally, the experimental data and experiments should be further analyzed to determine the cause of the irregularities and their influence on the final results.

6.4. MODEL APPLICATION

After the validation study the two aerodynamic models are used to simulate the aero-servo-elastic response of a smart wind turbine airfoil: the unsteady aerodynamic model and the URANS model. Both the controlled and uncontrolled aero-servo-elastic response to two deterministic gusts has been used to quantify differences between the two aerodynamic models for such coupled simulations.

At conditions where the aerodynamics is mainly linear and the gust frequency is low (reduced frequency below 0.1), both models correspond well when comparing the uncontrolled response for both gusts. Adding the controller, and thus flap motion, in-

creases the differences between the models. Differences in the forces, moment, translation and torsion are approximately 15%. It should be noted that both models predict a similar reduction in vertical displacement, which is where the controller is tuned for. However, when increasing the angle of attack and gust reduced frequency (up to 0.2), the differences in both the controlled and uncontrolled response increase. Especially the moment coefficient shows larger differences, which has an effect on the pitch angle and thus on the angle of attack. The most significant absolute differences are found for the flap deflections, which vary between the two models up to 0.5°.

From these results a closer look should be taken at the moment predictions of both models (and especially the unsteady aerodynamic model). These differences, although relatively small in amplitude, do cause a change in pitch angle, changing the effective angle of attack and therefore both the steady and unsteady response of the aero-servo-elastic model. With the increase in flexibility of the blades, the torsional moment becomes more important in predictions due to the influence on the effective angle of attack and thus the other forces. A first step would be to eliminate the structural model and purely study the moment (and force) response due to the gust with and without a controller.

6.5. FLUID-STRUCTURE-CONTROL INTERACTION FOR SMART ROTORS

Both the time integration method as well as the RBF method, are ready for large scale (3D) problems and thus for application within the FSCI model of a smart rotor. With the validation study and the direct comparison of the aero-servo-elastic response to a gust, first steps are made to increase the confidence of the method, or at least to quantify its accuracy compared to experiments and an engineering model.

6.6. OUTLOOK

With the newly gained knowledge in consistent time integration on collocated grids for incompressible flow, RBF mesh deformation and high fidelity modeling of a smart airfoil, an outlook is formulated on these aspects and their combinations.

6.6.1. TIME CONSISTENT FLUID-STRUCTURE INTERACTION

With the general formulation of the backward differencing schemes, higher order time integration methods can derived for the incompressible flow on moving collocated grids. A first step would be to develop the formulation for multi-stage methods, such as Runge-Kutta schemes. Currently, studies are being performed within the research group on the application of these methods.

Finally, the method has been applied to segregated sets of equations. The theory developed in this thesis could also be applied to the coupled system of equations (momentum, pressure and turbulence equations). However, additional effort is required to test and develop such an implementation.

6.6.2. RADIAL BASIS FUNCTION MESH DEFORMATION

The presented adaptive radial basis function mesh deformation already ensures a high level of automation and user-independency, while being efficient. However, there are 3 aspects of the method, which would potentially increase either the robustness or efficiency when improved upon: solving system of equation within the greedy algorithm, orthogonality deformation and accuracy of solving system of locally supported radial basis functions.

First, the most expensive non-parallel part of the method is sequently solving a system of equation within the greedy algorithm. Even though a single solve is not expensive due to the size of the system, the sequential nature of the method causes this to be the bottle-neck when scaling up the problem size. Two possible improvements are possible: change the selection algorithm in a non-sequential method, in which the system of equations does not need a new solve for each selection step. However, the number of points required should not increase, since this would directly increase the computational time of the interpolation part of the method. A second option would be to develop/use a method, which allows for parallel scaling within the selection algorithm, resulting in almost fully parallel scalable method.

Secondly, for higher Reynolds numbers and thus (very) high aspect ratio cells near the wall, the orthogonality becomes a problem. Even though the proposed automated explicit correction method ensures that the quality is not decreased due to the explicit correction, it does not decrease the non-orthogonality compared to the full RBF mesh deformation. Especially, in linear elastics, even the full RBF mesh deformation method has problems in maintaining a high orthogonality near the wall. A solution for this problem is to use the surface normals directly within the RBF mesh deformation, to interpolate these towards the inner mesh, ensuring a higher orthogonality at the wall. First steps are made, by combining this approach with the greedy algorithm, resulting in a improved mesh quality. However, next studies should focus on the application on large scale problems in combination with efforts to improve its efficiency.

Finally, during the study presented in this thesis, problems arose when using locally supported radial basis functions at high aspect ratio cell meshes. Locally supported radial basis functions lead to a less well conditions system of equations, resulting in a lower accuracy of the solution from the system of equations. When the accuracy is in the order of the first cell height, the mesh quality reduces significantly due to numerical round of errors. Improvements should be made to ensure that the numerical accuracy does not limit the application of these locally supported functions, since the often offer a higher efficiency due to the limited number control points needed.

6.6.3. HIGH FIDELITY MODELING OF SMART ROTORS

Finally, with the validation study and the application of the FSCI model to a 2D controlled airfoil responding to gusts, first steps are made in increasing the confidence of the method, while also identifying the problems still present. Two paths are suggested: 1) a more detailed study of the 2D response to identify possible improvements on both the high fidelity model and the engineering model, 2) continue towards 3D FSCI simulations of the smart rotor.

A step-by-step approach could be taken within the 2D comparison of the high fidelity model and the engineering model. First, the experimental data for the static airfoil should be investigated such that the validation study can be finalized. Secondly, only a force response to gusts, without structure or controller, should be analyzed to determine the if the differences in moment arise from here. Thirdly, adding a controller minimizing the vertical force, could identify the influence of the flap on the moment in these conditions, without the influence of the structural deformation. Fourthly, only the lateral degrees of freedom could be added to eliminate the influence of the moment on the pitch angle and thus angle of attack. Hereafter, adding the full 3 degree of freedom model in combination with a more realistic controller should provide a good insight in the origin of the differences between the two models. Finally, experimental validation of such a system would be great. First efforts are made by using linear actuators connected a numerical structural model and pressure sensors on the airfoil to mimic the fluid-structure interaction, while eliminating the uncertainties of the structural model.

A similar approach can be taken within future 3D studies. Both the time integration and the RBF mesh deformation can be applied directly for 3D problems. Having a discrete flap would require additional effort to cope with the discontinuities near the flap edges, but both overset grids as well as arbitrary mesh interfaces can cope with this. Comparing the 3D FSCI model results with results from an engineering model would provide valuable insight in the differences caused by the modeling of the flap on the overall performance of actively controlled trailing edge flaps on wind turbines. The presented methods and results give a first step towards such a study.

A

RBF MESH DEFORMATION

A.1. GREEDY SELECTION METHOD

As proposed by Rendall and Allen [1], the greedy algorithm is used to create subset of boundary points to control the deformation of the internal points. With an initial set of of control points ($\vec{\mathbf{x}}_c$), the interpolation condition is formulated:

$$\Delta\vec{\mathbf{x}}_c = \Phi_{c,c}\vec{\boldsymbol{\alpha}}_c, \tag{A.1}$$

where $\Delta\vec{\mathbf{x}}_c$ are the displacements of the control points $\vec{\mathbf{x}}_c$, $\Phi_{c,c}$ are the basis functions from control points to control points and $\vec{\boldsymbol{\alpha}}_c$ are the basis function coefficients for the control points. By solving this system the coefficients are obtained and the set of control points is used to determine the interpolated displacement of the complete set of boundary points:

$$\Delta\vec{\mathbf{x}}_b^* = \Phi_{b,c}\vec{\boldsymbol{\alpha}}_c. \tag{A.2}$$

Here $\Delta\vec{\mathbf{x}}_b^*$ are the interpolated boundary point displacements and $\Phi_{b,c}$ is the matrix containing the basis functions from control to boundary points. The maximum error between the interpolated displacement and the actual displacement of the boundary points is used to select a new control point:

$$\vec{\boldsymbol{\varepsilon}}_b = \Delta\vec{\mathbf{x}}_b^* - \Delta\vec{\mathbf{x}}_b \tag{A.3}$$

$$\left[\varepsilon_b^{max}, \vec{x}_{b\to c}^{\,n}\right] = \max\|\vec{\boldsymbol{\varepsilon}}_b\|_2 \tag{A.4}$$

where ε_b^{max} is the maximum boundary error and $\vec{x}_{b\to c}^{\,n}$ is the boundary point associated with the largest error, which is added to the set of control points. With the new set of control points the process is repeated from Equation A.1 until a convergence criteria is satisfied. The choice of convergence criteria is such that the selected set of control points is able to deform the internal mesh accordingly based on deformed shape of the boundary. Secondly, the influence of the amplitude of the deformation on the set of selected points is minimal, since radial basis function behave linearly with the displacement. Therefore,

the convergence criteria are normalized by the boundary displacement:

$$\frac{\|\vec{\boldsymbol{\varepsilon}}_b\|_2}{\|\Delta\vec{\mathbf{x}}_b\|_2} < \varepsilon_{tol} \qquad \&\& \qquad \frac{\|\vec{\boldsymbol{\varepsilon}}_b\|_\infty}{\|\Delta\vec{\mathbf{x}}_b\|_\infty} < \varepsilon_{tol} \tag{A.5}$$

where ε_{tol} is the user specified tolerance for the greedy selection. By normalizing the boundary error norm with the norm of the displacement the set of control points becomes (almost) independent of the amplitude and mainly dependent on the shape, which is desirable. In this way a set of points is chosen, which can present the shape until a certain accuracy (ε_{tol}). Summarizing the greedy algorithm by Procedure 2 and Algorithm 2.

Procedure 2 Boundary Error ($\vec{\mathbf{x}}_b, \vec{\mathbf{x}}_c, \Delta\vec{\mathbf{x}}_b, \Delta\vec{\mathbf{x}}_c$)

$\vec{\boldsymbol{\alpha}}_c = \Phi_{c,c}^{-1}\Delta\vec{\mathbf{x}}_c$
$\Delta\vec{\mathbf{x}}_b^* = \Phi_{b,c}\vec{\boldsymbol{\alpha}}_c$
$\vec{\boldsymbol{\varepsilon}}_b = \Delta\vec{\mathbf{x}}_b^* - \Delta\vec{\mathbf{x}}_b$
return $\vec{\boldsymbol{\varepsilon}}_b$

Algorithm 2 Greedy ($\vec{\mathbf{x}}_b, \vec{\mathbf{x}}_c, \Delta\vec{\mathbf{x}}_b, \Delta\vec{\mathbf{x}}_c$)

repeat
  $\vec{\boldsymbol{\varepsilon}}_b$ = Boundary Error ($\vec{\mathbf{x}}_b, \vec{\mathbf{x}}_c, \Delta\vec{\mathbf{x}}_b, \Delta\vec{\mathbf{x}}_c$)
  $\varepsilon_b^{max} = 10^{256}$
  $i_{max} = -1$
  for $i = 0$ **to** n_b **do** # Find largest boundary error
    if $\|\vec{\boldsymbol{\varepsilon}}_b[i]\|_2 < \varepsilon_b^{max}$ **then**
      $\varepsilon_b^{max} = \|\vec{\boldsymbol{\varepsilon}}_b[i]\|_2$
      $i_{max} = i$
    end if
    $\vec{\mathbf{x}}_c = [\vec{\mathbf{x}}_c, \vec{\mathbf{x}}_b[i_{max}]]$ # Update set of control points with newest boundary point
    $\Delta\vec{\mathbf{x}}_c = [\Delta\vec{\mathbf{x}}_c, \Delta\vec{\mathbf{x}}_b[i_{max}]]$
  end for
until $\left(\frac{\|\vec{\boldsymbol{\varepsilon}}_b\|_2}{\|\Delta\vec{\mathbf{x}}_b\|_2} < \varepsilon_{tol} \,\&\&\, \frac{\|\vec{\boldsymbol{\varepsilon}}_b\|_\infty}{\|\Delta\vec{\mathbf{x}}_b\|_\infty} < \varepsilon_{tol}\right)$

A.2. DERIVATION OF ASPECT RATIO DEPENDENCY

In this appendix the analytical relation between aspect ratio, displacement and the cell non-orthogonality (λ) is derived. This is done based on a rectangular cell, which is deformed by means of a translation of a single point in the direction parallel to the smallest rectangle side. Figure A.1 an illustration of this cell and it's deformation is shown. Here the x coordinate is chosen as coordinate in the longest edge direction and the y coordinate is chosen as coordinate in the shortest edge direction. To calculate the increase of

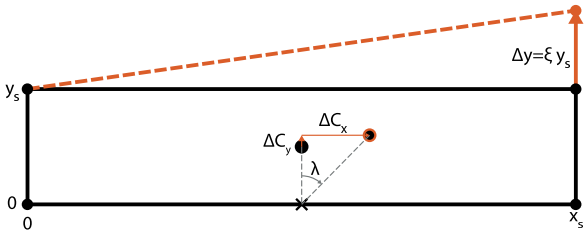

Figure A.1: A single rectangular cell and its displacement of the right top point. Due to this displacement the cell center will move, causing in increase in non-orthogonality.

non-orthogonality, the new cell center location is required:

$$\lambda = \tan^{-1}\left(\frac{C_x^d - C_x^0}{C_y^d}\right) = \tan^{-}1\left(\frac{\Delta C_x}{C_y^d}\right), \tag{A.6}$$

where C_x^0 is the x-coordinate of the cell center of the original undeformed cell, C_x^d is the x-coordinate of the cell center of the deformed cell, C_y^d is the y-coordinate of the cell center of the deformed cell, ΔC_x is the displacement of the cell center from it's original location in x-direction and ΔC_y is the displacement of the cell center from it's original location in y-direction. The cell center coordinates of a cell is calculated by:

$$C_x = \frac{1}{6S}\sum_{i=1}^{n_p}(x_i + x_{i+1})\left(x_i y_{i+1} - x_{i+1} y_i\right) \tag{A.7}$$

$$C_y = \frac{1}{6S}\sum_{i=1}^{n_p}\left(y_i + y_{i+1}\right)\left(x_i y_{i+1} - x_{i+1} y_i\right), \tag{A.8}$$

where S is the area of the cell:

$$S = \frac{1}{2}\sum_{i=1}^{n_p} x_i y_{i+1} - x_{i+1} y_i. \tag{A.9}$$

Here, n_p is the number of points of the cell and i is the index of the point in a anti-clockwise manner. For the non-deformed, and thus rectangular cell, the cell center is easily calculated by simple dividing the length of the short and long edge by two:

$$S^0 = y_s x_s \tag{A.10}$$

$$C_x^0 = \frac{1}{2}x_s \tag{A.11}$$

$$C_y^0 = \frac{1}{2}y_s. \tag{A.12}$$

For the deformed cell the area becomes:

$$\begin{aligned} S^d &= \frac{1}{2}\left[(0 - x_2 0) + \left(x_2 y_3 - x_3 0\right) + \left(x_3 y_4 - 0 y_3\right) + \left(0 - 0 y_4\right)\right] \\ &= \frac{1}{2}\left[x_2 y_3 + x_3 y_4\right] \\ &= \frac{1}{2}\left[x_s\left(y_s + \Delta y\right) + x_s y_s\right] = x_s y_s + \frac{1}{2}\Delta y x_s, \end{aligned} \tag{A.13}$$

and the deformed cell center becomes in x-direction becomes:

$$\begin{aligned} C_x^d &= \frac{1}{6S}\left[(x_2+x_3)\,x_2 y_3 + x_3 x_3 y_4\right] \\ &= \frac{1}{6S}\left[2x_s^2\left(y_s+\Delta y\right)+x_s^2 y_s\right] \\ &= \frac{1}{6S}\left[3x_s^2 y_s + 2x_s^2\Delta y\right] \\ &= \frac{3x_s^2 y_s + 2x_s^2\Delta y}{6\left(x_s y_s + \frac{1}{2}x_s\Delta y\right)} \\ &= \frac{4x_s\left(x_s y_s + \frac{1}{2}x_s\Delta y\right) - x_s^2 y_s}{6\left(x_s y_s + \frac{1}{2}x_s\Delta y\right)} \\ &= \frac{2}{3}x_s - \frac{1}{6}\frac{x_s y_s}{y_s+\frac{1}{2}\Delta y}, \end{aligned} \tag{A.14}$$

and in y-direction:

$$\begin{aligned} C_y^d &= \frac{1}{6S}\left[\left(y_s+\Delta y\right)^2 x_s + \left(2y_s+\Delta y\right)x_s y_s\right] \\ &= \frac{1}{6S}\left[\left(y_s+\Delta y\right)^2 x_s + 2y_s^2 x_s + \Delta y\, y_s x_s\right] \\ &= \frac{1}{6}\frac{\left(y_s+\Delta y\right)^2 x_s + 2y_s^2 x_s + \Delta y\, y_s x_s}{x_s y_s + \frac{1}{2}x_s\Delta y} \\ &= \frac{1}{3}y_s + \frac{1}{6}\frac{\left(y_s+\Delta y\right)^2}{y_s+\frac{1}{2}\Delta y}. \end{aligned} \tag{A.15}$$

