UNIVERSITY OF CALIFORNIA, SAN DIEGO

Fluid–Structure Interaction Analysis of Wind Turbines

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Structural Engineering

by

Ming-Chen Hsu

Committee in charge:

Professor Yuri Bazilevs, Chair
Professor Juan Carlos del Alamo
Professor David J. Benson
Professor Michael Holst
Professor Hyonny Kim
Professor Alison Lesley Marsden

2012

The Dissertation of Ming-Chen Hsu is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

_____
Chair

University of California, San Diego

2012

DEDICATION

To my wife, my cat and my parents.

EPIGRAPH

The purpose of computing is insight, not numbers.

—Richard Hamming

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SUPPLEMENTAL FILES

File 1: Supplementary QuickTime video related to Figure 3.32. NREL Phase VI wind turbine. Air speed contours at a planar cut for the 10 m/s case.

File 2: Supplementary QuickTime video related to Figure 3.32. NREL Phase VI wind turbine. Isosurfaces of air speed for the 10 m/s case.

File 3: Supplementary QuickTime video related to Figure 5.12. NREL 5 MW wind turbine FSI. Air speed contours at a planar cut for the 11.4 m/s case.

File 4: Supplementary QuickTime video related to Figure 5.11. NREL 5 MW wind turbine FSI. Isosurfaces of air speed for the 11.4 m/s case.

File 5: Supplementary QuickTime video related to Figure 5.13. NREL 5 MW wind turbine FSI. Isocontours of relative wind speed at a 30 m radial cut superposed on a moving blade.

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude and appreciation to my advisor, Prof. Yuri Bazilevs, for his wonderful guidance and inspiration throughout this research, for numerous pleasant and fruitful discussions, and for showing me interesting and important aspects of computational mechanics. This dissertation would not have been possible without his support and encouragement.

I would also like to thank my committee members, Prof. David Benson, Prof. Michael Holst, Prof. Hyonny Kim, Prof. Alison Marsden and Prof. Juan Carlos del Alamo, for the contribution of their time, experience, and guidance in my research efforts.

I am indebted to Prof. Tom Hughes, Prof. Tayfun Tezduyar, Prof. Kenji Takizawa, Prof. Yongjie (Jessica) Zhang, Prof. Kai-Uwe Bletzinger and Dr. Roland Wüchner for helpful suggestions and fruitful discussions on my research works.

I am grateful to my colleagues and friends, Dr. Ido Akkerman, Dr. Josef Kiendl, Dr. Emmanuel De Luycker, Dr. Wenyan Wang, Dr. Peter Stein, Chris Long, Juan Murcia-Delso, Kyungtae Kim, Hung-Wei Tseng, Artem Korobenko, Kenny Benner, Joe Chou, Amber Chu, Wayne Liu, Amy Lee, Momo Hsiao, Keng-Ming Chang, Wei-Cheng Lu, Shih-Fan Jang and Cliff Tsai, for their support, inspiration and friendship over the past several years.

Finally, words are not enough to express my gratitude to my wife, Ming-Huei, for her love, understanding and encouragement, to my cat, Mao, for accompanying me on my journey, and to my dad, Ting-Chi, my mom, Pi-Hsia, and my sister, Wei-Hsi, for their support and patience. Without my family, life would not be so rich and colorful. Their unconditional support made this dissertation possible.

Chapter 2, in part, is a reprint of the material as it appears in: "ALE–VMS and ST–VMS methods for computer modeling of wind-turbine rotor aerodynamics and fluid–structure interaction," (with Y. Bazilevs, K. Takizawa and T.E. Tezduyar), *Mathematical Models and Methods in Applied Sciences*, 2012. The dissertation author was the primary investigator and author of this paper.

Chapter 3, in part, is a reprint of the material as it appears in: "Wind turbine aerodynamics using ALE–VMS: Validation and the role of weakly enforced boundary con-

ditions," (with I. Akkerman and Y. Bazilevs), *Computational Mechanics*, 2012; "ALE–VMS and ST–VMS methods for computer modeling of wind-turbine rotor aerodynamics and fluid–structure interaction," (with Y. Bazilevs, K. Takizawa and T.E. Tezduyar), *Mathematical Models and Methods in Applied Sciences*, 2012; "High-performance computing of wind turbine aerodynamics using isogeometric analysis," (with I. Akkerman and Y. Bazilevs), *Computers & Fluids*, 2011. This chapter, in part, has also been submitted for publication of the material as it may appear in: "Finite element simulation of wind turbine aerodynamics: Validation study using NREL Phase VI experiment," (with I. Akkerman and Y. Bazilevs), *Wind Energy*, 2012. The dissertation author was the primary investigator and author of these papers.

Chapter 4, in part, is a reprint of the material as it appears in: "ALE–VMS and ST–VMS methods for computer modeling of wind-turbine rotor aerodynamics and fluid–structure interaction," (with Y. Bazilevs, K. Takizawa and T.E. Tezduyar), *Mathematical Models and Methods in Applied Sciences*, 2012; "A computational procedure for prebending of wind turbine blades," (Y. Bazilevs, J. Kiendl and D.J. Benson), *International Journal for Numerical Methods in Engineering*, 2012; "3D simulation of wind turbine rotors at full scale. Part II: Fluid–structure interaction modeling with composite blades," (Y. Bazilevs, J. Kiendl, R. Wüchner and K.-U. Bletzinger), *International Journal for Numerical Methods in Fluids*, 2011; "The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches," (with J. Kiendl, Y. Bazilevs, K.-U. Bletzinger and R. Wüchner), *Computer Methods in Applied Mechanics and Engineering*, 2010. The dissertation author was the primary investigator and author of these papers.

Chapter 5, in part, is a reprint of the material as it appears in: "ALE–VMS and ST–VMS methods for computer modeling of wind-turbine rotor aerodynamics and fluid–structure interaction," (Y. Bazilevs, K. Takizawa and T.E. Tezduyar), *Mathematical Models and Methods in Applied Sciences*, 2012; "Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines," (Y. Bazilevs and M.A. Scott), *Computer Methods in Applied Mechanics and Engineering*, 2012; "3D simulation of wind turbine rotors at full scale. Part II: Fluid–structure interaction modeling with composite blades," (Y. Bazilevs, J. Kiendl, R.

Wüchner and K.-U. Bletzinger), *International Journal for Numerical Methods in Fluids*, 2011. The dissertation author was the primary investigator and author of these papers.

VITA

2003	Bachelor of Science, National Taiwan University
2005	Master of Science, National Taiwan University
2008	Master of Science in Engineering, The University of Texas at Austin
2012	Doctor of Philosophy, University of California, San Diego

PUBLICATIONS

"ALE–VMS and ST–VMS methods for computer modeling of wind-turbine rotor aerodynamics and fluid–structure interaction," (with Y. Bazilevs, K. Takizawa and T.E. Tezduyar), *Mathematical Models and Methods in Applied Sciences*, 2012. doi:10.1142/S0218202512300025

"Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines," (with Y. Bazilevs and M.A. Scott), *Computer Methods in Applied Mechanics and Engineering*, 2012. doi:10.1016/j.cma.2012.03.028

"Wind turbine aerodynamics using ALE–VMS: Validation and the role of weakly enforced boundary conditions," (with I. Akkerman and Y. Bazilevs), *Computational Mechanics*, 2012. doi:10.1007/s00466-012-0686-x

"Fluid–structure interaction simulations of the Fontan procedure using variable wall properties," (with C.C. Long, Y. Bazilevs, J.A. Feinstein and A.L. Marsden), *International Journal for Numerical Methods in Biomedical Engineering*, Vol. 28, pp. 512–527, 2012.

"Operator- and template-based modeling of solid geometry for isogeometric analysis with application to vertical axis wind turbine simulation," (with P. Stein, Y. Bazilevs and K. Beucke), *Computer Methods in Applied Mechanics and Engineering*, Vol. 213-216, pp. 71–83, 2012.

"A computational procedure for prebending of wind turbine blades," (with Y. Bazilevs, J. Kiendl and D.J. Benson), *International Journal for Numerical Methods in Engineering*, Vol. 89, pp. 323–336, 2012.

"Numerical-performance studies for the stabilized space-time computation of wind-turbine rotor aerodynamics," (with K. Takizawa, B. Henicke, D. Montes, T.E. Tezduyar and Y. Bazilevs), *Computational Mechanics*, Vol. 48, pp. 647–657, 2011.

"Stabilized space-time computation of wind-turbine rotor aerodynamics," (with K. Takizawa, B. Henicke, T.E. Tezduyar and Y. Bazilevs), *Computational Mechanics*, Vol. 48, pp. 333–344, 2011.

"X-FEM in isogeometric analysis for linear fracture mechanics," (with E. De Luycker, D.J. Benson, T. Belytschko and Y. Bazilevs), *International Journal for Numerical Methods in Engineering*, Vol. 87, pp. 541–565, 2011.

"High-performance computing of wind turbine aerodynamics using isogeometric analysis," (with I. Akkerman and Y. Bazilevs), *Computers & Fluids*, Vol. 49, pp. 93–100, 2011.

"3D simulation of wind turbine rotors at full scale. Part II: Fluid–structure interaction modeling with composite blades," (Y. Bazilevs, J. Kiendl, R. Wüchner and K.-U. Bletzinger), *International Journal for Numerical Methods in Fluids*, Vol. 65, pp. 236–253, 2011.

"3D simulation of wind turbine rotors at full scale. Part I: Geometry modeling and aerodynamics," (with Y. Bazilevs, I. Akkerman, S. Wright, K. Takizawa, B. Henicke, T. Spielman and T.E. Tezduyar), *International Journal for Numerical Methods in Fluids*, Vol. 65, pp. 207–235, 2011.

"Blood vessel tissue prestress modeling for vascular fluid–structure interaction simulation," (with Y. Bazilevs), *Finite Elements in Analysis and Design*, Vol. 47, pp. 93–599, 2011.

"A large deformation, rotation-free, isogeometric shell," (D.J. Benson, Y. Bazilevs and T.J.R. Hughes), *Computer Methods in Applied Mechanics and Engineering*, Vol. 200, pp. 1367–1378, 2011.

"The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches," (with J. Kiendl, Y. Bazilevs, K.-U. Bletzinger and R. Wüchner), *Computer Methods in Applied Mechanics and Engineering*, Vol. 199, pp. 2403–2416, 2010.

"A generalized finite element formulation for arbitrary basis functions: From isogeometric analysis to XFEM," (D.J. Benson, Y. Bazilevs, E. De Luycker, M. Scott, T.J.R. Hughes and T. Belytschko), *International Journal for Numerical Methods in Engineering*, Vol. 83, pp. 765–785, 2010.

"Computational vascular fluid–structure interaction: Methodology and application to cerebral aneurysms," (with Y. Bazilevs, Y. Zhang, W. Wang, T. Kvamsdal, S. Hentschel and J.G. Isaksen) *Biomechanics and Modeling in Mechanobiology*, Vol. 9, pp. 481–498, 2010.

"A fully-coupled fluid–structure interaction simulation of cerebral aneurysms," (with Y. Bazilevs, Y. Zhang, W. Wang, X. Liang, T. Kvamsdal, R. Brekken and J.G. Isaksen), *Computational Mechanics*, Vol. 46, pp. 3–16, 2010.

"Isogeometric shell analysis: The Reissner–Mindlin shell," (with D.J. Benson, Y. Bazilevs and T.J.R. Hughes), *Computer Methods in Applied Mechanics and Engineering*, Vol. 199, pp. 276–289, 2010.

"Improving stability of multiscale formulations in flow simulations at small time steps," (with Y. Bazilevs, V.M. Calo, T.E. Tezduyar and T.J.R. Hughes), *Computer Methods in Applied Mechanics and Engineering*, Vol. 199, pp. 828–840, 2010.

"Computational fluid–structure interaction: Methods and application to a total cavopulmonary connection," (Y. Bazilevs, D.J. Benson, S. Sankaran and A.L. Marsden), *Computational Mechanics*, Vol. 45, pp. 77–89, 2009.

"High-fidelity tetrahedral mesh generation from medical imaging data for fluid–structure interaction analysis of cerebral aneurysms," (with Y. Zhang, W. Wang, X. Liang, Y. Bazilevs, T. Kvamsdal, R. Brekken and J.G. Isaksen), *Computer Modeling in Engineering & Sciences*, Vol. 42, pp. 131–150, 2009.

"Finite-element simulation of incompressible viscous flows in moving meshes," (with T.W.H. Sheu), *Numerical Heat Transfer, Part B: Fundamentals*, Vol. 56, pp. 38–57, 2009.

"Finite element simulation of wind turbine aerodynamics: Validation study using NREL Phase VI experiment," (with I. Akkerman and Y. Bazilevs), *Wind Energy*, in review, 2012.

ABSTRACT OF THE DISSERTATION

Fluid–Structure Interaction Analysis of Wind Turbines

by

Ming-Chen Hsu

Doctor of Philosophy in Structural Engineering

University of California, San Diego, 2012

Professor Yuri Bazilevs, Chair

Countries around the world are putting substantial effort into the development of wind energy technologies. The urgent need of renewable energy puts pressure on the wind energy industry research and development to enhance the current wind generation capabilities and decrease the associated costs. Currently most wind turbine aerodynamics and aeroelasticity simulations are performed using low-fidelity methods. These methods are simple to implement and fast to execute; however, the cases involving important features, such as unsteady flow, turbulence, and details of the wind turbine geometry, are beyond their range of applicability. In this dissertation, we introduce a paradigm shift in wind turbine analysis by developing 3D, complex geometry, time-dependent, multi-physics modeling procedures for wind turbine fluid–structure interaction (FSI).

The proposed framework consists of a collection of numerical methods combined into a single framework for FSI modeling and simulation of wind turbines at full scale. The use of the Navier–Stokes equations of incompressible flows for wind turbine aerodynamics is validated against experimental data. The structural modeling of the composite blades is based on the Kirchhoff–Love thin shell theory discretized using isogeometric analysis. The coupled FSI formulation is derived using the augmented Lagrangian approach and accommodates non-matching fluid–structure interface discretizations. The challenges of fluid–structural coupling and the handling of computational domains in relative motion are discussed, and the FSI computations of a 5 MW offshore baseline wind turbine are shown.

Chapter 1

Introduction

Countries around the world are putting substantial effort into the development of wind energy technologies. Currently in the EU, 50 GW of electricity comes from the on-land and 1 GW from offshore wind turbines. The EU target is to raise the on-land production to 130 GW and offshore to 50 GW by 2020. The latter figure represents a fiftyfold increase, which will require a significant investment and engineering effort. The US Government recently established the objective that wind power should supply 25% of the US energy needs by 2025. Achieving this objective will require nearly a 1200% increase in wind power capacity (from 25 GW to 305 GW). These ambitious wind energy goals put pressure on the wind energy industry research and development to significantly enhance the current wind generation capabilities in a short period of time and decrease the associated costs.

As the demands for wind energy grow, the wind energy industry is moving in the direction of very large-scale designs that are expected to reliably operate in a variety of environmental conditions. One such example is the floating wind turbines, which are expected to sustain not only increased wind loads, but also loads coming from ocean waves and underwater currents. Such designs must be analyzed with high-fidelity computational methods that incorporate complex 3D geometry, time dependence, and interaction between multiple physical systems. Accurate prediction of the aerodynamics, hydrodynamics, and structural behavior of wind turbines at full scale is even more critical in such cases in order to create reliable and efficient designs. The reliability and efficiency of the design directly translate into lower energy costs.

The present cost for wind energy is strongly dominated by the operations and maintenance of the wind turbine over its current 20-year design lifetime of the system [1]. Although the design lifetime is 20 years, it is shown in Echavarria *et al.* [2] that a typical wind turbine has averaged of 2.6 component failures per year during the first 10 years of operation. The failures usually occur within the gearbox, generator and rotor assembly, and the rotor assembly has been identified as the top opportunity for major advancement in design and performance improvements [2]. While wind turbine rotor failures occur due to a variety of reasons, fatigue failure of the wind turbine blades due to their everyday operation is recognized as one of the major causes. However, the industry is currently unable to predict these failure mechanisms, which leads to the unscheduled downtime, expensive maintenance and reduced capacity.

At the same time, offshore wind turbines are receiving increased attention. Winds in the offshore environment are typically stronger and more sustained than inland, providing a more reliable source of energy. However, offshore wind turbines are exposed to harsh environments and must be designed to withstand more severe loads than the inland wind turbines. Rotor blades of much larger diameter (> 120 m) are also being designed and built for better performance in the offshore environment. These are significant engineering challenges that must be addressed through advanced research and development, which also involves advanced and large-scale simulations.

Currently most wind turbine aerodynamics and aeroelasticity simulations are performed using low-fidelity methods, such as the Blade Element Momentum (BEM) theory for the rotor aerodynamics employed in conjunction with simplified structural models of the wind turbine blades and tower (see, e.g., Jonkman *et al.* [3, 4]). These simulations make use of steady (time-independent), 2D lumped-parameter aerodynamic models for airfoil cross-sections that are coupled with 1D beam-type structures to evaluate wind turbine blade designs and, specifically, their aerodynamic performance. These methods are simple to implement and fast to execute, which makes them attractive for industrial applications, especially if they are routinely used as part of the design cycle. However, due to the steady nature of the flow conditions and the lack of real 3D geometry and physics, these models are unable to adequately represent the system response to time-dependent phenomena, such as wind gusts, or phenomena attributable

to complex blade geometry and material composition, such as flow separation and reattachment and detailed blade deformations and stress distributions. It is precisely these more extreme events that cause gearbox and blade failures and significantly reduce the life cycle of wind turbines, leading to premature maintenance and repair and, as a result, to increased cost of wind energy. A more fundamental problem with these simple models is their non-hierarchical nature: it is virtually not possible to enhance them with features necessary for predicting more extreme events or richer physics without going to a more advanced modeling framework all together.

In this work we propose a paradigm shift in wind turbine modeling and simulation by developing 3D, complex geometry, time-dependent, multi-physics modeling procedures to obtain high-fidelity predictive simulation results for wind turbine fluid–structure interaction (FSI). We focus on predicting wind turbine rotor blade–air flow interaction phenomena for real wind turbines operating under real wind conditions and at full design scale. This is motivated by the fact that improved rotor blade efficiency directly translates to lower cost of wind energy conversion. However, simulation of wind turbines at full scale engenders a number of challenges. The flow is fully turbulent, requiring highly accurate methods, increased grid resolution, and large-scale high-performance computing. The presence of fluid boundary layers, where turbulence is created, complicates the situation further. Wind turbine blades are long, thin and slender structures with complex distribution of material properties. The numerical approach must have good approximation properties and avoid locking. The fluid–structure coupling are essential for accurate modeling of wind turbines. The motion and deformation of the wind turbine blades depend on the air flow, and the air flow patterns depend on the motion of the blades. The FSI framework must be accurate, efficient and robust to preclude divergence of the computations. The wind turbine simulations simultaneously involve moving and stationary components (such as rotor and tower, respectively) that must be handled correctly.

In recent years, there has been a growing body of research and development to address the above mentioned challenges and to raise the fidelity and predictive power of wind turbine simulations. These include standalone 3D computational fluid dynamics (CFD) simulations of wind turbines with a variety of methods [5–20], and standalone

structural analyses of rotor blades of complex geometry and material composition under assumed or computed off-line wind load conditions [21–26]. To the best of our knowledge, no coupled fluid–structure simulations of the full-scale wind turbines were attempted. While most of the CFD work on wind turbine aerodynamics is based on finite volume or finite difference discretizations, in this work we opt for numerical methods that are based on the weak or variational forms of the governing equations, such as the Finite Element Method (FEM) [27] or Isogeometric Analysis (IGA) [28, 29]. The latter is an alternative method to the FEM, which makes use of higher-order, smooth basis functions emanating from the fields of Computer-Aided Design and Computer Graphics. The choice of the variational approach is based on several considerations, the two of which are: 1. It enables the formulation of novel, multiscale turbulence modeling approaches [30], which do not rely on the empiricism of eddy-viscosity methods; 2. The coupling with structural mechanics (typically computed with FEM or IGA) for the purposes of FSI modeling is greatly simplified. These methods are summarized in what follows and are described in greater detail in the body of this dissertation.

Isogeometric analysis, first introduced by Hughes *et al.* [28] and further expanded on by many researchers [29, 31–41], is adopted as the geometry modeling and simulation framework for wind turbines. We use the IGA based on NURBS (non-uniform rational B-splines), which are more efficient than standard finite elements for representing complex, smooth geometries, such as wind turbine blades. Because the geometry and solution fields are represented using the same functional description, the integration of geometry modeling with structural design and computational analysis is greatly simplified. The IGA was successfully employed for computation of turbulent flows [30, 42–46], nonlinear structures [47–54], and FSI [55–60]. In most cases, IGA gave a clear advantage over standard low-order finite elements in terms of solution accuracy per-degree-of-freedom. This is in part attributable to the higher-order smoothness of the basis functions employed. Flows about rotating components are naturally handled in an isogeometric framework because all conic sections, and in particular, circular and cylindrical shapes, are represented exactly [61]. In addition, an isogeometric representation of the analysis-suitable geometry may be used to construct tetrahedral and hexahedral meshes for computations using the FEM. In this work, we use such tetrahedral

meshes for wind turbine aerodynamics computation using the Arbitrary Lagrangian–Eulerian (ALE) [62] versions of the Residual-Based Variational Multiscale (RBVMS) method [30, 63–65], which we refer to as the "ALE–VMS" method.