Assuming $\Delta y = \xi y_s$, the cell center coordinates are:

$$C_x^d = x_s\left(\frac{2}{3} - \frac{1}{6}\frac{y_s}{\left(y_s+\frac{1}{2}\xi y_s\right)}\right) = x_s\left(\frac{2}{3} - \frac{1}{6}\frac{1}{1+\frac{1}{2}\xi}\right) \tag{A.16}$$

$$C_y^d = \frac{1}{3}y_s + y_s\frac{1}{6}\frac{(1+\xi)^2}{1+\frac{1}{2}\xi} = y_s\frac{1}{6}\left(2+\frac{(1+\xi)^2}{1+\frac{1}{2}\xi}\right) = \frac{1}{6}y_s\frac{3+3\xi+\xi^2}{1+\frac{1}{2}\xi}. \tag{A.17}$$

With C_x^d the displacement of the coordinate in x-direction is derived:

$$\begin{aligned} \Delta C_x &= C_x^d - C_x^0 = x_s\left(\frac{2}{3} - \frac{1}{6}\frac{1}{1+\frac{1}{2}\xi} - \frac{1}{2}\right) = x_s\left(\frac{1}{6} - \frac{1}{6}\frac{1}{1+\frac{1}{2}\xi}\right) \\ &= x_s\frac{1}{6}\left(1 - \frac{1}{1+\frac{1}{2}\xi}\right) = x_s\frac{1}{12}\frac{\xi}{1+\frac{1}{2}\xi}. \end{aligned} \tag{A.18}$$

Finally, the argument of the tan^{-1} function for the non-orthogonality is found:

$$\begin{aligned} \frac{\Delta C_x}{C_y^d} &= \frac{x_s\left(\frac{1}{12}\xi\right)/\left(1+\frac{1}{2}\xi\right)}{y_s\frac{1}{6}\left(3+3\xi+\xi^2\right)/\left(1+\frac{1}{2}\xi\right)} \\ &= \frac{1}{2}\frac{\xi A}{3+3\xi+\xi^2}. \end{aligned} \tag{A.19}$$

where $A = \frac{x_s}{y_s}$ is the aspect ratio of the cell.

REFERENCES

[1] T. C. S. Rendall and C. B. Allen, *Efficient mesh motion using radial basis functions with data reduction algorithms,* Journal of Computational Physics **228**, 6231 (2009).

B

RADIAL BASIS FUNCTION MESH DEFORMATION INCLUDING BOUNDARY ORTHOGONALITY

B.1. INTRODUCTION

Radial Basis Functions have been gaining interest in the past decade due to its robustness, flexibility and achieved mesh quality. However, one of the problems which could arise is the non-orthogonality when considering linear elastic structures and thus linear deformations. In addition, as higher Reynolds numbers, the non-orthogoanlity for the first cells near the boundary (having high aspect ratios) degrades fast, due to their high sensitivity (derived in Appendix A.2) to small displacements. Therefore, using the measure of non-orhtogonality at boundaries could increase the mesh quality.

Attempts are made to include the orthogonality (direclty) in the mesh deformation method. For example in the Laplacian smoothing method by using quaternions as shown by Maruyama et al. [1]. Rendall and Allen have derived that a selection of Radial Basis Functions is very good in preserving the mesh orthogonality near the wall [2]. This is due to smooth derivative of the Radial Basis Functions near the wall. The orthogonality near the wall is also most crucial, since for FSI applications this is where the pressure should be solved most accurately. However, this is partly depending on the deformation type and direction. For linear elastic like deformations (where deformations in different directions are uncorrelated), this does not necessary hold. This is due to the independency of the interpolation in different directions for Radial Basis Function interpolation [2]. Therefore, the goal of this draft paper is to assess the possibilities of incorporating orthogonality in RBF mesh deformation without a large increase in cost.

Orthogonality is taken into account by incorporating a second set of point, which are related to the original surface points by means of the surface normal (as proposed by Gillebaart et al. [3]). This second set of points are used to limit the change in orthogonal-

ity near the wall. Since adding a second set of points will increase the computational cost significantly methods of reducing the used points are assessed. The basis is the greedy method as presented by Rendall and Allen for the original RBF mesh deformation [4]. They proposed a surface point selection algorithm, which ensures a minimum error in surface representation. In this study the same algorithm is used as basis to develop two new methods to ensure a minimum change in orthogonality.

At first the RBF method and its greedy selection algorithm is shortly explained, after which the new orthogonal preserving algorithm is discussed. An academic test case with linear elastic deformation will be used to show the effectiveness of the method and how the modified greedy algorithms are used in the new method.

B.2. METHOD

In this section the Radial Basis Function mesh deformation, as discussed in De Boer et al.[5], is stated first, after which the greedy algorithm is shortly explained (as presented by Rendall and Allen[4]). After this the new part of the method to preserve the orthogonality better is discussed.

B.2.1. RADIAL BASIS FUNCTION MESH DEFORMATION

The RBF mesh deformation method is based on interpolating the surface displacements to the internal mesh by means of radial basis functions:

$$s(\mathbf{x}) = \sum_{j=1}^{n_b} \alpha_j \phi\left(\left\|\mathbf{x} - \mathbf{x}_{b_j}\right\|\right), \tag{B.1}$$

where $\mathbf{x}_{b_j} = \left[x_{b_j}, y_{b_j}, z_{b_j}\right]$ are the boundary centers at which the displacement is known, ϕ is the radial basis function as function of the Euclidean distance $\|\mathbf{x}\|$, n_b are the number of boundary points and α_j is the function coefficient. In addition to the interpolation function the following condition must be satisfied:

$$s\left(\mathbf{x_{b_j}}\right) = \Delta\mathbf{x}_{b_j}, \tag{B.2}$$

where $\Delta\mathbf{x}_b$ are the displacements of the boundary points. Combining this condition with Eq. (B.1) the coefficients α_j are determined:

$$\Delta\mathbf{x}_{b_j} = \Phi_{b,b}\alpha. \tag{B.3}$$

Here $\Phi_{b,b}$ is the matrix containing n_b by n_b RBF function evaluations: $\Phi_{b_i,b_j} = \phi\left(\left\|\mathbf{x}_{b_i} - \mathbf{x}_{b_j}\right\|\right)$. With the obtained coefficients (α) the internal point deformation is interpolated:

$$\Delta\mathbf{x}_{in_j} = s\left(\mathbf{x}_{in_j}\right). \tag{B.4}$$

The displacement in each spatial direction can be interpolated separately. In matrix notation this becomes:

$$\Delta\mathbf{x}_{in_j} = H_{in,b}\alpha \tag{B.5}$$

B.2.2. GREEDY SELECTION

The cost of the RBF mesh deformation is scaling with both the number of internal points (n_{in}) as well as the number of boundary points (n_b). Rendall and Allen proposed the greedy algorithm for reducing the number of boundary point by means of selecting the boundary points with the largest geometrical error until a specified maximum error is ensured [4]. The error on the boundary is given by:

$$\epsilon_b = \Delta\mathbf{x}_b - \overbrace{H_{b,c}\Phi_{c,c}^{-1}\Delta\mathbf{x}_c}^{\widetilde{\Delta\mathbf{x}}_b}, \tag{B.6}$$

where $\Delta\mathbf{x}_c$ is the displacement of the control points, which is a subset of the boundary displacements ($\Delta\mathbf{x}_b$), $H_{b,c}$ contains the function evaluations from control points to boundary points and $\Phi_{c,c}^{-1}$ is the inverse of the system of equations for the control points obtained from Eq. (B.3). The greedy algorithm starts with a set of points (denoted by $\mathbf{x}_{c_0}$) and evaluates the error as stated in Eq. (B.6). Consequently, the boundary point related to the maximum boundary error:

$$\mathbf{x}_{c_{new}} \leftarrow max\left(\epsilon_b\right), \tag{B.7}$$

is added to the subset of control points. With the newly obtained set of control points the new boundary error is calculated and the new control point is selected. This process repeats itself until either all points are selected (resulting in a zero boundary error) or when a tolerance on the maximum error is reached. At the end a reduced set of points $\mathbf{x}_c$ is obtained, which is used to deform the internal points.

B.2.3. ORTHOGONAL DRIVEN MESH DEFORMATION

In general RBF mesh deformation is performing very well in preserving orthogonality of the mesh (near the walls)[2, 6]. However, this is not directly ensured in the original framework of the RBF mesh deformation. Cases with linear elastic deformations can cause problems on this aspect, as will be shown with an academic case. The proposed method is based on adding a second set of points near the boundary to incorporate the orthogonality of the wall. This second set of points is a point in the normal direction of the wall at a distance of half the first cell height:

$$\mathbf{x}_{e_j}^0 = \mathbf{x}_{b_j}^0 + \mathbf{n}_{b_j}^0\left(\frac{1}{2}h_{c_j}^0\right), \tag{B.8}$$

where superscript 0 indicates the value at the initial (undeformed) state, $\mathbf{n}_{b_j}$ is boundary normal of point j and h_{c_j} is the initial first cell height of cell next to the boundary point. The prescribed deformation of the new set of points is given by:

$$\Delta\mathbf{x}_{e_j} = \Delta\mathbf{x}_{b_j} + \left(\mathbf{n}_{b_j}^n - \mathbf{n}_{b_j}^0\right)\left(\frac{1}{2}h_{c_j}^0\right), \tag{B.9}$$

where superscript n indicates the value at the new deformed state. This points are added to the boundary points in Eq. (B.1) and the interpolation condition in Eq. (B.2). With

this the change in orientation can be incorporated. However, this will make the algorithm significantly more costly. Therefore, the greedy algorithm is also used to determine which subset of points is used for both a good geometrical approximation as well as a good orthogonality. The error in the orthogonality is determined by means of the angle between the normals:

$$\theta_{b_j} = \cos^{-1}\left(1 - \mathbf{n}^n_{b_j} \cdot \tilde{\mathbf{n}}^n_{b_j}\right), \tag{B.10}$$

where

$$\tilde{\mathbf{n}}^n_{b_j} = \frac{\tilde{\mathbf{x}}^n_{b_j} - \tilde{\mathbf{x}}^n_{e_j}}{\left\|\tilde{\mathbf{x}}^n_{b_j} - \tilde{\mathbf{x}}^n_{e_j}\right\|}, \tag{B.11}$$

with $\tilde{\cdot}$ indicating the estimated value by means of RBF interpolation using the subset of points. As with the geometrical point selection, also for the angle a tolerance is given: θ_{tol}. When both the convergence criteria are satisfied the selection of points is stopped and the final subset is obtained. Two approaches are considered for coping with the two tolerances: always selecting a set of two points (one new $\mathbf{x}_{b_j}$ and its corresponding $\mathbf{x}_{e_j}$), only selecting the corresponding point ($\mathbf{x}_{e_j}$) when the maximum angle is larger than the tolerance. In both approaches the selection can either be geometrically or orthogonal driven.

B.3. RESULTS

To demonstrate a first application of the proposed method, a 1D dimensional deforming membrane in a 2D channel is used. This is illustrated in figure B.1. The domain

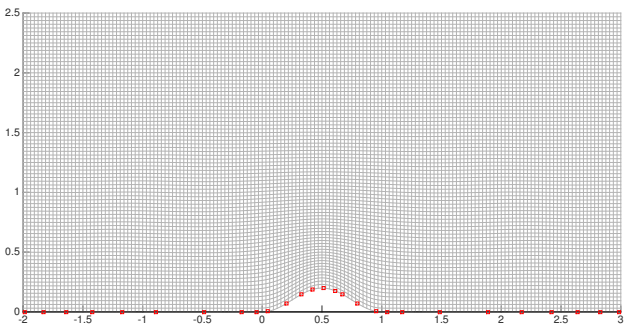

Figure B.1: Deformed mesh of 1D deforming membrane in channel case obtained with original RBF and original greedy. Squares indicate the used surface points.

spans from -2 to 3 in x-direction and from 0 to 2.5 in y-direction. Number of points in x-direction (N_x) is 161, while in y-direction 81 points are used (N_y). At the bottom of the

domain a displacement in y-drection is given by:

$$\Delta y_b = 0.1\left(\frac{1}{2} - \frac{1}{2}\cos\left(2 * pi * x_b\right)\right) \text{ for } 0 \leq x_s \leq 1 \tag{B.12}$$

$$\Delta y_b = 0.0 \text{ for } x_s > 1 \text{ and } x_s < 0 \tag{B.13}$$

In x-direction the displacement is zero, causing the degeneration of the orthogonality, as can be seen in figure B.1. The other boundaries are static. A compact supported WendlandC2 with a radius of 2 is used for the mesh deformation.

B.3.1. ORTHOGONAL RBF

First the RBF mesh deformation is considered without the greedy algorithm. The orthogonality on the bottom surface is shown in figure B.2 for the original RBF and the orthogonal RBF. For the orthogonal RBF both the actual orthogonality and the measured orthogonality, which will be used in the greedy algorithm, is shown. The orthogonality

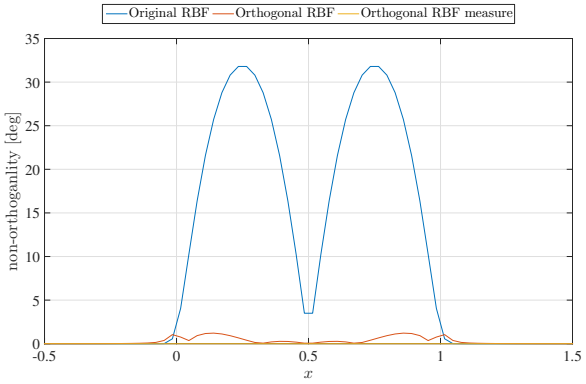

Figure B.2: Non-Orthogonality on bottom surface for original RBF and Orthogonal RBF.

is greatly reduced by adding the extra points: from a maximum of 31.9 to a maximum of 1.3. However, the number of boundary points used is twice as big: 160 for the original RBF and 320 for the orthogonal RBF. To limit the total number of points the greedy selection is used. Each greedy iteration one set of points is added. Depending on if the algorithm is geometrically driven or orthogonal driven the point set with the maximum geometrical error or the point set with the largest orthogonality error is used, respectively. The results of both approaches is given in figure B.3. It can indeed be seen that the greedy algorithm works well in both approaches. However, adding two points each greedy iterations is inefficient after a certain amount of iterations. Especially for the orthogonality, adding two sets of points is ineffective, since the measured orthogonality is already very close to zero. Therefore, the second algorithm is proposed, where only a set of points is added in a greedy iteration when the orthogonality is still high.

B.3.2. OPTIMIZED GREEDY FOR ORTHOGONAL RBF

In this optimized greedy method, a set of points is only added when the orthogonality requirement is not yet satisfied. When it is satisfied, only single points will be added

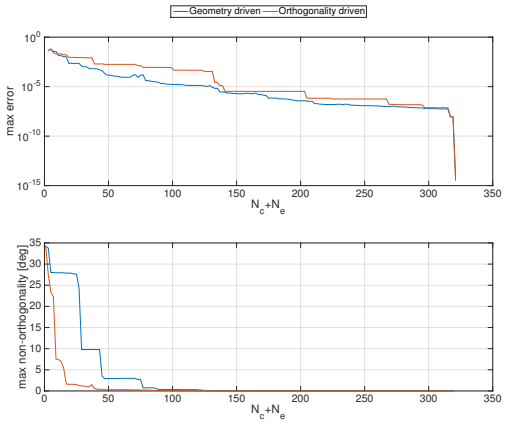

Figure B.3: Maximum surface error and maximum Non-Orthogonality on bottom surface versus number of total control points. Results are shown for both geometrical driven and orthogonal driven approach.

per iteration if the geometry requirement is not yet satisfied. In table B.1 the results (total number of control points) can be found for a variation of parameters: tolerances for geometry and orthogonality, greedy algorithm and driver (geometry or orthogonality). From the results it can be concluded that using the optimized greedy algorithm provides the best results, i.e. the lowest number of control points. For the less strict tolerance on the geometry (cases 1 to 4), there is no benefit of using the optimized greedy. However, from these cases it can already be seen that using the orthogonal driven approach reduces the number of control points used. For the stricter geometric tolerance (case 5 to 12), the optimized greedy does perform significantly better compared to the others. For the same orthogonality tolerance a decrease of used control points of 34% is found (from 62 to 40).
Compared to the original greedy (not taking into account any extra points) 50% in the number of control point is found. However, the orthogonality is reduced by a factor of 6 (from 32 to below 5). In addition, the optimized greedy algorithm nicely converges to the original greedy when the tolerance on the orthogonality is raised (case 8 to 12). To compare to the original deformed mesh in figure B.4 the deformed mesh with the optimized greedy is shown (case 8).

B.4. CONCLUSION

Orthogonality is explicitly added in the RBF mesh deformation algorithm by adding a second set of points. In addition two new greedy algorithms are proposed for limiting the total number of control points. To test the proposed methods a 2D channel test case with a 1D linearly deforming membrane is used. For the orthogonal RBF mesh deformation the orthogonality is reduced from 31.9 to 1.3 degree when all points are used. To achieve this, the number of control points has doubled. To limit the number of control points the (altered) greedy algorithm is used. Using the original greedy method and selecting two points does reduce the total number of points, but can results in a sub opti-

Table B.1: Number of control points selected for the three algorithms with different requirements. Results are given for both the surface driven or orthogonal driven method.

Case	max(ϵ_s)	θ_{tol}	Algorithm	Driver	N_c	N_e	N_c+N_e
0a	10^{-2}	-	orig. greedy	Geom.	9	0	9
1	10^{-2}	5	ortho. greedy	Geom.	17	17	34
2	10^{-2}	5	ortho. greedy	Ortho.	9	9	18
3	10^{-2}	5	opt. greedy	Geom.	17	17	34
4	10^{-2}	5	opt. greedy	Ortho.	9	9	18
0a	10^{-4}	-	orig. greedy	Geom.	27	0	27
5	10^{-4}	5	ortho. greedy	Geom.	29	29	58
6	10^{-4}	5	ortho. greedy	Ortho.	31	31	62
7	10^{-4}	5	opt. greedy	Geom.	29	23	52
8	10^{-4}	5	opt. greedy	Ortho.	31	9	40
9	10^{-4}	10	opt. greedy	Ortho.	34	5	39
10	10^{-4}	20	opt. greedy	Ortho.	34	5	39
11	10^{-4}	30	opt. greedy	Ortho.	31	2	33
12	10^{-4}	40	opt. greedy	Ortho.	27	0	27

mal set of points. Therefore the optimized greedy algorithm has been studied. Here only a pair of points is selected when the orthogonality requirement is not yet satisfied. If the orthogonality requirement is satisfied, a single point will be added to satisfy the geometric requirement (if needed). Using this optimized greedy algorithm with a orthogonality driven approach, results in the lowest number selected control points for a given set of requirements (40). Compared to the original RBF (without orthogonality addition) an increase of 50% is found (from 27 points).
These methods show promising initial results. Especially, since the method is equivalent to the original greedy if the orthogonality requirement is not strict (or the orthogonality is well preserved automatically). However, these are only initial results and further study is needed to assess the proposed method. Variations of mesh size, first cell height and deformations will be considered. In addition a (3D) engineering test case will be used to assess the robustness and efficiency of the proposed method on larger scale problems.

REFERENCES

[1] D. Maruyama, D. Bailly, and G. Carrier, *High quality mesh deformation using quaternions,* AIAA Journal **52**, 1 (2012).

[2] T. C. S. Rendall and C. B. Allen, *Parallel efficient mesh motion using radial basis functions with application to multi-bladed rotors,* International Journal for Numerical Methods in Engineering **81**, 89 (2010).

[3] T. Gillebaart, A. V. Zuijlen, and H. Bijl, *Aerodynamic analysis of the wing flexibility and the clap-and-peel motion of the hovering DelFly II,* Proceedings of the International Micro Air Vehicles conference 2011 summer edition , 92 (2011).

B

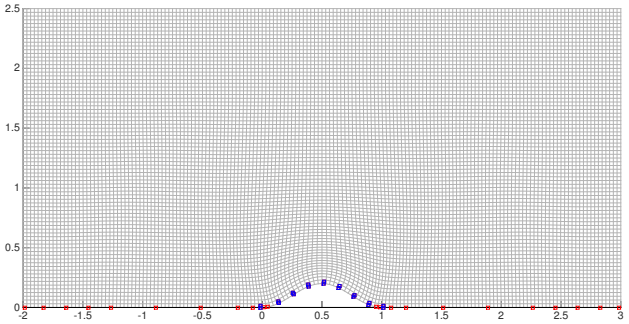

Figure B.4: Deformed mesh of 1D deforming membrane in channel case obtained with orthogonal RBF and optimized greedy. Squares indicate the used surface points and the circles the used point pairs for orthogonality.

[4] T. C. S. Rendall and C. B. Allen, *Efficient mesh motion using radial basis functions with data reduction algorithms,* Journal of Computational Physics **228**, 6231 (2009).

[5] A. de Boer, M. S. van der Schoot, and H. Bijl, *Mesh deformation based on radial basis function interpolation,* Computers and Structures **85**, 784 (2007).

[6] G. Wang, H. H. Mian, Z.-Y. Ye, and J.-D. Lee, *Improved point selection method for hybrid-unstructured mesh deformation using radial basis functions,* AIAA Journal **53**, 1016 (2015).

C

Absolute Unsteady Aerodynamics Validation Polars

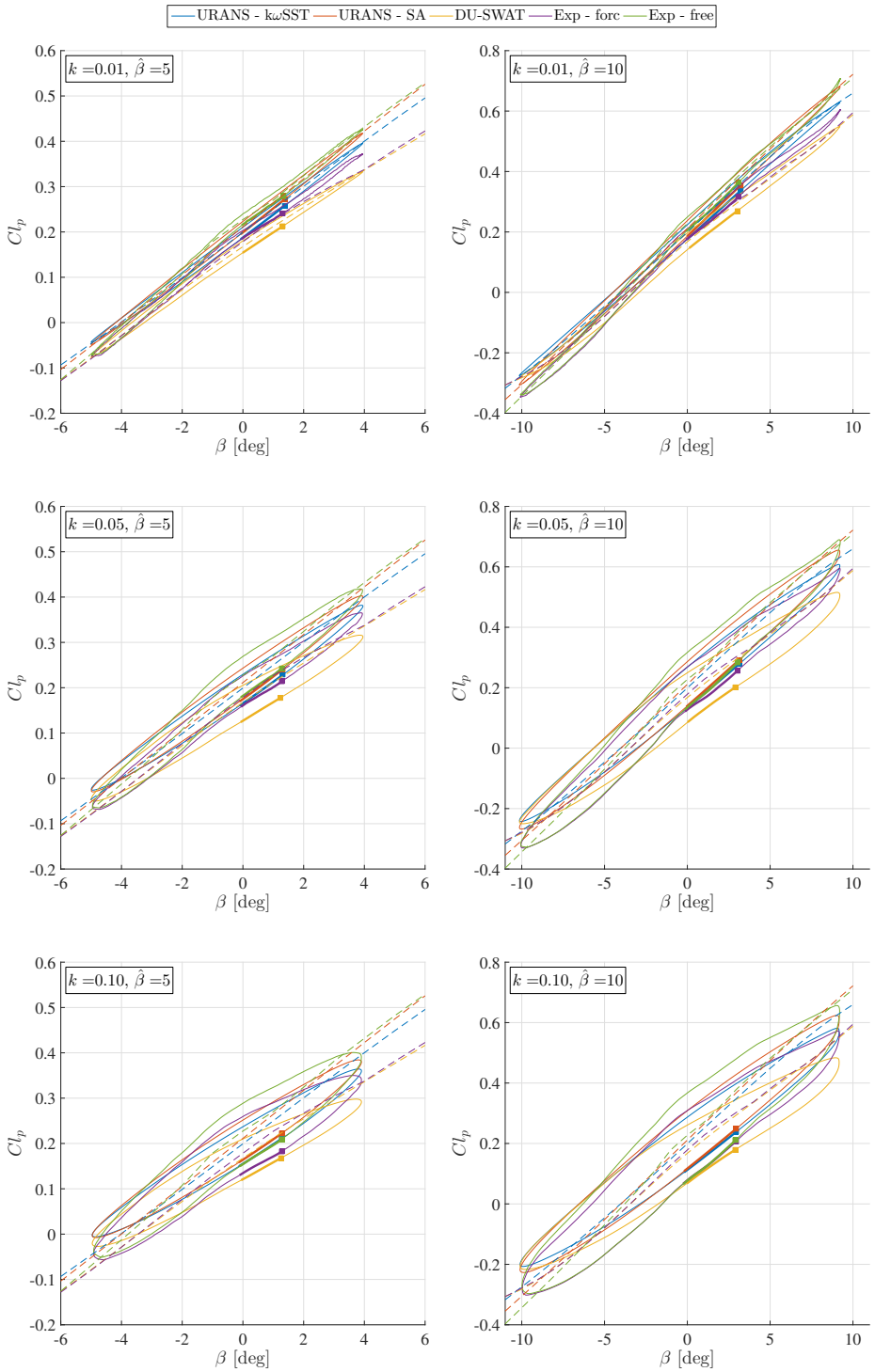

Figure C.1: Unsteady Cl_p for $\alpha = 0.0°$ at 3 different reduced frequencies and two flap amplitudes. The thicker line with square end indicates the direction of the loop, going from the beginning of the thicker line towards the square end.

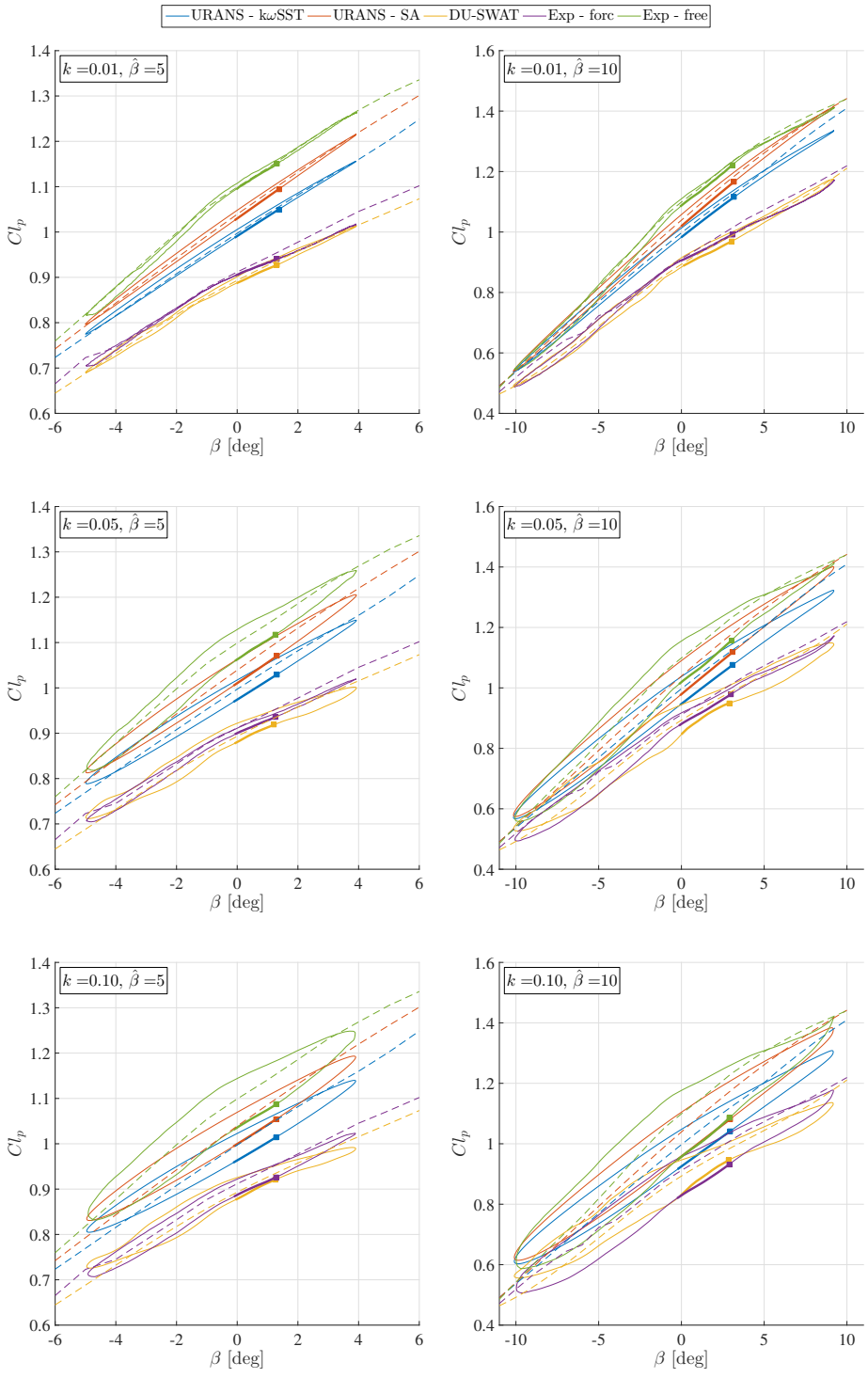

Figure C.2: Unsteady Cl_p for $\alpha = 8.0°$ at 3 different reduced frequencies and two flap amplitudes. The thicker line with square end indicates the direction of the loop, going from the beginning of the thicker line towards the square end.