The ALE–VMS formulation of the Navier–Stokes equations of incompressible flows is employed in this work for the modeling of wind turbine aerodynamics. The ALE–VMS formulation originated from the RBVMS formulation of incompressible turbulent flows proposed in Bazilevs *et al.* [30] for stationary meshes, and may be thought of as an extension of the RBVMS method to moving meshes. As such, it was presented for the first time in Bazilevs *et al.* [57] in the context of FSI. Although ALE–VMS gave reasonably good results for several important turbulent flows, it was evident in Bazilevs *et al.* [30] and Akkerman *et al.* [44] that to obtain accurate results for wall-bounded turbulent flows the method required relatively fine resolution of the boundary layers. This fact makes ALE–VMS a somewhat costly technology for full-scale wall-bounded turbulent flows at high Reynolds numbers, which are characteristic of the present application. For this reason, weakly enforced essential boundary condition formulation was introduced in Bazilevs and Hughes [66], which significantly improved the performance of the ALE–VMS formulation in the presence of unresolved boundary layers [42, 43, 46]. The weak boundary condition formulation may be thought of as an extension of Nitsche's method [67] to the Navier–Stokes equations of incompressible flows. Another interpretation of the weak boundary condition formulation is that it is a Discontinuous Galerkin method (see, e.g., Arnold *et al.* [68]), where the continuity of the basis functions is enforced everywhere in the domain interior, but not at the domain boundary.

Wind turbine blades are long, thin and slender structures that are made of several structural components with complex distribution of material properties, requiring both advanced computational model generation and simulation methods. The numerical approach for structural mechanics must have good accuracy and avoid locking, which is typical of thin structures [69]. Wind turbine blades are manufactured using multi-layer composite materials that also require appropriate numerical treatment. In this work, the blade structure is governed by the isogeometric Kirchhoff–Love composite shell formulation with the aid of the bending strip method. This rotation-free formulation is appropriate for thin-shell structures comprised of multiple C^1- or higher-order continu-

ous surface patches that are joined or merged with continuity no greater than C^0. The Kirchhoff–Love shell theory that relies on higher-order continuity of the basis functions is employed in the patch interior as in Kiendl *et al.* [52]. Strips of fictitious material with unidirectional bending stiffness and with zero mass and membrane stiffness are added at patch interfaces in the overlapping fashion. The direction of bending stiffness is chosen to be transverse to the patch interface. This choice leads to an approximate satisfaction of the appropriate kinematic constraints at patch interfaces without introducing additional stiffness to the shell structure. Discretizations such as NURBS [70], T-splines [36, 37] or Subdivision surfaces [71–73] are perfectly suited for the proposed structural modeling method. We also note that the lack of rotational degrees of freedom in the structural discretization facilitates the strong FSI coupling.

In this work, we also focus on the coupling strategies, specific to IGA, for FSI. We formulate computational approaches for IGA that make use of non-matching discretizations of the domain geometry at the interface between the fluid and structure subdomains. The proposed FSI methodology is based on the augmented Lagrangian concept. We formally eliminate the Lagrange multiplier on the fluid–structure interface to achieve a formulation purely in terms of the primal variables (i.e., velocity and pressure for the fluid and velocity for the structural problem). The fluid and structural mechanics trial and test functions are not assumed to be equal at the interface, and the FSI coupling is taken care of in the variational formulation. This presents a convenient point of departure for a discrete FSI formulation using non-matching fluid–structure interface discretizations. In the non-matching FSI formulation one needs a functional definition of the kinematic quantities (velocity, displacement, etc.) and tractions on the fluid and structure meshes, and the ability to "transfer" these quantities from one mesh to the other. The L^2-projection is chosen in this work for the transfer of kinematic and traction data between the subdomains.

The non-matching FSI coupling procedures allow greater flexibility in the computational analysis, and alleviate the difficulties of geometry modeling arising from the necessity to construct matching multi-physics interface discretizations. Furthermore, the mesh resolution of the structure and fluid mechanics problems may be tailored to the analysis requirements of the individual subsystems, leading to improved computational

efficiency. The increased complexity of structural geometry places heavy demands on the fluid volume mesh (or exterior mesh) generation around that structure. However, no "automatic mesh generation" software for IGA exists to this day. Volumetric geometry modeling and mesh generation for IGA is, at this point, an active area of research that is still in its infancy (see Wang *et al.* [74, 75] for recent results for this challenging problem). As a result, in order to take advantage of the superior accuracy of IGA for structural mechanics applications (see, e.g., Cottrell *et al.* [31] and Auricchio *et al.* [38]), and to leverage the existing advanced volumetric mesh generation tools for standard finite elements, we propose to couple FEM for fluid mechanics and IGA for structural dynamics. Although IGA was shown to produce results that are of better per-degree-of-freedom quality than standard FEM for fluid mechanics and, especially, for turbulent flows [44], good-quality aerodynamics may still be achieved with standard low-order FEM, with a manageable number of degrees of freedom. This is demonstrated in this work.

This dissertation is outlined as follows. In Chapter 2, we review the basis of IGA. We also focus on geometry modeling and describe a template-based method for creating an analysis-suitable NURBS geometry for FSI simulation of wind turbine rotors. We apply the developed method to construct the NREL 5 MW offshore baseline wind turbine rotor [4]. In Chapter 3, we introduce the ALE–VMS formulation of the Navier–Stokes equations of incompressible flow augmented with weakly enforced essential boundary conditions for wind turbine aerodynamics. We present the aerodynamics computations of the NREL 5 MW wind turbine rotor and the NREL Phase VI wind turbine. The simulation results of the NREL Phase VI wind turbine are compared with the experimental data extensively. A sliding interface method is also presented to handle computational domains in relative motion and for the simulation of the full wind turbine configuration to study the effect of blade–tower interaction. We conclude this chapter by showing the importance of employing weak boundary conditions for predicting wind turbine aerodynamics using the ALE–VMS method. In Chapter 4, we present the structural mechanics formulation, which is based on the Kirchhoff–Love thin-shell theory and the bending strip method. We also describe our composite material modeling procedures for wind turbine blades. We present the numerical results, which include linear and non-linear

benchmark problems, and the application to the NREL 5 MW offshore baseline wind turbine blade. A method and algorithm for pre-bending of wind turbine blades is also described. In Chapter 5, we propose an non-matching FSI methodology based on the augmented Lagrangian concept and formally eliminate the Lagrange multiplier on the fluid–structure interface to achieve a formulation purely in terms of the primal variables. The fluid and structural mechanics trial and test functions are not assumed to be equal at the interface, and the FSI coupling is taken care of in the variational formulation. We show an application of the proposed FSI formulation to the simulation of the NREL 5 MW wind turbine rotor, where the aerodynamics domain is modeled using volumetric quadratic NURBS and the rotor structure is modeled using a cubic T-spline-based discretization of a rotation-free Kirchhoff–Love shell. We also show the FSI coupling of a T-spline shell with a low-order FEM discretization of the aerodynamics equations. A discussion of the mesh motion procedures employed in the FSI simulations of the wind turbine rotor is also provided. In Chapter 6, we draw conclusions and outline future research directions.

Chapter 2

Isogeometric Analysis and Geometry Modeling for Wind Turbine

2.1 Basics of the Isogeometric Analysis

The concept of Isogeometric Analysis (IGA) was introduced by Hughes *et al.* [28] in 2005. The introduction of this innovative idea represents a need for a tighter integration between engineering design and simulation. The concept of IGA also signifies a closer collaboration in terms of the approaches used in the fields of engineering design, which primarily uses Computer-Aided Design (CAD), and engineering simulation, which is mainly based on the Finite Element Method (FEM). The main idea behind IGA is to focus on a single geometric model. The single geometric model can be utilized directly as an analysis model, or can be used to build geometrically precise analysis models. To demonstrate such an idea, a change from the classical FEM to an analysis procedure based on CAD representations is required.

Several possible technologies from computational geometry may be used in IGA. Non-uniform rational B-splines (NURBS) [70, 76, 77] is the industry standard tool for geometry modeling and is most widely used in engineering design. The major strengths of NURBS include: they are convenient for free-form surface modeling; they can represent all conic sections (and therefore circles, cylinders, spheres, ellipsoids, and other special geometries) exactly; there exist many efficient and numerically stable

algorithms to generate NURBS objects. Besides, NURBS also have useful mathematical properties, such as good approximation, and the ability to be refined through knot insertion. Therefore, NURBS, a CAD standard representing many years of development, were the natural starting point for IGA.

A recently developed generalization of NURBS technology is T-splines [78, 79]. The main idea of T-splines is that they extend NURBS to permit local refinement (and coarsening). T-splines are backward- and forward-compatible with NURBS, and this feature makes them an attractive CAD technology. Preliminary exploration of T-splines as IGA technology may be found in Bazilevs *et al.* [37] and Dörfel *et al.* [36]. Recent results on linear independence and improved local refinement algorithms may be found in Li *et al.* [80] and Scott *et al.* [81]. The work of Wang *et al.* [75] represents a recent attempt to construct solid T-Splines from existing hexahedral meshes.

Recent efforts in IGA to define standardized file formats for data exchange between geometry modeling and computational analysis software enabled straightforward solution of complicated structural problems that involve large deformation, plasticity and contact, using well-validated commercial FEM software [49–51, 82, 83]. Furthermore, IGA is able to handle many applications that otherwise create significant challenges to standard finite element technology [47, 52, 84, 85]. For more detailed mathematical developments, basis function research, geometry modeling, model quality assessment, and early applications, the reader is referred to Cottrell *et al.* [29] and references therein.

2.2 Analysis-Suitable Geometry Construction for Wind Turbine Rotors

Isogeometric analysis is adopted in this work as the geometry modeling and simulation framework for wind turbines. In this section we propose a template-based wind turbine geometry modeling approach that makes use of surface and volumetric NURBS. The method entails construction of one or more (small number of) template geometries of wind turbine designs. The templates include and closely approximate the geometry of the rotor blades and hub, and the flow domain around. The template geometry is

then deformed to the actual geometry of the wind turbine by appropriately minimizing the error between them. Once the model is generated, an analysis-ready geometry is produced with user control over mesh refinement and domain partitioning for efficient parallel processing. The advantage of this approach is that it can be specialized and optimized for a particular class of geometries. For example, a template-based geometry modeling approach was developed and successfully employed for NURBS modeling and FSI simulation of vascular blood flow with patient-specific data in Zhang *et al.* [55].

2.2.1 Wind Turbine Rotor Blade and Hub Geometry Construction

As a first step we construct a template for the structural model of the rotor. Here, the structural model is limited to a surface (shell) representation of the wind turbine blade, the hub, and their attachment zone. The blade surface is assumed to be composed of a collection of airfoil shapes that are lofted in the blade axis direction.

The geometry of the rotor blade considered in this example is based on the NREL 5 MW offshore baseline wind turbine described in Jonkman *et al.* [4]. The blade geometry data taken from the reference is summarized in Table 2.1. A 61 m blade is attached to a hub with radius of 2 m, which gives the total rotor radius of 63 m. The blade is composed of several airfoil types provided in the rightmost column of the table, and the corresponding profiles are given in Figure 2.1. The first portion of the blade is a perfect cylinder. Further away from the root the cylinder is smoothly blended into a series of DU (Delft University) airfoils. At the 44.55 m location away from the root the NACA64 profile is used to define the blade all the way to the tip (see Figure 2.2). The remaining parameters from Table 2.1 are defined in Figure 2.2: “RNodes” is the distance from the rotor center to the airfoil cross-section in the blade axis direction. “AeroTwst” is the twist angle for a given cross-section. The blades are twisted to enhance the aerodynamic performance. “Chord” is the chord length of the airfoil. “AeroOrig” is the location of the aerodynamic center. For most of the blade airfoil cross-sections, the aerodynamic center is taken at 25% of the chord length from the leading edge. To accommodate the cylindrical shape at the root, the aerodynamic center is gradually moved to 50% of the chord length. This is not reported in Jonkman *et al.* [4], but mentioned in Kooijman *et al.* [86].

Table 2.1: Wind turbine rotor geometry definition for NREL 5 MW offshore baseline wind turbine [4].

RNodes (m)	AeroTwst (°)	Chord (m)	AeroCent (-)	AeroOrig (-)	Airfoil (-)
2.0000	0.000	3.542	0.2500	0.50	Cylinder
2.8667	0.000	3.542	0.2500	0.50	Cylinder
5.6000	0.000	3.854	0.2218	0.44	Cylinder
8.3333	0.000	4.167	0.1883	0.38	Cylinder
11.7500	13.308	4.557	0.1465	0.30	DU40
15.8500	11.480	4.652	0.1250	0.25	DU35
19.9500	10.162	4.458	0.1250	0.25	DU35
24.0500	9.011	4.249	0.1250	0.25	DU30
28.1500	7.795	4.007	0.1250	0.25	DU25
32.2500	6.544	3.748	0.1250	0.25	DU25
36.3500	5.361	3.502	0.1250	0.25	DU21
40.4500	4.188	3.256	0.1250	0.25	DU21
44.5500	3.125	3.010	0.1250	0.25	NACA64
48.6500	2.310	2.764	0.1250	0.25	NACA64
52.7500	1.526	2.518	0.1250	0.25	NACA64
56.1667	0.863	2.313	0.1250	0.25	NACA64
58.9000	0.370	2.086	0.1250	0.25	NACA64
61.6333	0.106	1.419	0.1250	0.25	NACA64
62.9000	0.000	0.700	0.1250	0.25	NACA64
63.0000	0.000	0.000	0.1250	0.25	NACA64

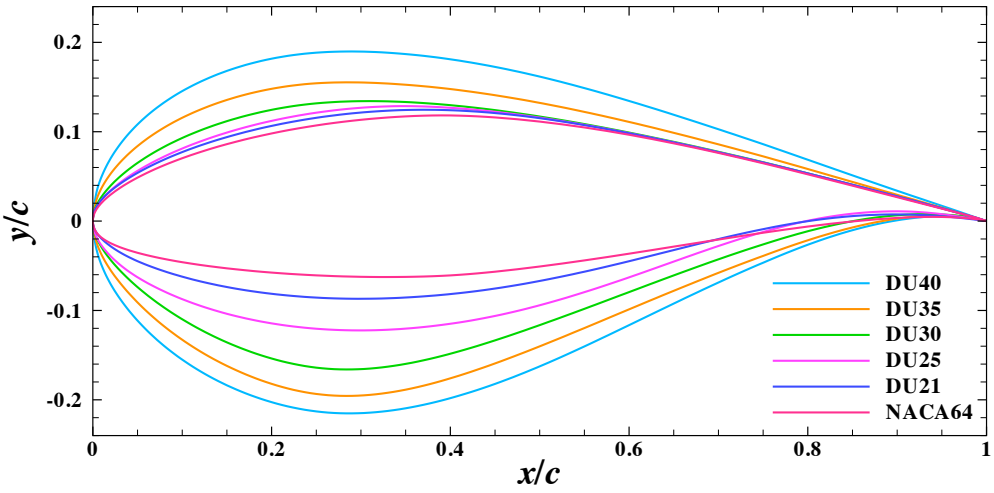


Figure 2.1: Airfoil cross-sections used in the design of the wind turbine rotor blades.

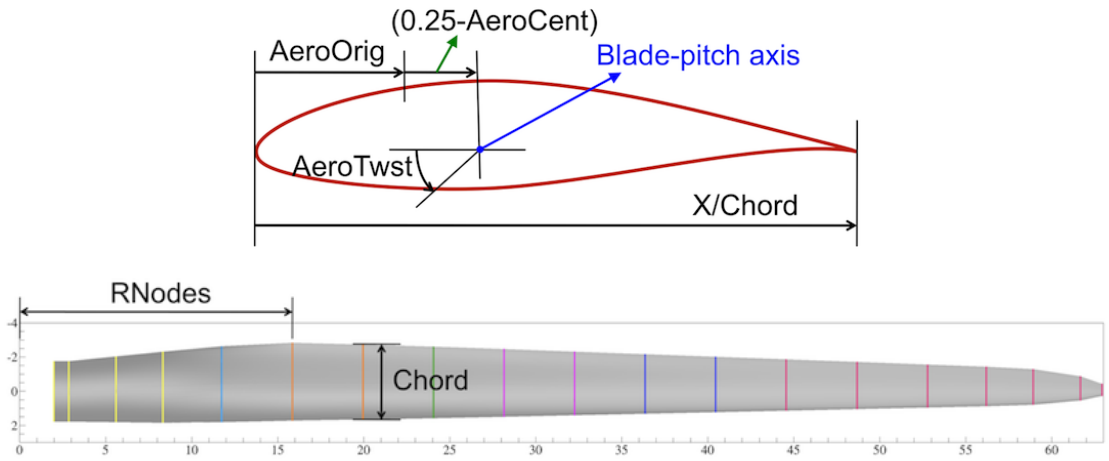
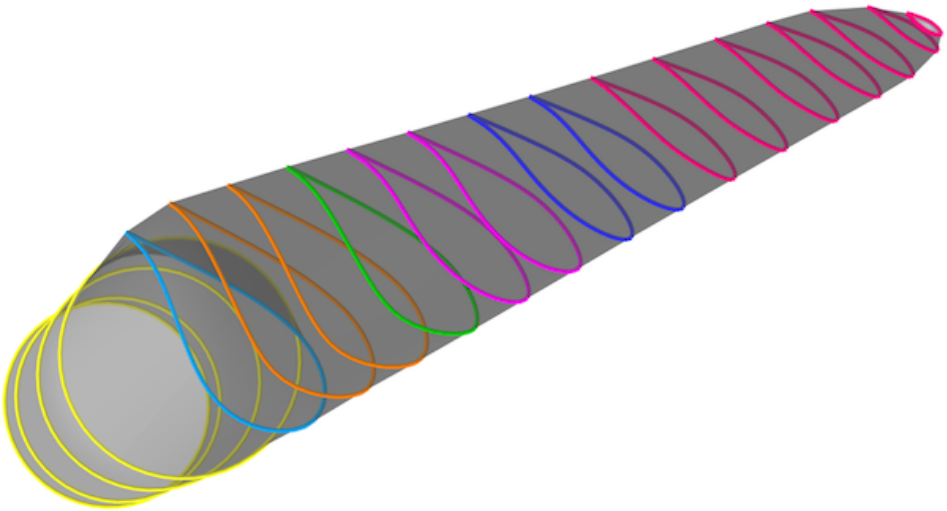


Figure 2.2: Illustration of quantities from Table 2.1.

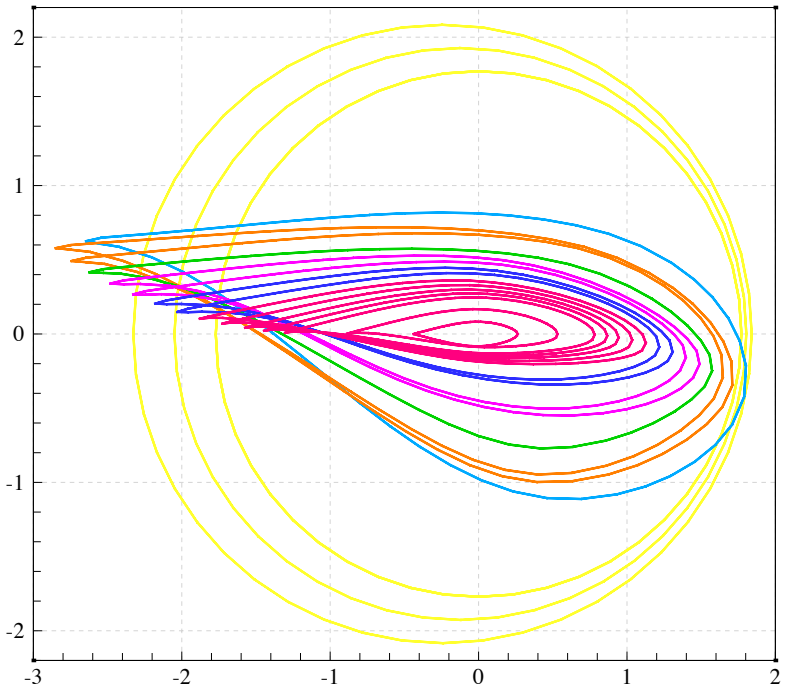
Remark 2.1. There is some redundancy in the parameters given in Table 2.1. The variable "AeroCent" is used as an input to FAST [3], which is the aerodynamics modeling software that is typically used for wind turbine rotor computations. FAST is based on look-up tables and provides blade cross-section steady-state lift and drag forces given the airfoil type, relative wind speed, and angle of attack. The effects of the hub, trailing edge turbulence, and blade tip are modeled using empirical relationships. FAST assumes that the blade-pitch axis passes through each airfoil section at 25% chord length, and defines "AeroCent − 0.25" to be the fractional distance to the aerodynamic center from the blade-pitch axis along the chordline, positive toward the trailing edge. Therefore, "AeroOrig + (0.25 − AeroCent)" gives the location of where the blade-pitch axis passes through each airfoil cross-section. Although for our purposes this added complexity is unnecessary, the same naming system is used for backward compatibility with the reference reports.