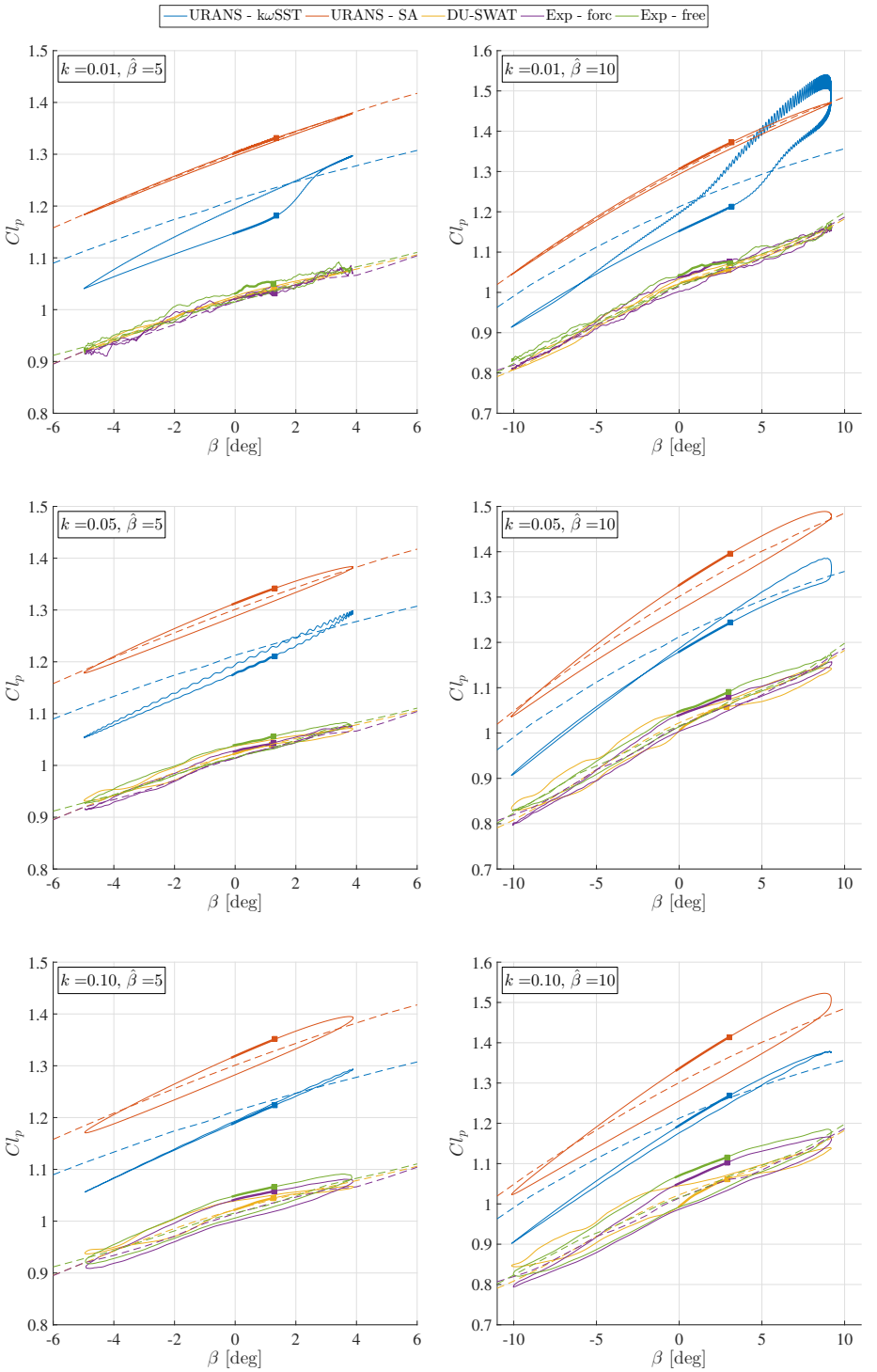

Figure C.3: Unsteady Cl_p for $\alpha = 18.0°$ at 3 different reduced frequencies and two flap amplitudes. The thicker line with square end indicates the direction of the loop, going from the beginning of the thicker line towards the square end.

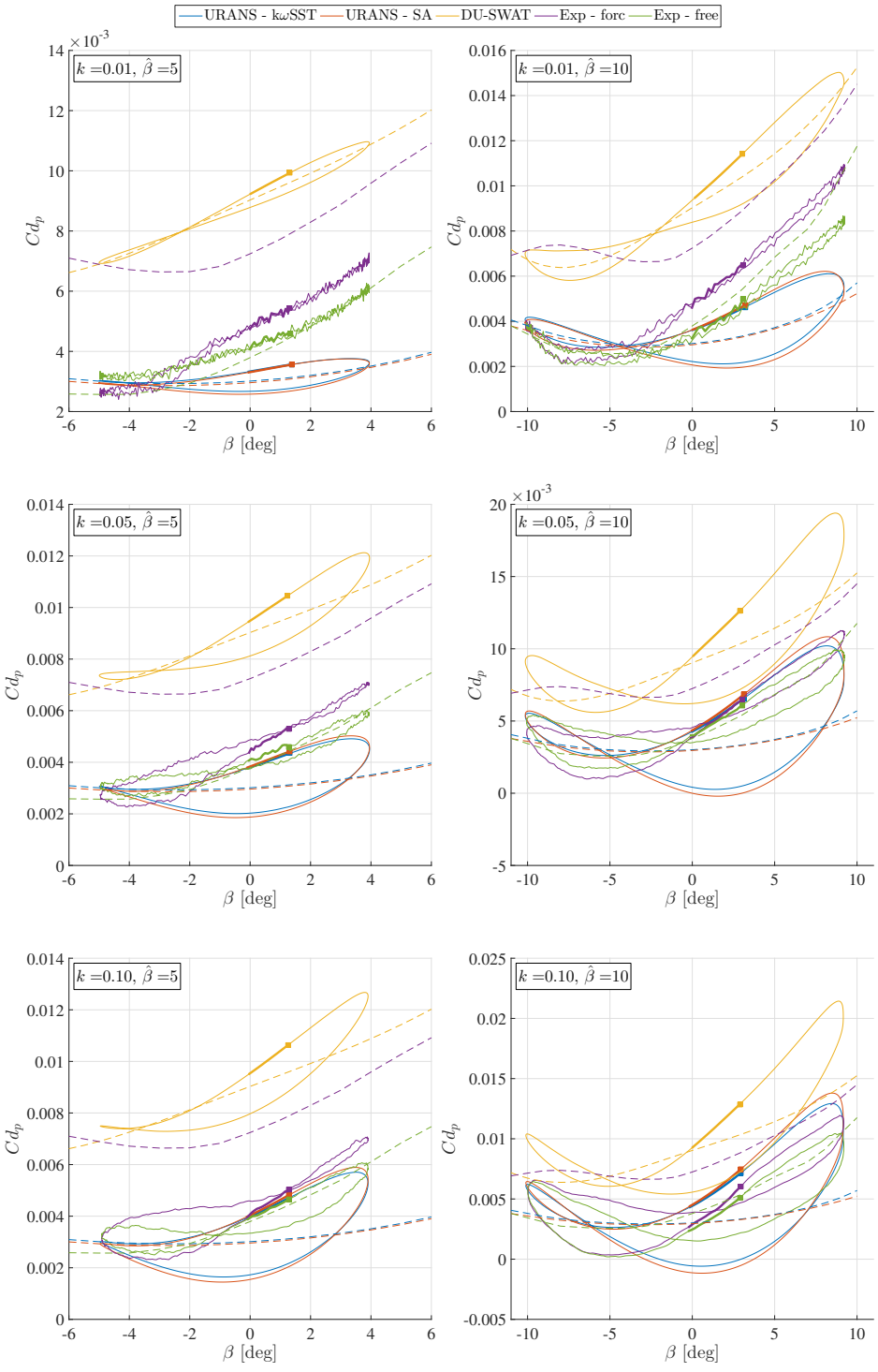

Figure C.4: Unsteady Cd_p for $\alpha = 0.0°$ at 3 different reduced frequencies and two flap amplitudes. The thicker line with square end indicates the direction of the loop, going from the beginning of the thicker line towards the square end.

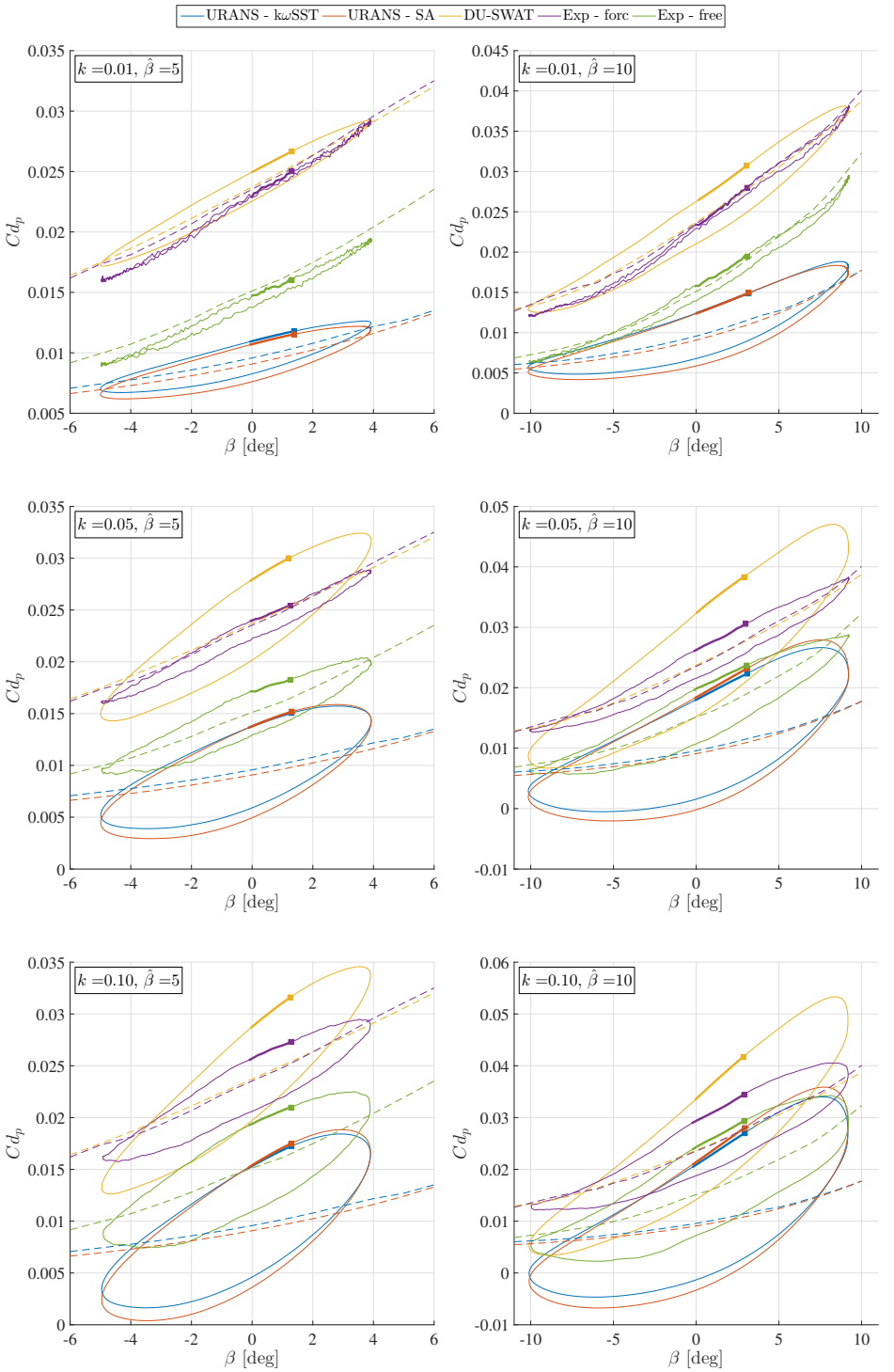

Figure C.5: Unsteady Cd_p for $\alpha = 8.0°$ at 3 different reduced frequencies and two flap amplitudes. The thicker line with square end indicates the direction of the loop, going from the beginning of the thicker line towards the square end.

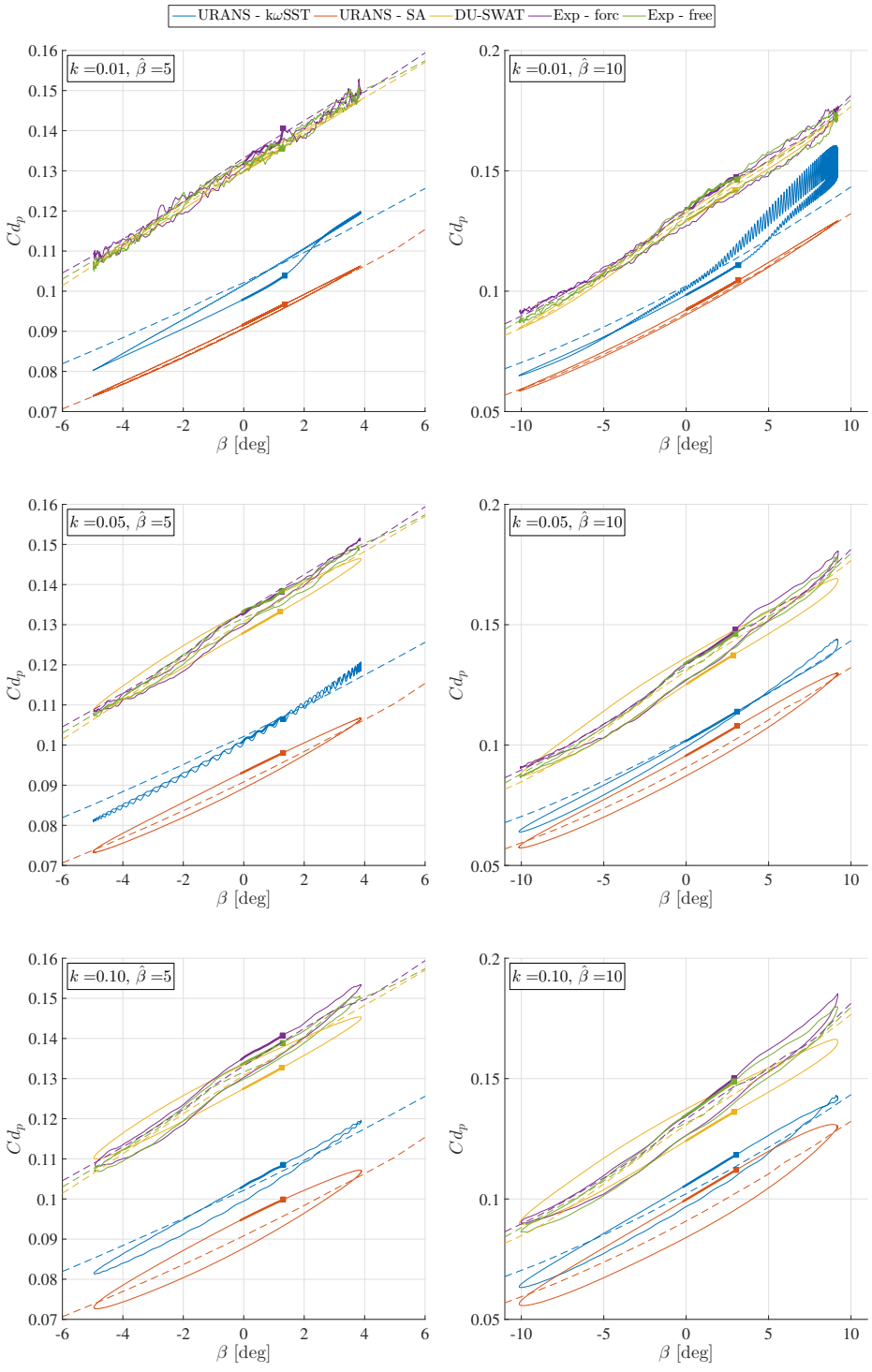

Figure C.6: Unsteady Cd_p for $\alpha = 18.0°$ at 3 different reduced frequencies and two flap amplitudes. The thicker line with square end indicates the direction of the loop, going from the beginning of the thicker line towards the square end.

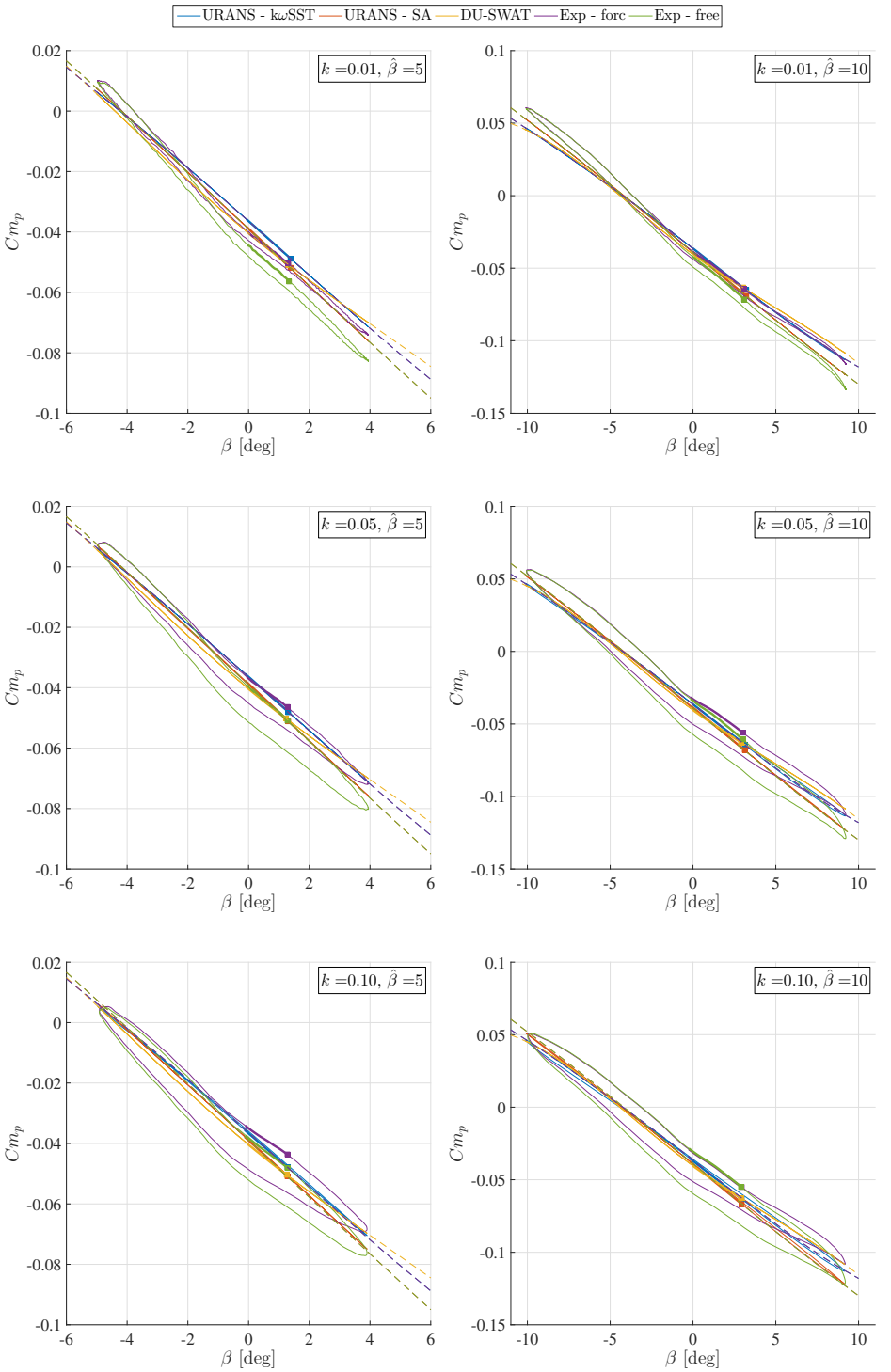

Figure C.7: Unsteady Cm_p for $\alpha = 0.0°$ at 3 different reduced frequencies and two flap amplitudes. The thicker line with square end indicates the direction of the loop, going from the beginning of the thicker line towards the square end.

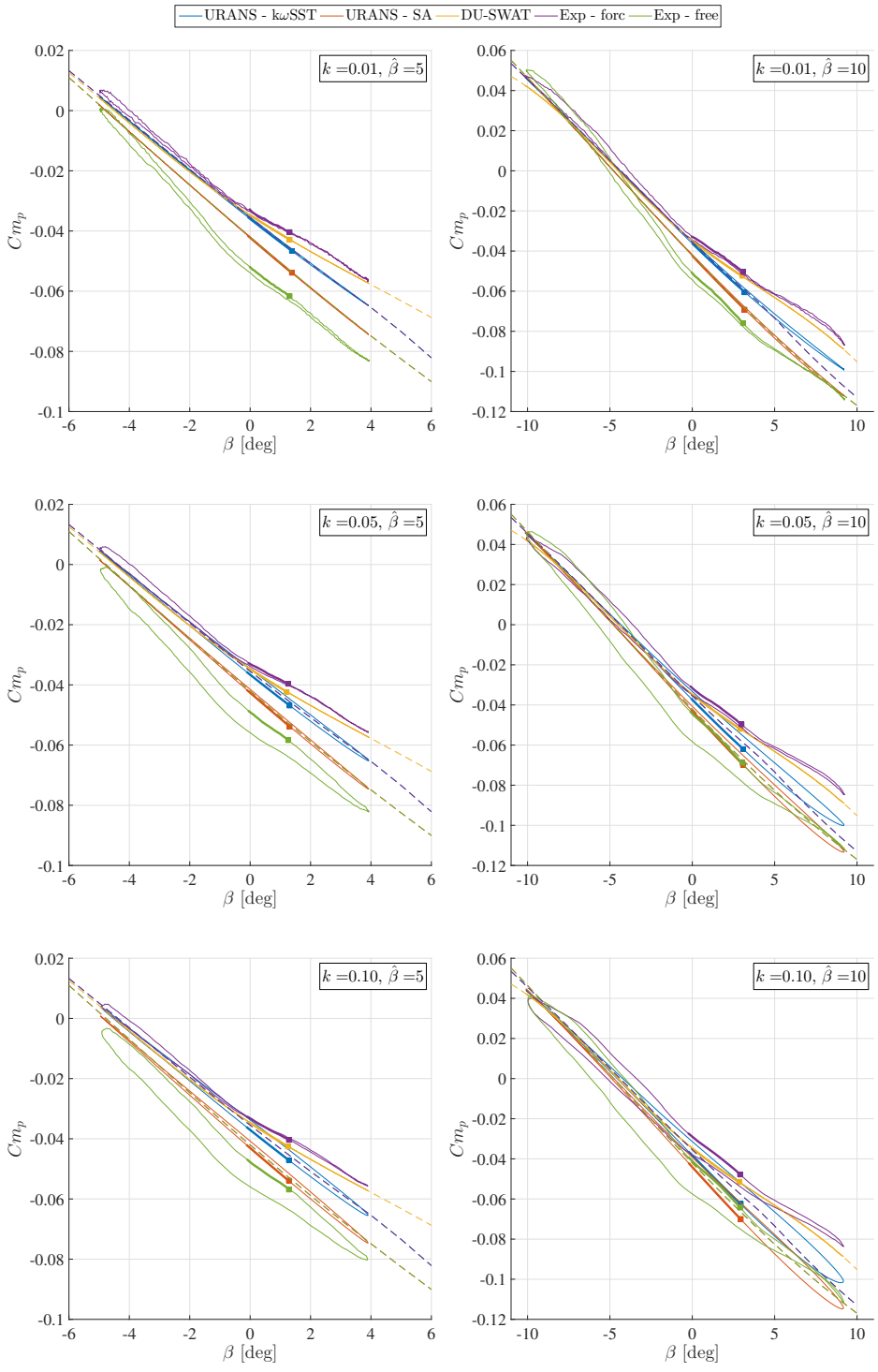

Figure C.8: Unsteady Cm_p for $\alpha = 8.0°$ at 3 different reduced frequencies and two flap amplitudes. The thicker line with square end indicates the direction of the loop, going from the beginning of the thicker line towards the square end.

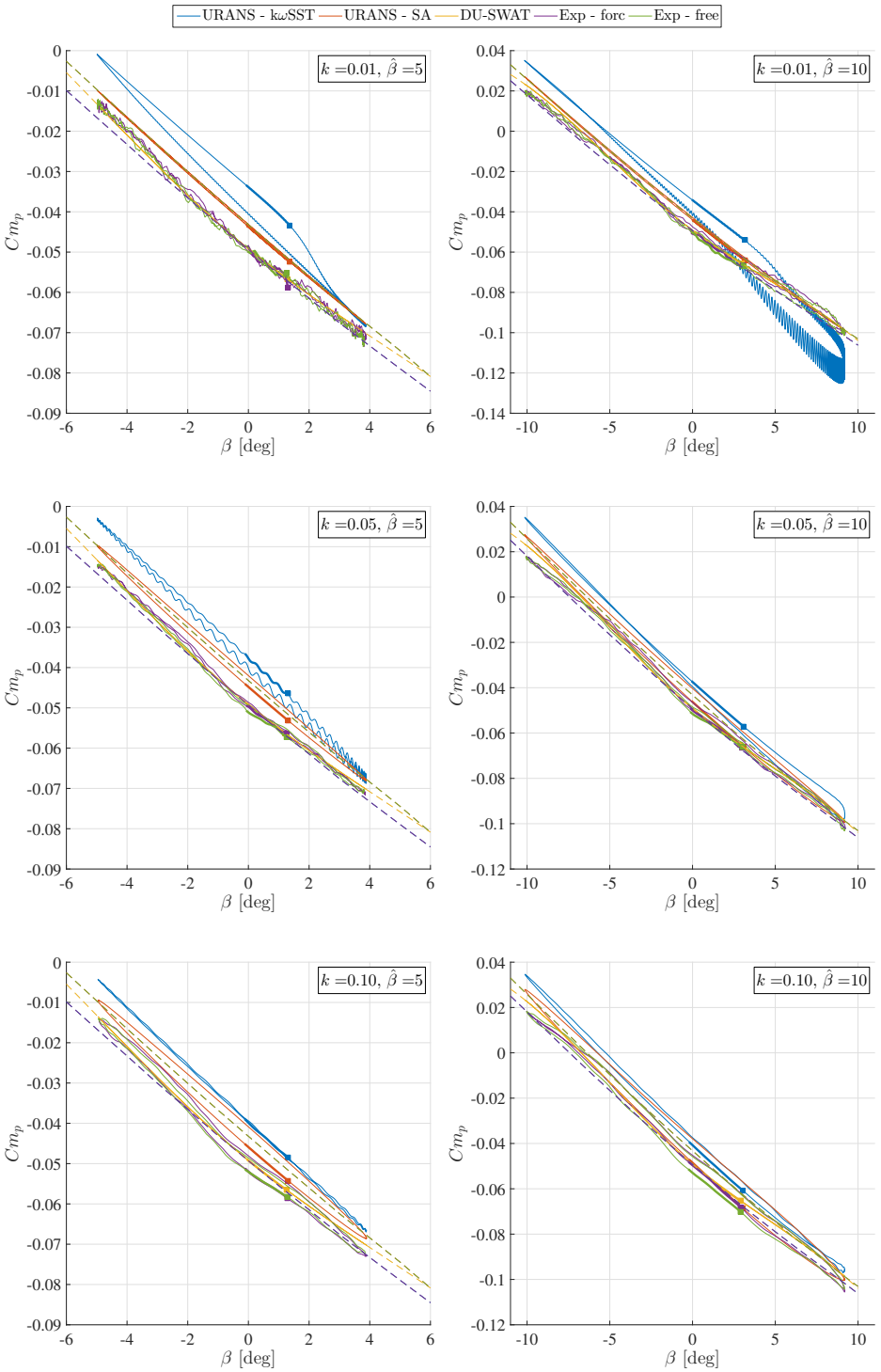

Figure C.9: Unsteady Cm_p for $\alpha = 18.0°$ at 3 different reduced frequencies and two flap amplitudes. The thicker line with square end indicates the direction of the loop, going from the beginning of the thicker line towards the square end.