For each blade cross-section we use quadratic NURBS to represent the 2D airfoil shape. The weights of the NURBS functions are set to unity. The weights are adjusted near the root to represent the circular cross-sections of the blade exactly. The cross-sections are lofted in the blade axis direction, also using quadratic NURBS and unity weights. This geometry modeling procedure produces a smooth rotor blade surface using a relatively small number of input parameters, which is an advantage of the isogeometric representation. The final shape of the blade along with the airfoil cross-

sections is shown in Figure 2.3(a). Figure 2.3(b) shows a top view of the blade in which the twisting of the cross-sections is evident.



(a)



(b)

Figure 2.3: (a) Airfoil cross-sections superposed on the wind turbine blade. (b) Top view of a subset of the airfoil cross-sections illustrating blade twisting.

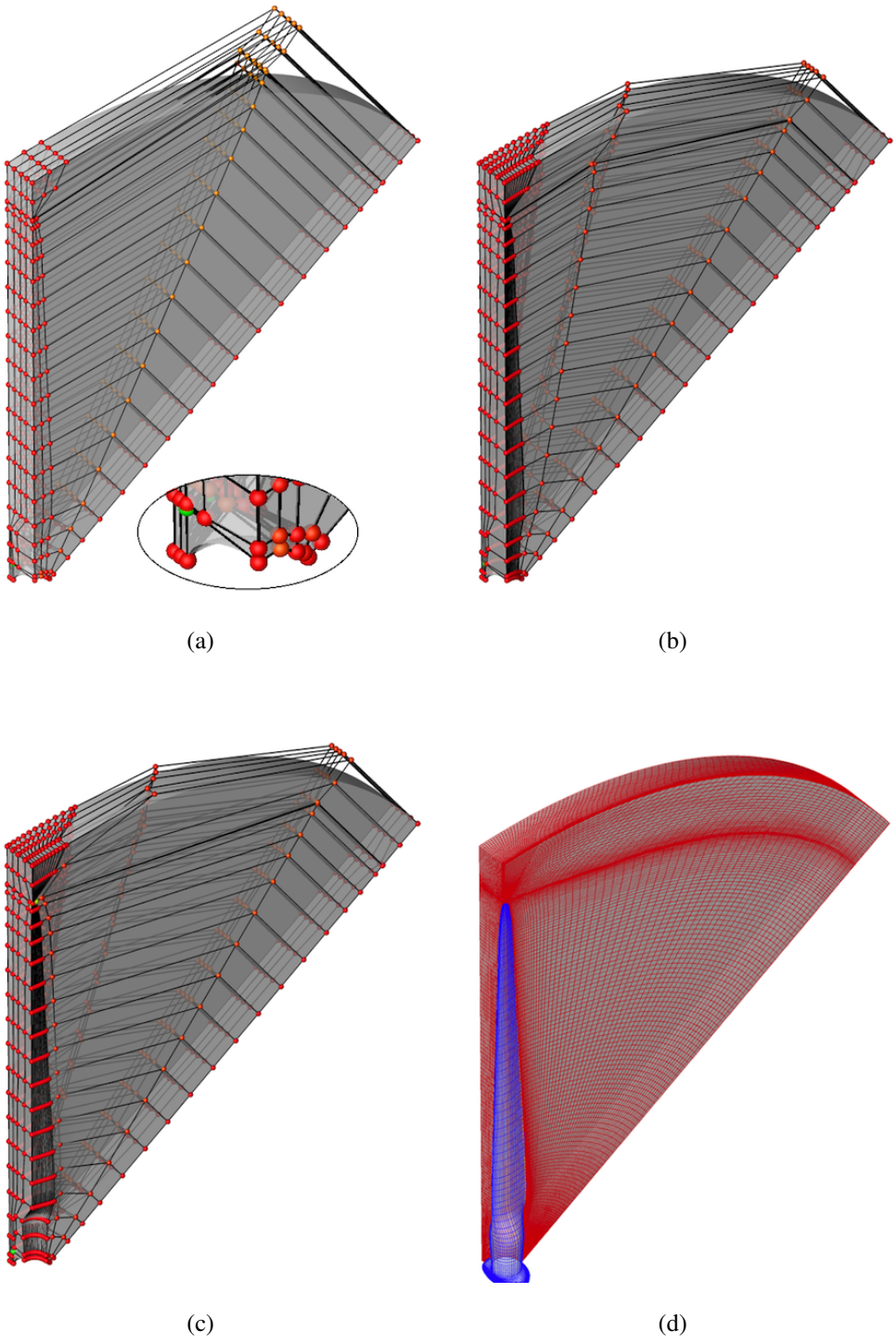
2.2.2 Volumetric NURBS

Given the rotor blade surface description, the surrounding fluid domain volume is constructed next. The blade surface is split into four patches of similar size, which we call the blade surface patches. The splitting is done at the leading and trailing edges, as well as half-way in between on both sides of the blade. The volumetric fluid domain near the blade is generated for each one of the blade surface patches. As a final step, the fluid domain patches are merged such that the outer boundary of the fluid domain is a perfect cylinder.

For each of the blade surface patches we create a 60° pie-shaped domain using a minimum required number of control points. The control points at the bottom of the patch are moved to accommodate the shape of the rotor hub (see Figure 2.4(a)). As a next step, we perform knot insertion and move the new control points such that their locations coincide with those of the blade surface patch. See Figure 2.4(b) and Figure 2.4(c) for knot insertion and control point re-positioning, respectively. This generates an a-priori conforming discretization between the volumetric fluid domain and the surface of the structural model, suitable for FSI analysis. Finally, the fluid domain is refined in all parametric directions for analysis. Figure 2.4(d) shows the rotor surface mesh and one of the fluid mesh subdomains adjacent to it. The remaining fluid subdomains are generated in the same manner.

The resultant fluid NURBS mesh may be embedded into a larger domain for the purposes of simulation. In this work we take this larger domain to also be a cylinder. For computational efficiency, only one-third of the domain is modeled. The fluid volume mesh, corresponding to one-third of the fluid domain, consists of 1,449,000 quadratic NURBS elements (and a similar number of control points) and is shown in Figure 2.5(a). The fluid mesh cross-section that also shows the details of mesh refinement in the boundary layer is shown in Figure 2.5(b). The wind turbine rotor surface model and outer cylindrical domain of the three-blade configuration are shown in Figure 2.6. Mesh refinement was concentrated near the wind turbine blade surface. No special attention was paid to mesh design in the outer regions of the computational domain.

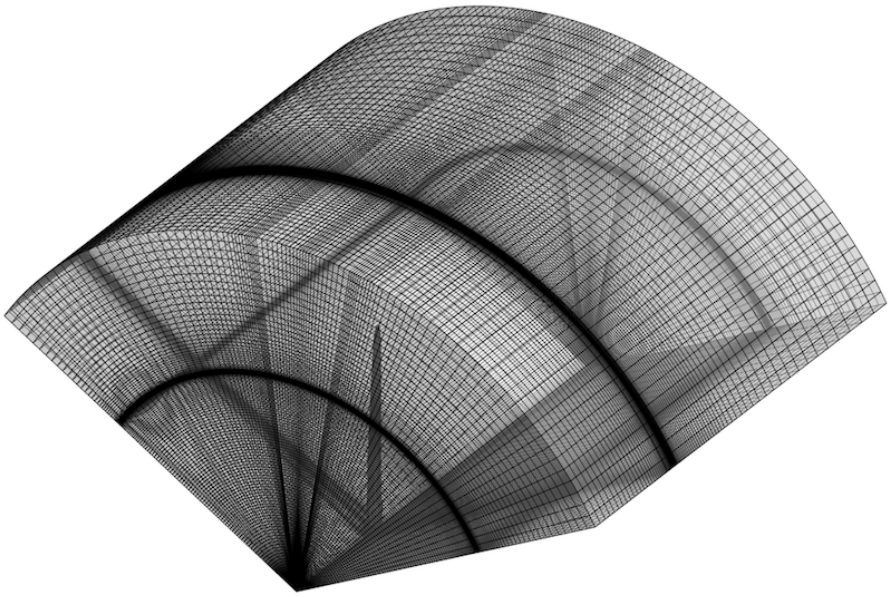
Remark 2.2. We note that the template-based nature of the proposed approach and a relatively small number of input parameters for model building allow one to alter the



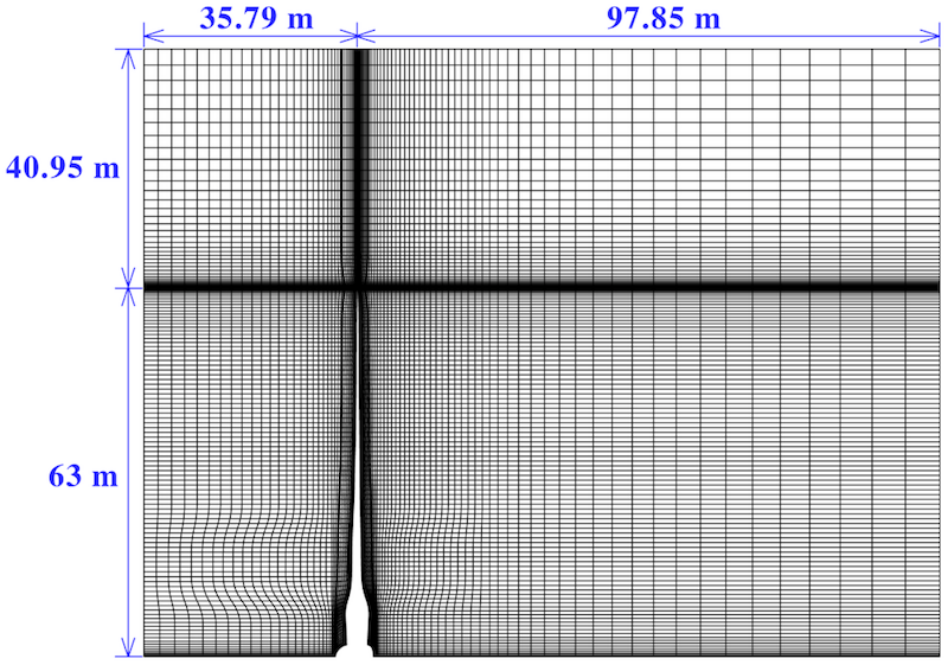
(a) (b)

(c) (d)

Figure 2.4: Stages of analysis-suitable geometry construction for wind turbine rotor simulations.