D

Relative Unsteady Aerodynamics Validation Polars

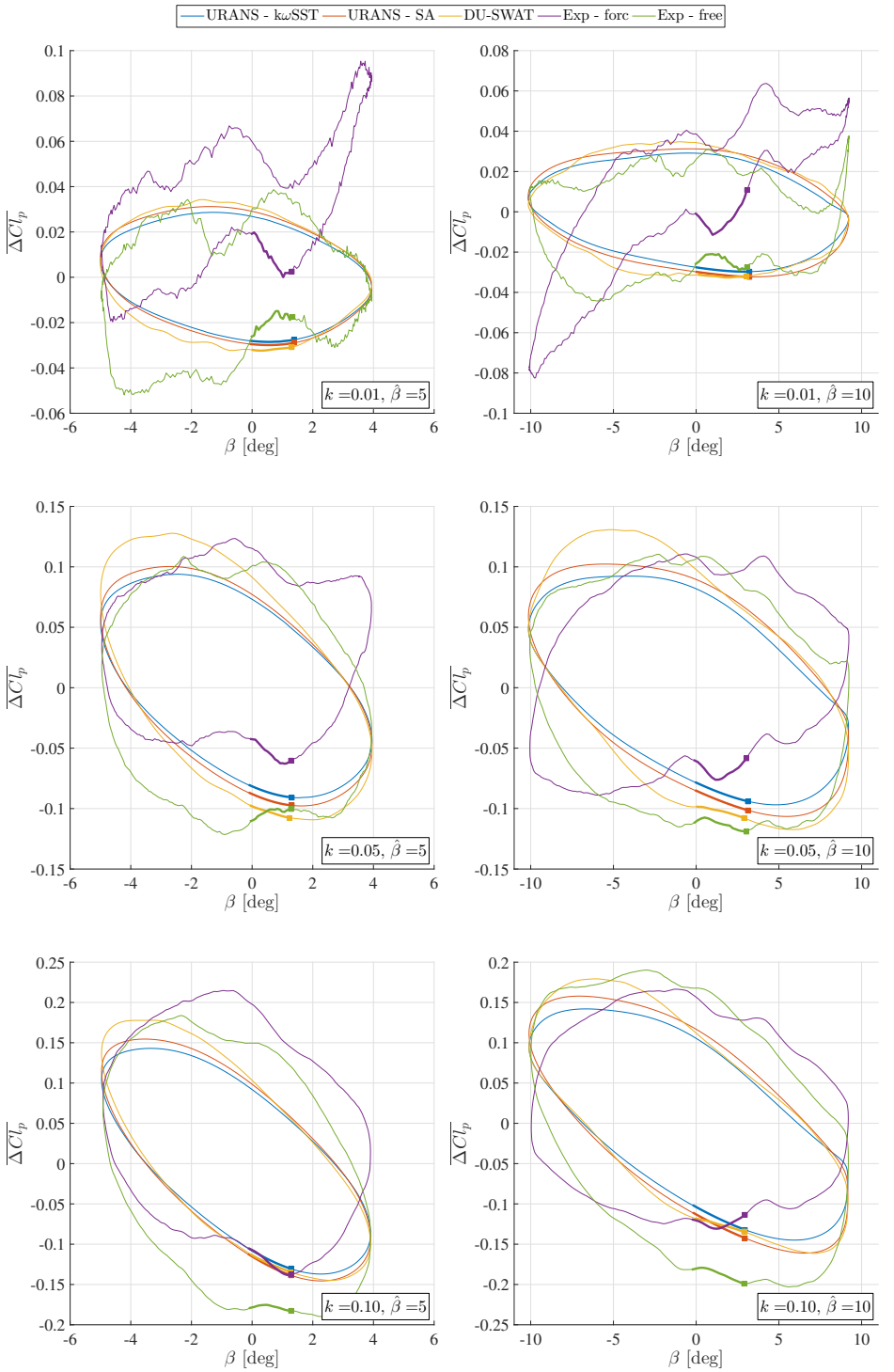

Figure D.1: Unsteady normalized $\overline{\Delta Cl_p}$ for $\alpha = 0.0°$ at 3 different reduced frequencies and two flap amplitudes. $\overline{\Delta Cl_p}$ is obtained by deducting the corresponding steady-state values at the corresponding angle of attack and flap angle for each model separately and normalizing this with the band of the experimental steady-state results. The thicker line with square end indicates the direction of the loop, going from the beginning of the thicker line towards the square end.

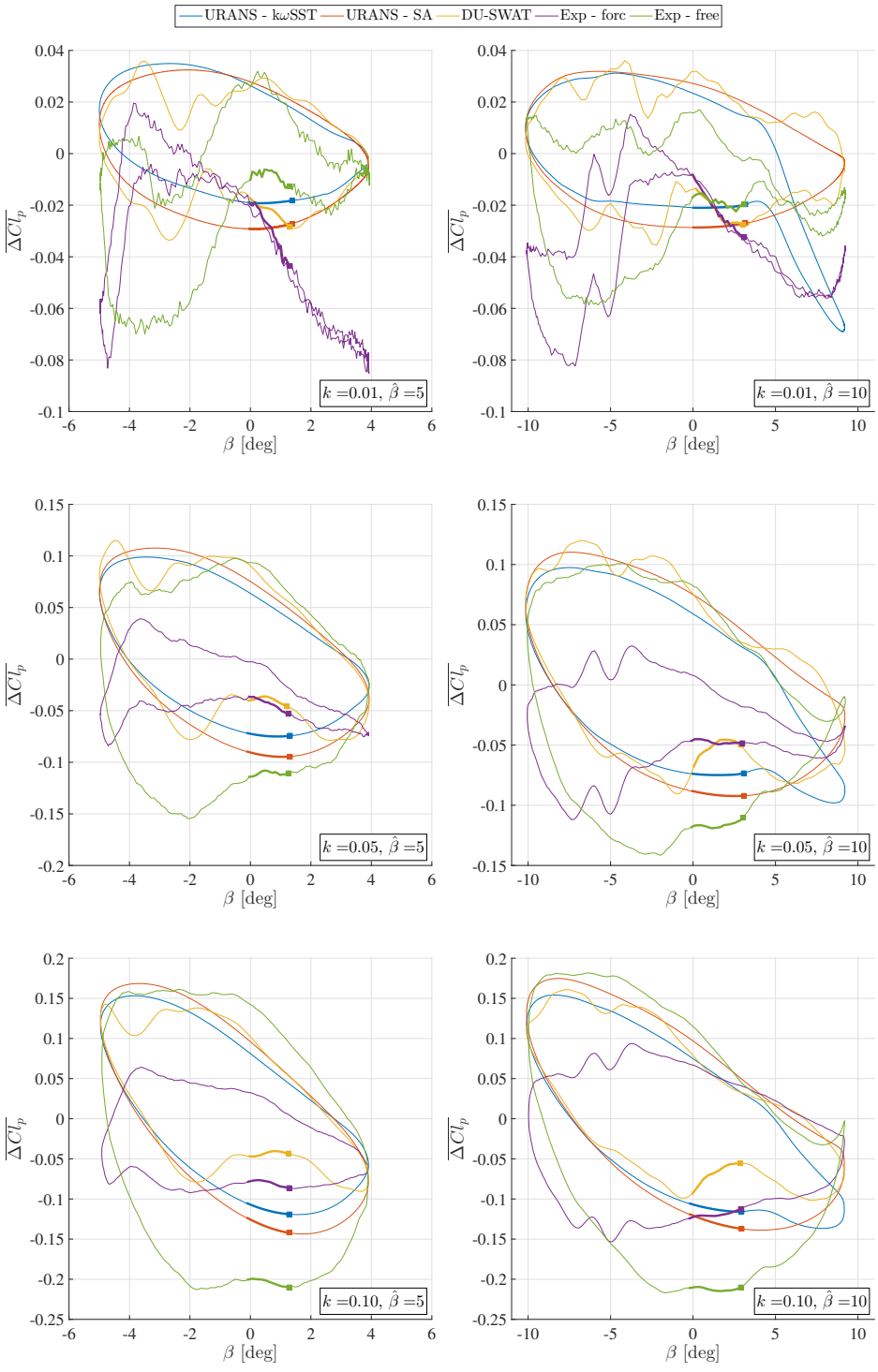

Figure D.2: Unsteady normalized $\overline{\Delta Cl_p}$ for $\alpha = 8.0°$ at 3 different reduced frequencies and two flap amplitudes. $\overline{\Delta Cl_p}$ is obtained by deducting the corresponding steady-state values at the corresponding angle of attack and flap angle for each model separately and normalizing this with the band of the experimental steady-state results. The thicker line with square end indicates the direction of the loop, going from the beginning of the thicker line towards the square end.

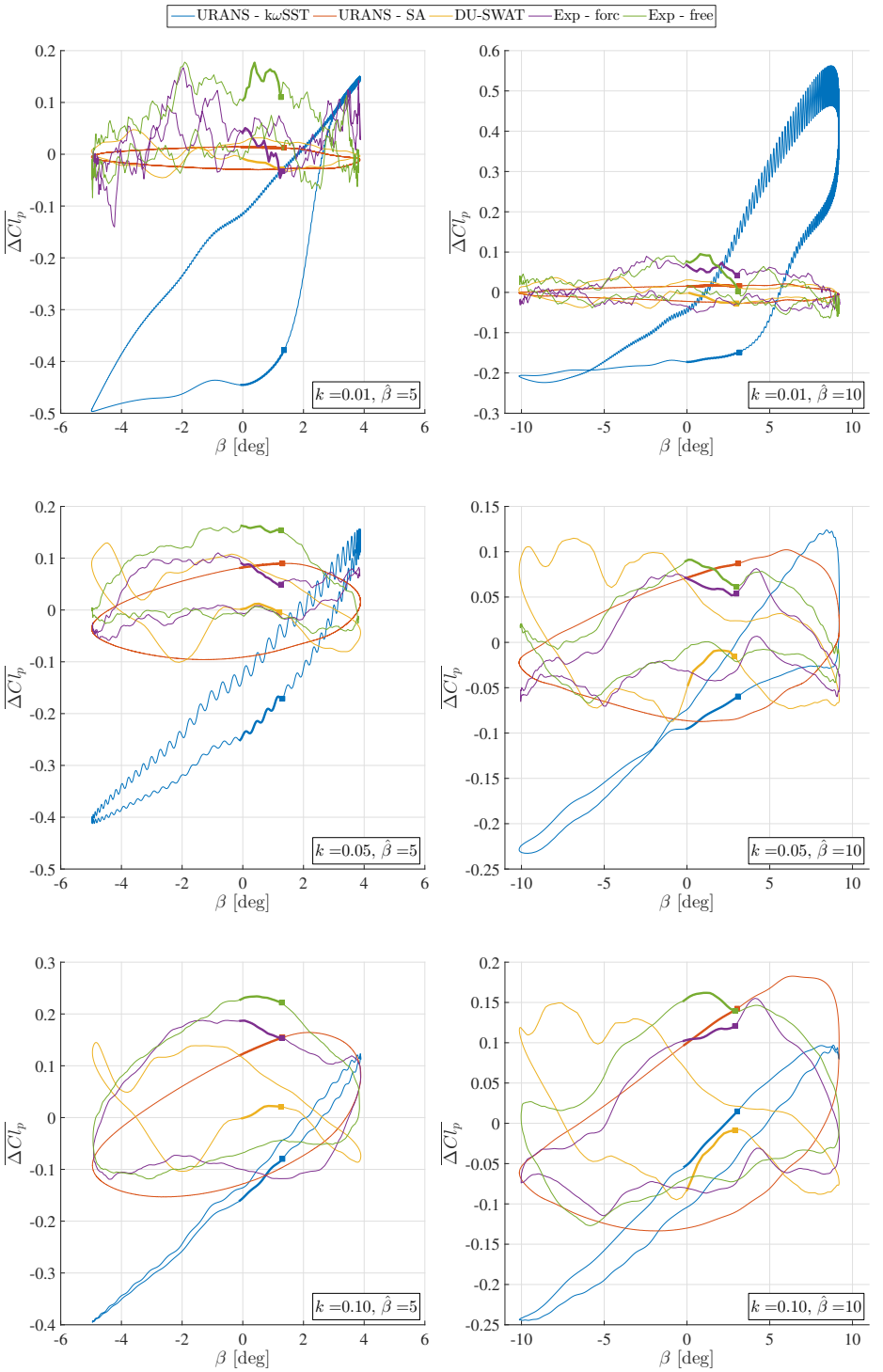

Figure D.3: Unsteady normalized $\overline{\Delta Cl_p}$ for α = 18.0° at 3 different reduced frequencies and two flap amplitudes. $\overline{\Delta Cl_p}$ is obtained by deducting the corresponding steady-state values at the corresponding angle of attack and flap angle for each model separately and normalizing this with the band of the experimental steady-state results. The thicker line with square end indicates the direction of the loop, going from the beginning of the thicker line towards the square end.

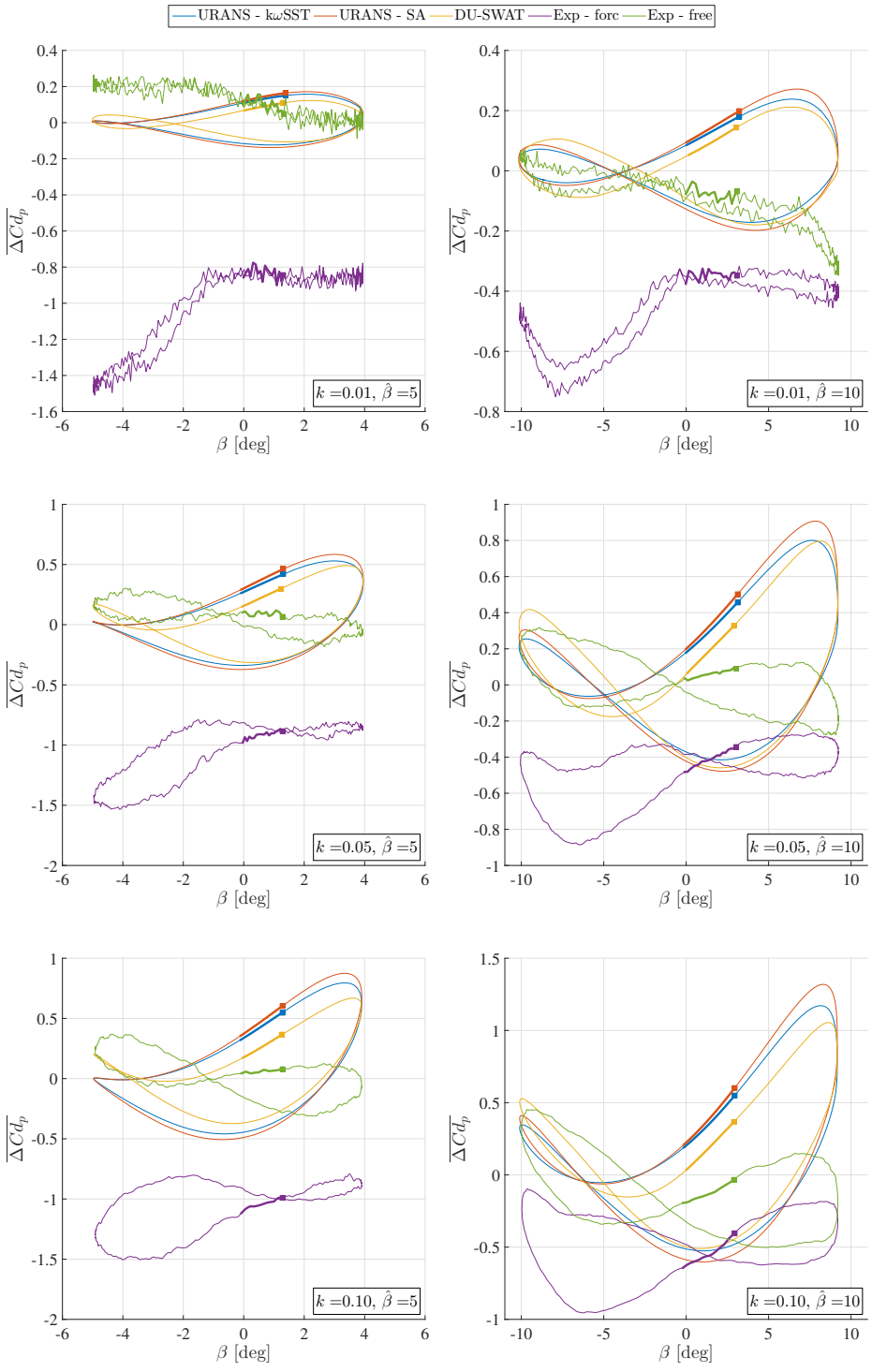

Figure D.4: Unsteady normalized $\overline{\Delta Cd_p}$ for $\alpha = 0.0°$ at 3 different reduced frequencies and two flap amplitudes. $\overline{\Delta Cd_p}$ is obtained by deducting the corresponding steady-state values at the corresponding angle of attack and flap angle for each model separately and normalizing this with the band of the experimental steady-state results. The thicker line with square end indicates the direction of the loop, going from the beginning of the thicker line towards the square end.