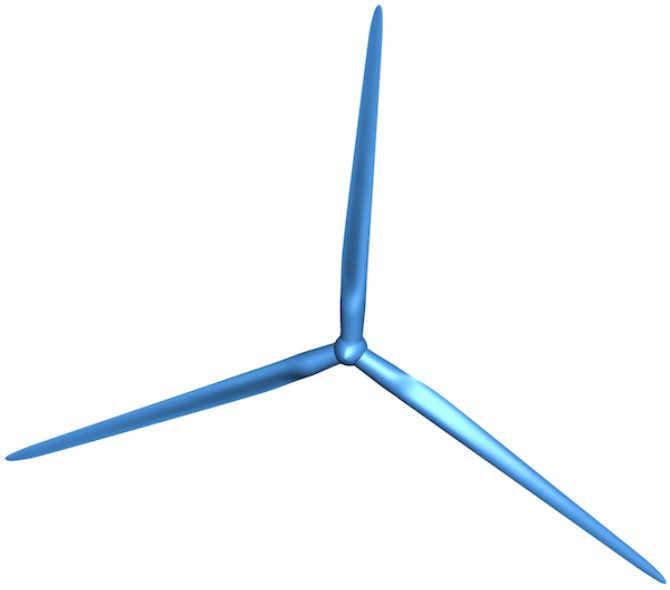


(a)

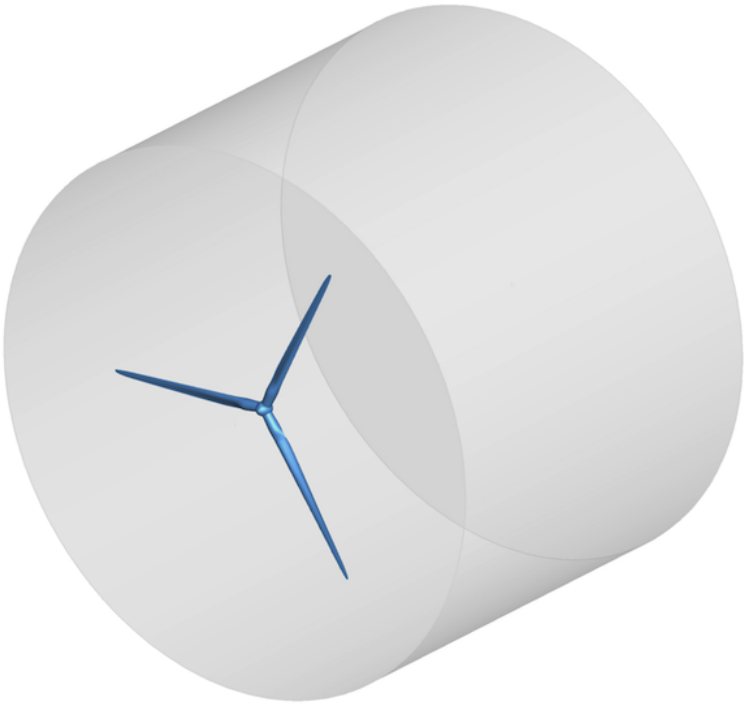


(b)

Figure 2.5: (a) Volumetric NURBS mesh of the computational domain. (b) A planar cut to illustrate mesh grading toward the rotor blade.



(a)



(b)

Figure 2.6: (a) Wind turbine rotor surface. (b) Full problem domain. Both are obtained by merging the 120° rotationally-periodic domains.

wind turbine rotor models with minimal effort.

2.3 Acknowledgements

Chapter 2, in part, is a reprint of the material as it appears in: "ALE–VMS and ST–VMS methods for computer modeling of wind-turbine rotor aerodynamics and fluid–structure interaction," (with Y. Bazilevs, K. Takizawa and T.E. Tezduyar), *Mathematical Models and Methods in Applied Sciences*, 2012. The dissertation author was the primary investigator and author of this paper.

Chapter 3

Modeling and Simulation of Wind Turbine Aerodynamics

3.1 ALE–VMS Formulations of the Navier–Stokes Equations of Incompressible Flows

The wind turbine aerodynamics is governed by the Navier–Stokes equations of incompressible flow posed on a moving domain. The domain motion is handled using the arbitrary Lagrangian–Eulerian (ALE) framework [62]. Although aerodynamic phenomena are generally described using the Navier–Stokes equations of compressible flow, the incompressible-flow assumption is valid for the present application where the Mach number typically low.

3.1.1 Continuous Problem

In what follows, $\Omega_t \in \mathbb{R}^d$, $d = 2, 3$, denotes the spatial domain of the aerodynamics problem with boundary Γ_t at time $t \in (0, T)$. The subscript t indicates that the fluid mechanics spatial domain is time-dependent. The Navier–Stokes equations of incompressible flows in the ALE frame may be written on Ω_t and $\forall t \in (0, T)$ as

$$\begin{aligned} \rho\left(\left.\frac{\partial \mathbf{u}}{\partial t}\right|_{\hat{x}} + (\mathbf{u} - \hat{\mathbf{u}}) \cdot \boldsymbol{\nabla}\mathbf{u} - \mathbf{f}\right) - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} &= \mathbf{0}, \\ \boldsymbol{\nabla} \cdot \mathbf{u} &= 0, \end{aligned} \tag{3.1}$$

where ρ is the density of the fluid, $\mathbf{u}$ is the velocity, $\hat{\mathbf{u}}$ is the velocity of the fluid domain Ω_t, $\mathbf{f}$ is the external force, and $\boldsymbol{\sigma}$ is the stress tensor defined as

$$\boldsymbol{\sigma}(\mathbf{u}, p) = -p\mathbf{I} + 2\mu\boldsymbol{\varepsilon}(\mathbf{u}). \tag{3.2}$$

Here p is the pressure, $\mathbf{I}$ is the identity tensor, μ is the dynamic viscosity, and $\boldsymbol{\varepsilon}(\mathbf{u})$ is the strain-rate tensor given by

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2}\left(\boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^T\right). \tag{3.3}$$

In Eq. (3.1), the notation $\Big|_{\hat{x}}$ implies that the time derivative is taken with respect to a fixed referential-domain spatial coordinates $\hat{\mathbf{x}}$. The spatial gradients $\boldsymbol{\nabla}$ are taken with respect to the spatial coordinates $\mathbf{x}$ of the current configuration.

3.1.2 ALE–VMS Formulation of Fluid Mechanics

The ALE–VMS formulation of the continuum aerodynamics formulation is given as follows: find $\mathbf{u}^h \in \mathcal{S}_u^h$ and $p^h \in \mathcal{S}_p^h$, such that $\forall \mathbf{w}^h \in \mathcal{V}_u^h$ and $\forall q^h \in \mathcal{V}_p^h$:

$$B^{\mathrm{VMS}}(\{\mathbf{w}^h, q^h\}, \{\mathbf{u}^h, p^h\}; \hat{\mathbf{u}}^h) - F(\{\mathbf{w}^h, q^h\}) = 0, \tag{3.4}$$

where

$$\begin{aligned}
&B^{\mathrm{VMS}}(\{\mathbf{w}^h, q^h\}, \{\mathbf{u}^h, p^h\}; \hat{\mathbf{u}}^h) = \\
&\int_{\Omega_t} \mathbf{w}^h \cdot \rho\left(\left.\frac{\partial \mathbf{u}^h}{\partial t}\right|_{\hat{x}} + \left(\mathbf{u}^h - \hat{\mathbf{u}}^h\right)\cdot\boldsymbol{\nabla}\mathbf{u}^h\right)\,\mathrm{d}\Omega + \int_{\Omega_t} \boldsymbol{\varepsilon}\left(\mathbf{w}^h\right) : \boldsymbol{\sigma}\left(\mathbf{u}^h, p^h\right)\,\mathrm{d}\Omega \\
&+ \int_{\Omega_t} q^h \boldsymbol{\nabla}\cdot\mathbf{u}^h\,\mathrm{d}\Omega \\
&+ \sum_{e=1}^{N_{\mathrm{el}}} \int_{\Omega_t^e} \tau_{\mathrm{M}}\left(\left(\mathbf{u}^h - \hat{\mathbf{u}}^h\right)\cdot\boldsymbol{\nabla}\mathbf{w}^h + \frac{\boldsymbol{\nabla} q^h}{\rho}\right)\cdot\mathbf{r}_{\mathrm{M}}\left(\mathbf{u}^h, p^h\right)\,\mathrm{d}\Omega \\
&+ \sum_{e=1}^{N_{\mathrm{el}}} \int_{\Omega_t^e} \rho\tau_{\mathrm{C}}\boldsymbol{\nabla}\cdot\mathbf{w}^h r_{\mathrm{C}}(\mathbf{u}^h)\,\mathrm{d}\Omega \\
&- \sum_{e=1}^{N_{\mathrm{el}}} \int_{\Omega_t^e} \tau_{\mathrm{M}}\mathbf{w}^h\cdot\left(\mathbf{r}_{\mathrm{M}}\left(\mathbf{u}^h, p^h\right)\cdot\boldsymbol{\nabla}\mathbf{u}^h\right)\,\mathrm{d}\Omega \\
&- \sum_{e=1}^{N_{\mathrm{el}}} \int_{\Omega_t^e} \frac{\boldsymbol{\nabla}\mathbf{w}^h}{\rho} : \left(\tau_{\mathrm{M}}\mathbf{r}_{\mathrm{M}}\left(\mathbf{u}^h, p^h\right)\right)\otimes\left(\tau_{\mathrm{M}}\mathbf{r}_{\mathrm{M}}\left(\mathbf{u}^h, p^h\right)\right)\,\mathrm{d}\Omega,
\end{aligned} \tag{3.5}$$

and

$$F(\{\mathbf{w}^h, q^h\}) = \int_{\Omega_t} \mathbf{w}^h \cdot \rho \mathbf{f} \, \mathrm{d}\Omega + \int_{(\Gamma_t)_\mathrm{h}} \mathbf{w}^h \cdot \mathbf{h} \, \mathrm{d}\Gamma. \tag{3.6}$$

In the above formulation the time-dependent fluid domain Ω_t is divided into N_el individual spatial element subdomains denoted by Ω_t^e. The trial function spaces $\mathcal{S}_u^h$ for the velocity and $\mathcal{S}_p^h$ for the pressure, as well as the corresponding test function spaces $\mathcal{V}_u^h$ and $\mathcal{V}_p^h$ are assumed to be of equal order, and the superscrip h denotes their dependence on the mesh size. In Eqs. (3.4)-(3.6), $\hat{\mathbf{u}}^h$ is the mesh velocity, $\mathbf{h}$ is the prescribed traction boundary condition on $(\Gamma_t)_\mathrm{h}$, and $\mathbf{r}_\mathrm{M}$ and r_C are the residuals of the momentum and continuity equations, respectively, given by

$$\mathbf{r}_\mathrm{M}(\mathbf{u}^h, p^h) = \rho \left(\left. \frac{\partial \mathbf{u}^h}{\partial t} \right|_{\hat{x}} + \left(\mathbf{u}^h - \hat{\mathbf{u}}^h \right) \cdot \boldsymbol{\nabla} \mathbf{u}^h - \mathbf{f} \right) - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} \left(\mathbf{u}^h, p^h \right) \tag{3.7}$$

and

$$r_\mathrm{C}(\mathbf{u}^h) = \boldsymbol{\nabla} \cdot \mathbf{u}^h. \tag{3.8}$$

Also in Eq. (3.5), τ_M and τ_C are the stabilization parameters defined as

$$\tau_\mathrm{M} = \left(\frac{4}{\Delta t^2} + \left(\mathbf{u}^h - \hat{\mathbf{u}}^h \right) \cdot \mathbf{G} \left(\mathbf{u}^h - \hat{\mathbf{u}}^h \right) + C_I \nu^2 \mathbf{G} : \mathbf{G} \right)^{-1/2} \tag{3.9}$$

and

$$\tau_\mathrm{C} = \left(\mathrm{tr}\mathbf{G} \; \tilde{\tau}_\mathrm{M} \right)^{-1}, \tag{3.10}$$

where

$$\tilde{\tau}_\mathrm{M} = \left(\left(\mathbf{u}^h - \hat{\mathbf{u}}^h \right) \cdot \mathbf{G} \left(\mathbf{u}^h - \hat{\mathbf{u}}^h \right) + C_I \nu^2 \mathbf{G} : \mathbf{G} \right)^{-1/2}. \tag{3.11}$$

Here, Δt is the time-step size, $\nu = \mu/\rho$ is the kinematic viscosity, $\mathbf{G}$ is the element metric tensor defined as

$$G_{ij} = \sum_{k=1}^{d} \frac{\partial \xi_k}{\partial x_i} \frac{\partial \xi_k}{\partial x_j}, \tag{3.12}$$

where $\partial \boldsymbol{\xi} / \partial \mathbf{x}$ is the inverse Jacobian of the element mapping between the parametric and physical domain, tr$\mathbf{G}$ is the trace of $\mathbf{G}$, and C_I is a positive constant, independent of the mesh size, derived from an appropriate element-wise inverse estimate [87–89].

Remark 3.1. The stabilization parameters τ_{M} and τ_{C} in the above equations originate from stabilized finite element methods for fluid dynamics (see, e.g., [45, 90–97]). The stabilization parameters were designed and studied extensively in the context of stabilized finite element formulations of linear model problems of direct relevance to fluid mechanics. These model problems include advection–diffusion, Stokes and Oseen equations. The design of τ_{M} and τ_{C} is such that optimal convergence with respect to the mesh size and polynomial order of discretization is attained for these cases (see, e.g., Hughes *et al.* [96] and references therein). Furthermore, enhanced stability for advection-dominated flows and the ability to conveniently employ the same basis functions for velocity and pressure variables for incompressible flow are some of the attractive outcomes of this method. More recently, the stabilization parameters were derived in the context of the variational multiscale (VMS) methods [63, 98] and were interpreted as the appropriate averages of the small-scale Green's function, a key mathematical object in the theory of VMS methods (see Hughes and Sangalli [65] for an elaboration).

Remark 3.2. The ALE–VMS formulation is a moving-mesh extension of the residual-based variational multiscale (RBVMS) turbulence modeling technique proposed for stationary meshes by Bazilevs *et al.* [30]. It was also presented by Bazilevs *et al.* [57] for moving meshes in the context of FSI. Calderer and Masud [99] derived a variant of the ALE–VMS method in the context of residual-free bubbles [100]. Recently, Takizawa and Tezduyar [101] introduced a space–time version of the RBVMS method for computations with moving meshes. Note that, in contrast to classical turbulence modeling approaches, no ad hoc eddy viscosity terms are present in the RBVMS or ALE–VMS formulation, which is one of its distinguishing features.

3.1.3 Weakly Enforced Essential Boundary Conditions

In this section we state the formulation of the weakly enforced essential boundary conditions. This was first proposed by Bazilevs and Hughes [66] for the advection–diffusion equation and Navier–Stokes equations of incompressible flows in an effort to improve the accuracy of stabilized and multiscale formulations in the presence of unresolved boundary layers. The weak boundary condition formulation may thought of as an extension of Nitsche's method [67] to the case of the Navier–Stokes equations of

incompressible flow. Another interpretation of the weak boundary condition formulation is that it is a Discontinuous Galerkin method (see, e.g., Arnold *et al.* [68]), where the continuity of the basis functions is enforced everywhere in the domain interior, but not at the domain boundary. The method for the weakly enforced boundary condition was further refined and studied in a set of challenging wall-bounded turbulent flows in Bazilevs *et al.* [42,43,46].

To account for the weak enforcement of the essential boundary conditions, we remove the essential boundary conditions from the trial and test function sets $\mathcal{S}_u^h$ and $\mathcal{V}_u^h$, and modify the discrete variational formulation (3.4) as: find $\mathbf{u}^h \in \mathcal{S}_u^h$ and $p^h \in \mathcal{S}_p^h$, such that $\forall \mathbf{w}^h \in \mathcal{V}_u^h$ and $\forall q^h \in \mathcal{V}_p^h$:

$$B^{\mathrm{VMS}}(\{\mathbf{w}^h, q^h\}, \{\mathbf{u}^h, p^h\}; \hat{\mathbf{u}}^h) + B^{\mathrm{WBC}}(\{\mathbf{w}^h, q^h\}, \{\mathbf{u}^h, p^h\}; \hat{\mathbf{u}}^h) - F(\{\mathbf{w}^h, q^h\}) = 0, \qquad (3.13)$$

where

$$\begin{aligned}
&B^{\mathrm{WBC}}(\{\mathbf{w}^h, q^h\}, \{\mathbf{u}^h, p^h\}; \hat{\mathbf{u}}^h) = \\
&- \sum_{b=1}^{N_{\mathrm{eb}}} \int_{\Gamma_t^b \cap (\Gamma_t)_{\mathrm{g}}} \mathbf{w}^h \cdot \boldsymbol{\sigma}\left(\mathbf{u}^h, p^h\right) \mathbf{n} \, \mathrm{d}\Gamma \\
&- \sum_{b=1}^{N_{\mathrm{eb}}} \int_{\Gamma_t^b \cap (\Gamma_t)_{\mathrm{g}}} \left(2\mu\boldsymbol{\varepsilon}\left(\mathbf{w}^h\right)\mathbf{n} + q^h\mathbf{n}\right) \cdot \left(\mathbf{u}^h - \mathbf{g}\right) \, \mathrm{d}\Gamma \\
&- \sum_{b=1}^{N_{\mathrm{eb}}} \int_{\Gamma_t^b \cap (\Gamma_t)_{\mathrm{g}}^-} \mathbf{w}^h \cdot \rho\left(\left(\mathbf{u}^h - \hat{\mathbf{u}}^h\right) \cdot \mathbf{n}\right)\left(\mathbf{u}^h - \mathbf{g}\right) \, \mathrm{d}\Gamma \\
&+ \sum_{b=1}^{N_{\mathrm{eb}}} \int_{\Gamma_t^b \cap (\Gamma_t)_{\mathrm{g}}} \tau_{\mathrm{B}} \mathbf{w}^h \cdot \left(\mathbf{u}^h - \mathbf{g}\right) \, \mathrm{d}\Gamma. \qquad (3.14)
\end{aligned}$$

Here $(\Gamma_t)_{\mathrm{g}}$ is the part of the boundary where the velocity boundary condition $\mathbf{g}$ is set, and $\mathbf{n}$ is the unit normal vector. The boundary $(\Gamma_t)_{\mathrm{g}}$ is decomposed into N_{eb} surface elements denoted by Γ_t^b, and $(\Gamma_t)_{\mathrm{g}}^-$ is defined as the "inflow" part of $(\Gamma_t)_{\mathrm{g}}$:

$$(\Gamma_t)_{\mathrm{g}}^- = \left\{ \mathbf{x} \;\middle|\; \left(\mathbf{u}^h - \hat{\mathbf{u}}^h\right) \cdot \mathbf{n} < 0, \; \forall \mathbf{x} \subset (\Gamma_t)_{\mathrm{g}} \right\}. \qquad (3.15)$$

If $(\Gamma_t)_{\mathrm{g}}$ coincides with the moving wall (rigid or flexible), then $\mathbf{g}$ is the prescribed wall velocity.

The importance of each term in the right-hand-side of Eq. (3.14) is as follows: The first term is the consistency term. It is necessary to ensure that the discrete formulation is identically satisfied by the exact solution of the Navier–Stokes equations,

which, in turn, has implications on the accuracy of the discrete formulation. Also note that this term cancels with the contributions coming from the integration-by-parts of the stress terms in Eq. (3.5), thus correctly removing traction boundary conditions from the no-slip boundary. The second term is the so-called adjoint consistency term. Its role is less intuitive, as it ensures that the analytical solution of the adjoint equations, when introduced in place of the linear momentum and continuity equation test functions, also satisfies the discrete formulation. Adjoint consistency is linked to optimal convergence of the discrete solution in lower-order norms (see, e.g., Arnold *et al.* [68]). The third term leads to better satisfaction of the inflow boundary conditions. The last term is penalty-like, in that it penalizes the deviation of the discrete solution from its prescribed value at the boundary. This term is necessary to ensure the stability of the discrete formulation, which may be lost due to the introduction of the consistency and adjoint consistency terms.