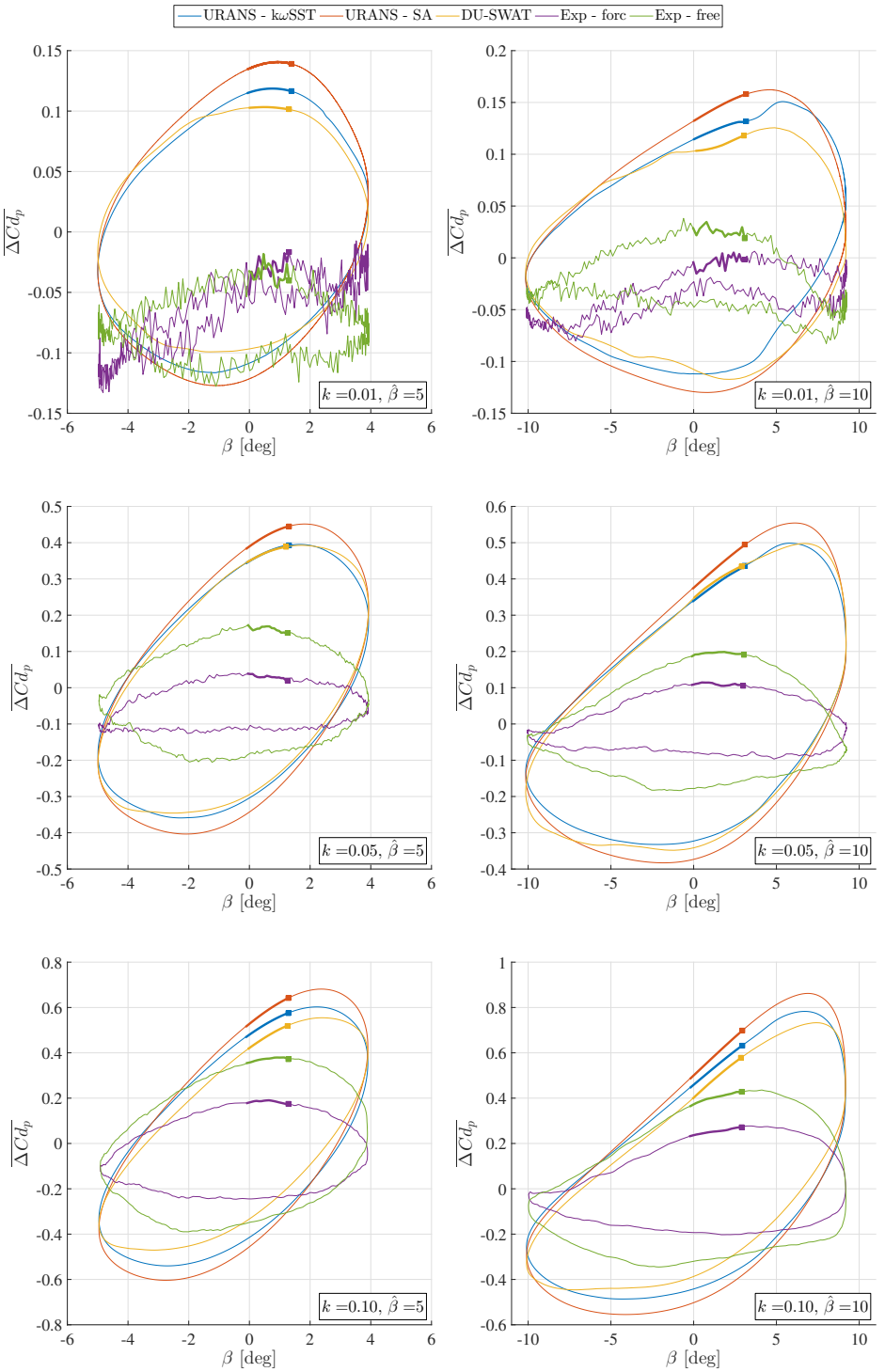

Figure D.5: Unsteady normalized $\overline{\Delta Cd_p}$ for $\alpha = 8.0°$ at 3 different reduced frequencies and two flap amplitudes. $\overline{\Delta Cd_p}$ is obtained by deducting the corresponding steady-state values at the corresponding angle of attack and flap angle for each model separately and normalizing this with the band of the experimental steady-state results. The thicker line with square end indicates the direction of the loop, going from the beginning of the thicker line towards the square end.

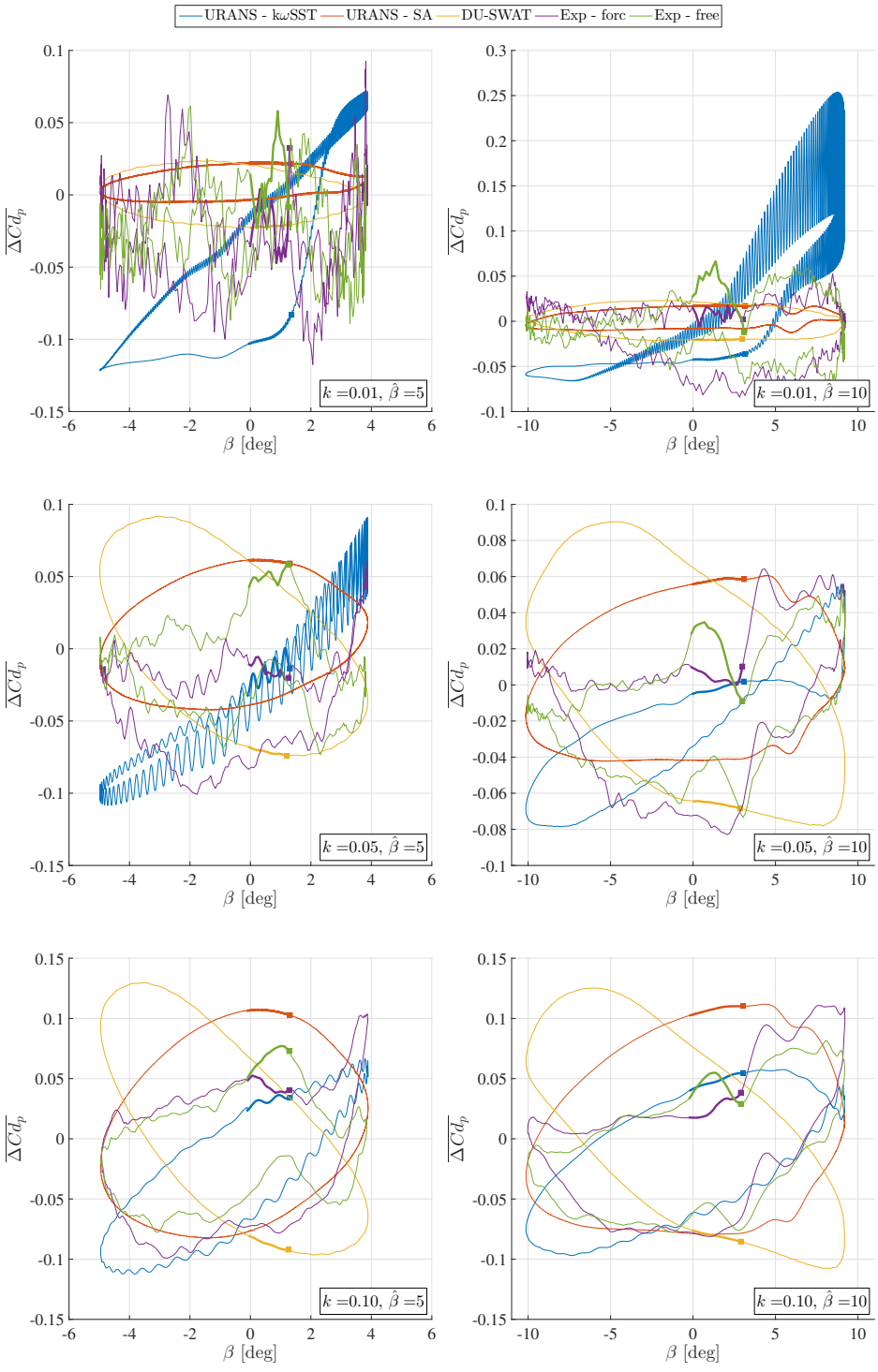

Figure D.6: Unsteady normalized $\overline{\Delta Cd_p}$ for α = 18.0° at 3 different reduced frequencies and two flap amplitudes. $\overline{\Delta Cd_p}$ is obtained by deducting the corresponding steady-state values at the corresponding angle of attack and flap angle for each model separately and normalizing this with the band of the experimental steady-state results. The thicker line with square end indicates the direction of the loop, going from the beginning of the thicker line towards the square end.

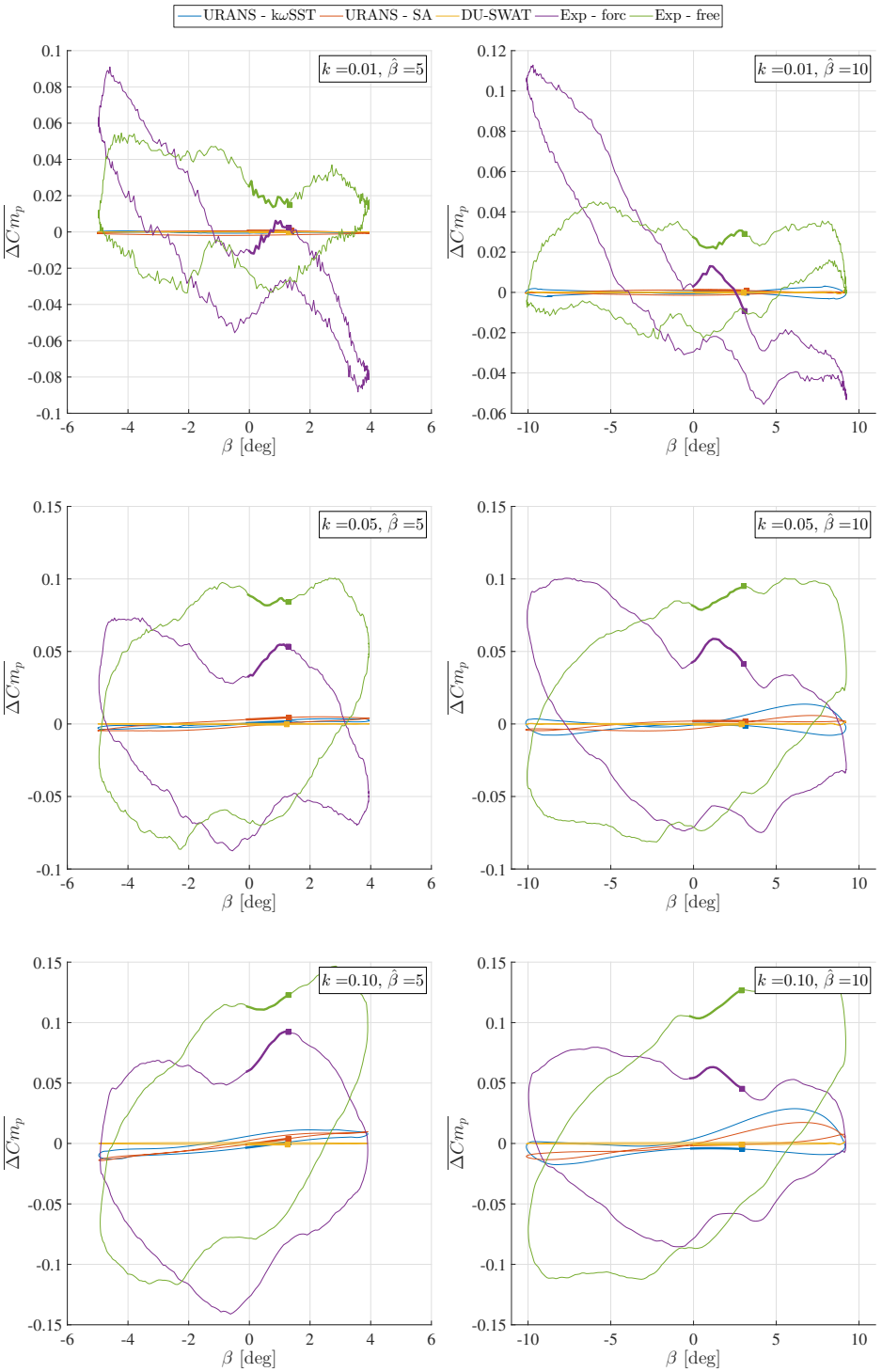

Figure D.7: Unsteady normalized $\overline{\Delta Cm_p}$ for $\alpha = 0.0°$ at 3 different reduced frequencies and two flap amplitudes. $\overline{\Delta Cm_p}$ is obtained by deducting the corresponding steady-state values at the corresponding angle of attack and flap angle for each model separately and normalizing this with the band of the experimental steady-state results. The thicker line with square end indicates the direction of the loop, going from the beginning of the thicker line towards the square end.

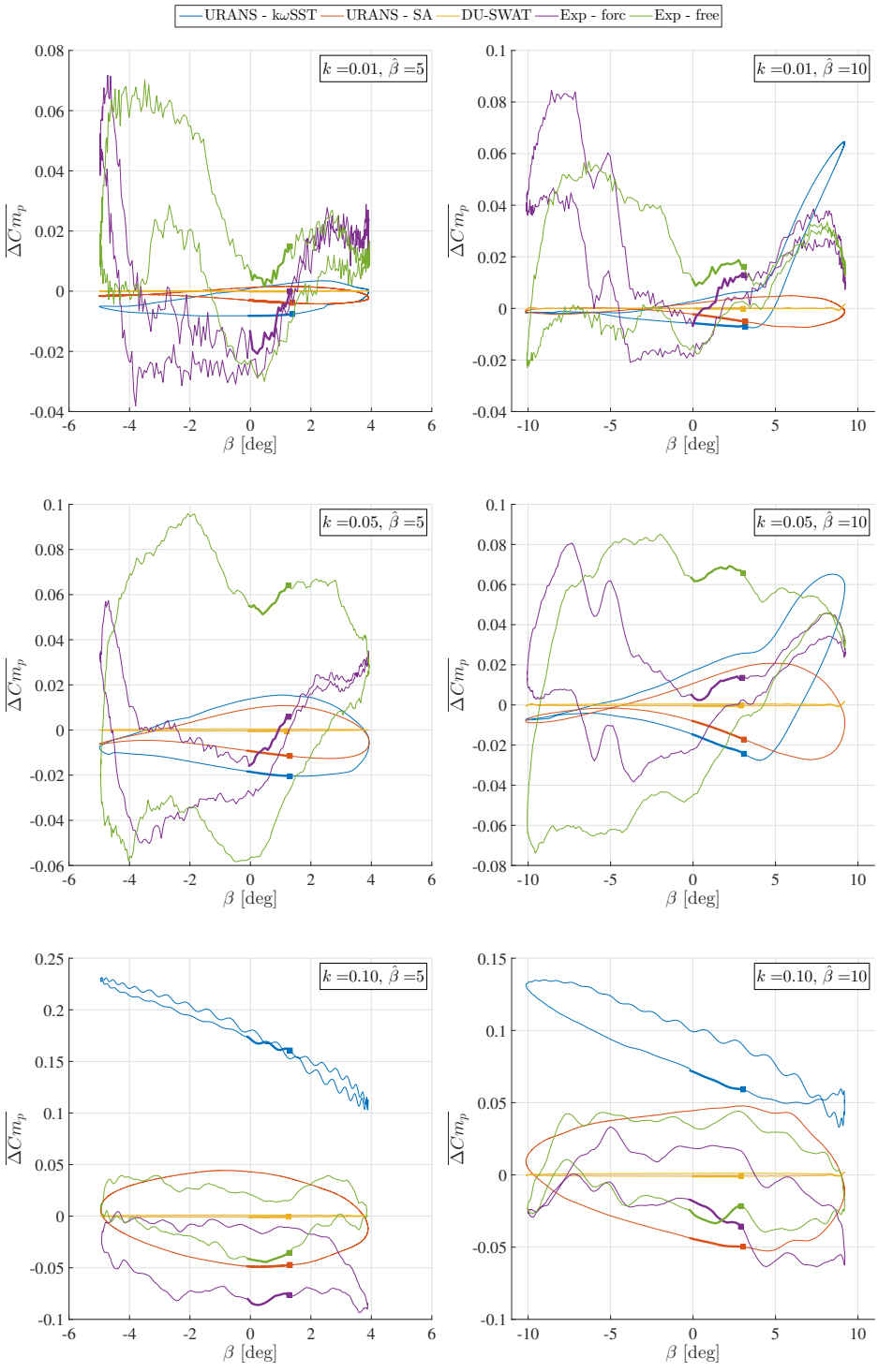

Figure D.8: Unsteady normalized $\overline{\Delta Cm_p}$ for α = 8.0° at 3 different reduced frequencies and two flap amplitudes. $\overline{\Delta Cm_p}$ is obtained by deducting the corresponding steady-state values at the corresponding angle of attack and flap angle for each model separately and normalizing this with the band of the experimental steady-state results. The thicker line with square end indicates the direction of the loop, going from the beginning of the thicker line towards the square end.

D

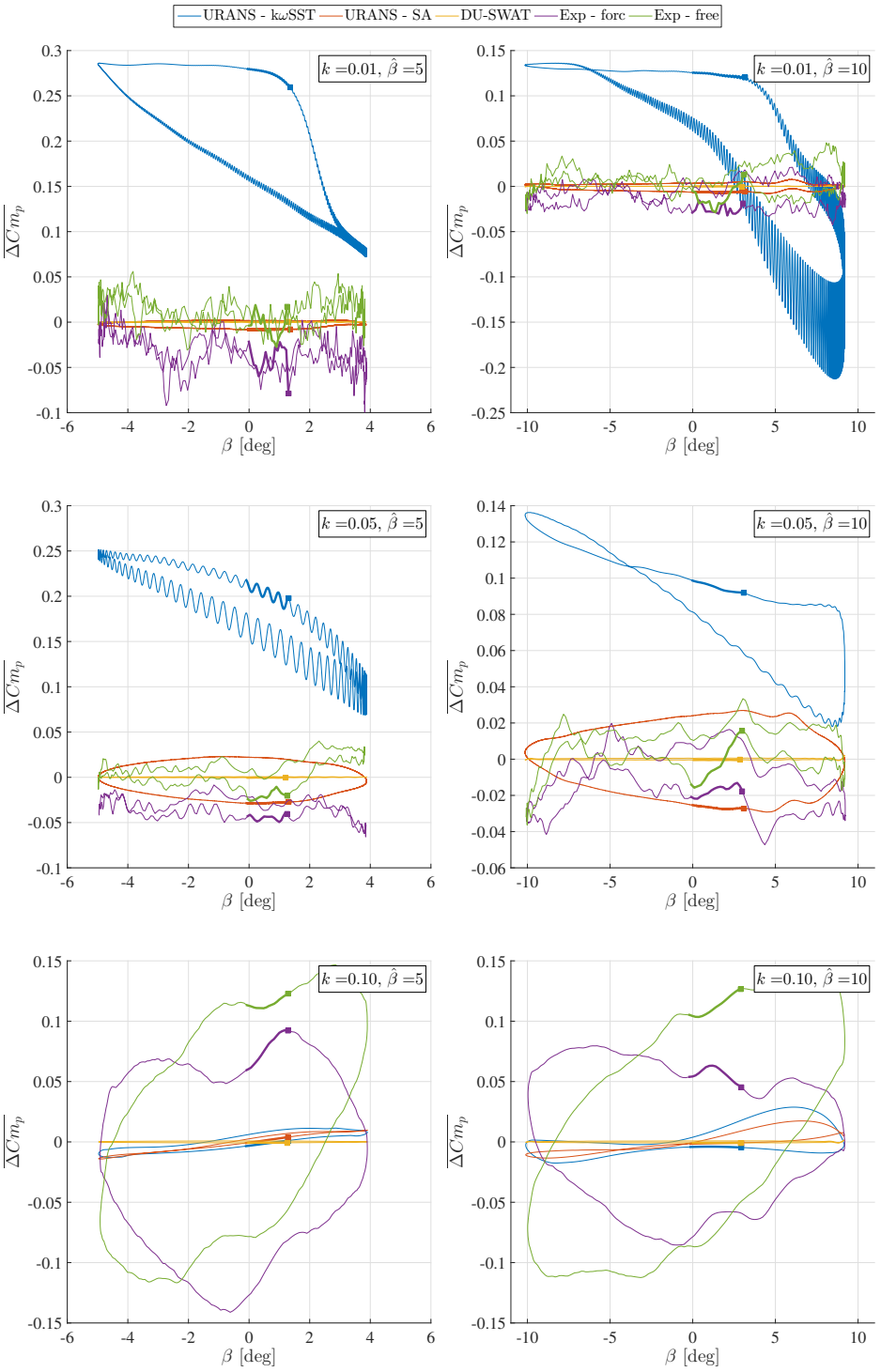

Figure D.9: Unsteady normalized $\overline{\Delta Cm_p}$ for α = 18.0° at 3 different reduced frequencies and two flap amplitudes. $\overline{\Delta Cm_p}$ is obtained by deducting the corresponding steady-state values at the corresponding angle of attack and flap angle for each model separately and normalizing this with the band of the experimental steady-state results. The thicker line with square end indicates the direction of the loop, going from the beginning of the thicker line towards the square end.

ACKNOWLEDGEMENTS

I would like to start by thanking my promotor Hester Bijl and co-promotor Alexander Herman (Sander) van Zuijlen. They started by asking me to do a Ph.D. under their supervision, for which I'm still grateful. Hester always motivated me by being positive and showing enthusiasm for the paths I took during my research. Lengthy discussions with Sander lead to new insights, both within my own research as with the guidance of master students. Doing the oral exams for the course Fluid-Structure Interaction has introduced me to level of education, which I've been a fan of ever since. Both of them taught me a lot about both research as well as having a more blunt and positive attitude.

While Hester and Sander has been a constant factor in my Ph.D., Gijs van Kuik and Jan-Willem van Wingerden have been great project supervisors, always providing a fresh look and interesting views on my topic as well as the project as a whole. In the same way Lars Bernhammer provided me with additional paths to take within my research. Additionally, I would like to thank Carlos Simão Ferreira for asking my to be part of the supervisory team of Alen Dedeic, resulting in my involvement in the Avatar project.

During the years many colleagues left and arrived, from which everyone one provided new topics to the lunch table, coffee machine discussions and department activities. However, I would like to thank a couple of them in person. First of all David, with whom I've had a great time in the past year by working together on both the time integration and mesh deformation. Together we've rebuild parts of OpenFOAM to our own liking, which was fun to do. Ye I would like to thank for both the fun moments at conferences and meetings, and the discussions we've had on several (OpenFOAM related) topics, from which I will always remember: "What if you have a cell, and you want to divide it into four cells...". Sachin, you've been helping me out with the control part of my thesis and more recently I could finally return the favor by helping you out with OpenFOAM. Liesbeth, Wouter, Rogier and Iliass I would like to thank you for providing distractions from the research, especially during lunch. Also, thanks for being patient after I had another monologue during lunch about something to do with socialism or another (random) topic, resulting in me not finishing my lunch within a normal amount of time. On this topic I would like to thank Martin, for being a partner in crime concerning the socialistic ideas.

Hamid, Bastian, Lena, Michael, Matthias and Sebastian I would like to thank you for both making the OpenFOAM workshop a success and having good times during the several conferences we've attended together. Bram Lof, Alen Dedeic, Hrvoje Dorotic and Michael Deaves thank you for being my Master students, which I've been supervising with great joy.

Outside of the University I would like to thank some long lasting friends, especially Jasper and Michael. Being part of the golden triangle has been (and still is) a great honor,

especially during Monday (and now Tuesday) evenings. The Boysov Summer I would like to thank for the great trips we've made the past years in Denmark, Norway, Sweden, Scotland, Lowlands, Ibiza, Poland, Belarus, Ukraine, Georgia and Faroer Islands. Having such a great lasting group of friends (Michael, Jasper, Dorus, Jarl, Peter, Floortje, Guus, Rick en Olivier) has been a joy during every occasion we could come up with to get together.

Pannenkoeken has been one of the most awesome new things of the past years, giving me Marleen and Roosmarie as new very close friends. Our evenings together has been a great distraction of my worries about my research. Marleen thanks for the night long conversations and Roos thanks for being my closest friend these past years and taking on adventures together.

Barend thanks for being my roommate the past years, our home was and still is a good place to live. Emmanuel and Peter, over the years the frequency of our encounters has varied significantly, but every time we meet again, it feels like old times. All my other friends I did not mention by name that came and went over the past years, we have had great adventures which I hope we will continue to have.

Finally, I would like to thank my family. Marleen, every time we meet I enjoy how easy we talk about a wide variety of topics, both on which we agree and disagree, serious and non-serious. Erik, I always like the activities we do together: football, tennis, snowboarding, biking, watching ajax, going to the movies and more recently, meeting at the coffee machine at the office. Mom and dad, thanks for always supporting me in my decisions and being there to discuss everything, welcome me home and enjoy the special moments in life.

List of Publications

Journal Publications

1. **T. Gillebaart**, W.B. Tay A.H. van Zuijlen, H. Bijl, *A modified ALE method for fluid flows around bodies moving in close proximity*, Computers and Structures, p. 1-11, **145** (2014).

2. **T. Gillebaart**, L.O. Bernhammer, A.H. van Zuijlen, G.A.M. van Kuik, *Active flap control on an aeroelastic wind turbine airfoil in gust conditions using both a CFD and an engineering model*, Journal of Physics: Conference Series, p. 1-10, **524** (2014)

3. **T. Gillebaart**, D.S. Blom, A.H. van Zuijlen, H. Bijl, *Time consistent fluid structure interaction on collocated grids for incompressible flow*, Computer Methods in Applied Mechanics and Engineering, p. 159-182, **298** (2016)

4. Y. Zhang, **T. Gillebaart**, A.H. van Zuijlen, G.J.W. Bussel ,H. Bijl, *Experimental and numerical investigations of aerodynamic loads and 3D flow over non-rotating MEXICO blades*, Wind Energy, *revision submitted*

5. **T. Gillebaart**, D.S. Blom, A.H. van Zuijlen, H. Bijl, *Adaptive Radial Basis Function mesh deformation using data reduction*, Journal of Computational Physics, *revision submitted*

6. C. J. Simão Ferreira, D. Baldacchino, D. Ragni , S. Bernardy, N. Timmer, **T. Gillebaart**, A. Dedeic, *Unsteady measurements of the DU95W180 airfoil with oscillating flap*, Wind Energy, *in preparation*

7. C. J. Simão Ferreira, A.G. Salcedo, **T. Gillebaart**, M. Reijerkerk, *Numerical model validation of oscillating trailing edge flaps for an DU95W180 airfoil*, Wind Energy, in preparation

8. D.S. Blom, V. Kazemi-Kamyab, **T. Gillebaart**, A.H. van Zuijlen, H. Bijl, *Application of spectral deferred corrections to incompressible flow on unstructured grids*, submitted

Conference Publications

1. **T. Gillebaart**, A.H. van Zuijlen, H. Bijl, *Aerodynamic analysis of the wing flexibility and the clap-and-peel motion of the hovering DelFly II*, Proceedings of the International Micro Air Vehicles Conference 2011 Summer Edition, 2011.

2. **T. Gillebaart**, A.H. van Zuijlen, H. Bijl, *Fluid-Structure Response of a Wind Turbine Airfoil to Global and Traveling Gusts*, Proceedings of 9th PhD Seminar on Wind Energy in Europe, 2013.

3. Y. Zhang, **T. Gillebaart**, G.J.W. Bussel, H. Bijl, *Validation of a transition model for the DU91-W2-250 airfoil*, Proceedings of 9th PhD Seminar on Wind Energy in Europe, 2013.

4. **T. Gillebaart**, A.H. van Zuijlen, H. Bijl, *Radial basis function mesh deformation including surface orthogonality*, Proceedings of SciTECH, 2016.

CURRICULUM VITÆ

Thijs Gillebaart was born on 17 December, 1986 in Heemskerk, The Netherlands. In Castricum he attended secondary school at the Jac. P. Thijsse College from 1999 to 2005.

He studied Aerospace Engineering at the Delft University of Technology, where in 2008 he obtained his Bachelor of Science degree. His specialization in Aerodynamics from 2008 to 2011 resulted in his Master of Science degree in Aerodynamics from the Delft University of Technology. The Master of Science program was finalized with the Master Thesis: “Influence of flexibility on the clap and peel movement of the DelFly II".

In February 2012 he started his Ph.D. titled “Towards efficient Fluid-Structure-Control Interaction for Smart Rotors", under the supervision of Hester Bijl and Alexander Herman van Zuijlen. As a Ph.D. student he focussed on development of consistent time integration for moving meshes, efficient Radial Basis Function mesh deformation and validation of numerical models for wind turbine airfoil with actively controlled trailing edge flaps. These studies resulted in the present thesis.