The weak boundary condition formulation is numerically stable if

$$\tau_{\mathrm{B}} = \frac{C_I^B \mu}{h_n}, \tag{3.16}$$

where h_n is the wall-normal element size, and C_I^B is a sufficiently large positive constant computed from an appropriate element-level inverse estimate (see, e.g., [87–89]). The constant C_I^B depends on the space dimension d, the element type (tetrahedron, hexahedron, etc.), and the polynomial order of the basis function. For example, it is sufficient to take $4.0 \leq C_I^B \leq 8.0$ to obtain a stable discrete solution for a linear tetrahedron. The wall-normal element size may be computed from the element metric tensor:

$$h_n = (\mathbf{n} \cdot \mathbf{G}\mathbf{n})^{1/2}. \tag{3.17}$$

Remark 3.3. Rather than setting the no-slip boundary conditions exactly, the weak boundary condition formulation gives the no-slip solution only in the limit as $h_n \to 0$. As a result, coarse discretizations do not need to struggle to resolve the boundary layers. The flow simply slips on the solid boundary. Because of this added flexibility, the weak boundary condition enforcement tends to produce more accurate results on meshes that are too coarse to capture the boundary layer solution. However, as the mesh is refined to capture the boundary layer, the weak and strong boundary condition formulations produce nearly identical results (see Bazilevs *et al.* [42]).

Remark 3.4. Although the weak boundary condition formulation is also stable for very large values of C_I^B, we do not favor that. Large values of C_I^B place a heavy penalization on the no-slip condition, and the above mentioned flexibility of the method is lost together with the associated accuracy benefits. We favor using a C_I^B that is just large enough to guarantee the stability of the discrete formulation.

3.2 Parallel Implementation

Turbulent flows, especially in the regime of large eddy simulation (LES), require substantial grid resolution for accuracy. Parallel computing is thus essential to efficiently compute turbulence. Although several references present applications of isogeometric analysis to turbulent flow, no discussion is given with regard to parallel implementation employed and parallel performance achieved by NURBS discretizations. For this reason, here, for the first time, we describe our parallel implementation of isogeometric analysis that allows us to achieve strong scalability on massively parallel computers.

There are mainly two stages that are involved in the computational work for standard finite elements or isogeometric analysis. The first stage consists of numerical integration over the elements and formation of the equation system (i.e. assembly of the left-hand-side tangent matrix and right-hand-side residual vector). The second stage consists of solving the linear system of equations using an iterative method. In this work we employ the Generalized Minimum Residual (GMRES) method [102].

Our parallel implementation [103] makes use of the Message Passing Interface (MPI) libraries and is adapted from Karanam *et al.* [104]. We use the non-overlapping element-based partitioning of the computational domain. The domain is partitioned into subdomains with similar work load as a preprocessing step, and each subdomain is individually assigned to a processor (or compute core). As a result, each element is uniquely assigned to a single partition and the assembly stage is highly scalable, both in terms of speed and memory.

However, element-based partitioning leads to shared control points[1] at the interprocessor boundaries. Figure 3.1(a) shows a computational domain that is partitioned

[1]Control points in isogeometric analysis are analogs of nodes in standard finite elements. In isogeometric analysis the degrees of freedom reside on control points and are referred to as control variables [28].

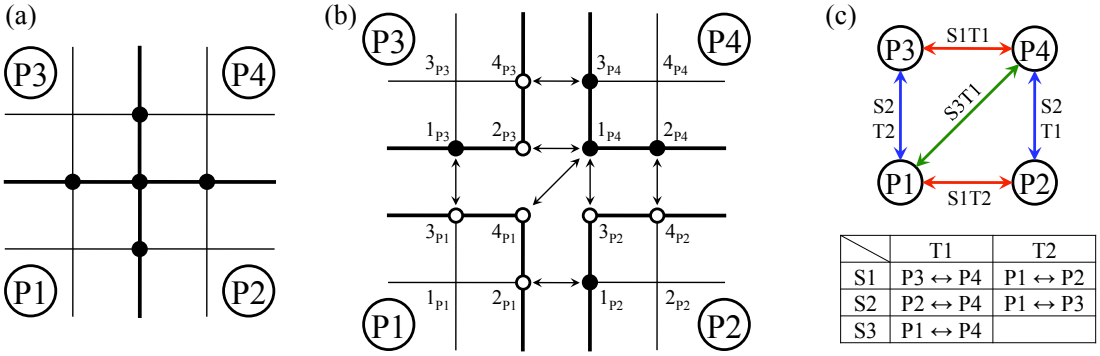


Figure 3.1: An example of the domain decomposition and parallel communications. (a) The computational domain is partitioned into four subdomains and each subdomain is individually assigned to a processor (P). Black dots are the control points shared by the subdomains. (b) The shared control points are designated to be "masters" (black dots) or "slaves" (white dots) in each subdomain. The arrows indicate the correspondence between the master and slave control points. (c) Communication stages (S) and tasks (T) for this example.

into four subdomains. Black dots are the control points shared by the subdomains, while the rest of the control points reside solely in their subdomains and are not shared. Typically, a good balance of elements with sufficient work load in each partition and a minimum amount of communication between the partitions result in a reasonable control-point balance as well. This also helps maintain good scalability in solving the resultant linear system of equations using an iterative solver.

Each subdomain will generally have shared control points requiring inter-processor communication, as shown in Figure 3.1(b). The shared control points are the points that reside on different partitions that originally referred to the same global control point number in the non-partitioned domain. In a set of control points that referred to the same global control point number, only one is designated to be a "master" and the rest are "slaves". Figure 3.1(b) shows an example where the black dots are the master control points and the white dots are the slave control points. The arrows indicate the correspondence between the master and slave control points.

Every partition contains information about its portion of the work and its interaction with neighboring partitions. The interaction between neighboring partitions is defined based on shared control points. The design is such that only the master control point will be in charge of data accumulation and update. As a result, the communica-

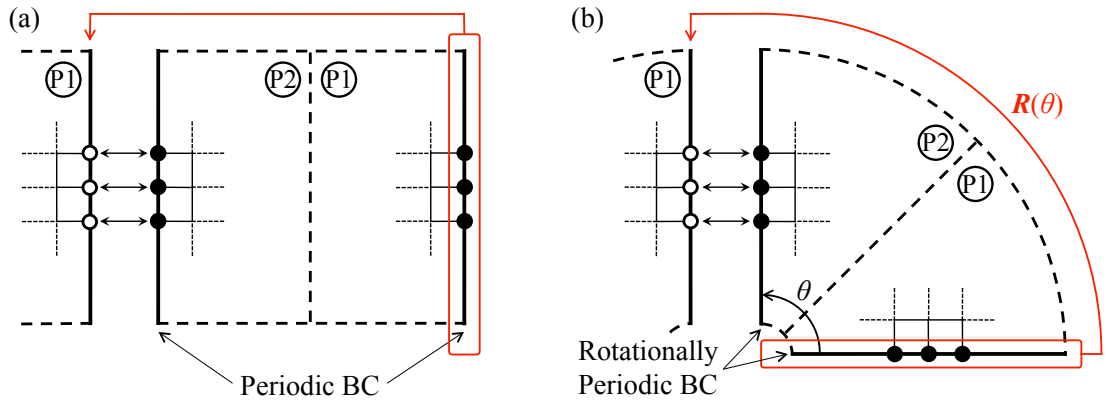


Figure 3.2: Illustration of parallelization procedures for (a) periodic boundaries, including (b) boundaries with rotational periodicity. Here $\mathbf{R}(\theta)$ is the rotation matrix for the rotationally periodic boundary conditions.

tions will only exist between master and slave control points. No communications are needed between slave control points.

The process of communicating the information to and from the master control points is decomposed into so-called communication stages. Each stage involves several communication tasks, which are one-to-one communications between two unique processors. During a communication task the data for the shared control points on a given processor is packaged into a single array (using MPI derived data types) and sent (or received) as a single message. Figure 3.1(c) illustrates the communication stages and tasks for the example shown. Note that no communication will take place between processors 2 and 3, because the share control point is designated to be a slave on these processors. The master–slave relationships and the communication stages and tasks are defined in the pre-processing step.

The master–slave framework is also well suited for handling periodic boundaries as shown in Figure 3.2(a). For rotationally periodic boundary conditions, in addition to defining the necessary inter-processor communication structures, further operations are needed and are discussed in what follows.

There are two major types of communications. The first type is where the data is accumulated at the master control points from their slave control points to obtain a "complete value". A control point is said to have a complete value if it has the same value in the partitioned case as in the unpartitioned case [105]. For example, after nu-

merical integration on each local partition, non-shared control points will have complete values in the right-hand-side vector. However, values for shared control points are incomplete on each subdomain because their contributions are distributed among the partitions that shares the same control point. Communications between processors are needed in this situation. Slave control points will send and add their incomplete values to the corresponding master control points. After finishing all the communication stages, the master control points have complete values, while the slave control points are assigned zero values for the right-hand-side vector.

The second type of the communication is where the complete values are copied from the masters to update their slave control points. For example, after solving the linear system, the master control points have complete updated value of the solutions. Communications will then take place to send back these solutions from the master control points to their corresponding slave control points. As a result, the slave control points will also have the complete value of the solutions.

For rotationally periodic boundaries the complete value of the right-hand-side vector is obtained by first rotating the linear momentum residual at the slave control points to the correct coordinate system, and then sending and adding the contributions to the master control points. Likewise, when the value of the velocity vector is computed on the master control points, it is first rotated back and then sent to the slave control points. In order to rotate the right-hand-side and solution vectors, we construct an appropriate rotation matrix at the rotationally periodic control points. Figure 3.2(b) illustrates this case. Note that no rotation is necessary to communicate the scalar pressure variable or the continuity equation residual at the rotationally periodic boundaries.

To solve the linear system using GMRES, only the product of the left-hand-side matrix with a right-hand-side vector is needed. To efficiently compute this product in parallel, a vector, which is assembled at the processor level and communicated globally, is multiplied against a local-to-the-processor assembled matrix in parallel. To assess convergence and decide whether to continue with the iteration process or update the solution, a new communication is performed.

Note that for a fixed-size problem partitioned into large number of subdomains, the number of elements per partition becomes relatively small. This may result in sig-

nificant imbalance of control points since the control point balance is not explicitly requested in the procedure. Furthermore, the percentage of shared control points increases and communication stages and tasks may also increase. All these may eventually becomes detrimental to scaling.

3.3 Aerodynamic Simulations of the NREL 5 MW Offshore Baseline Wind Turbine Rotor

In this section we present the aerodynamic simulations of the NREL 5 MW Offshore baseline wind turbine rotor [4] using the ALE–VMS method and NURBS-based isogeometric analysis. The NREL 5 MW offshore baseline wind turbine is a conventional three-bladed upwind turbine with a rotor diameter of 126 m. It was first proposed by Jonkman *et al.* [4] to support concept studies aimed at assessing offshore wind technology. The detailed geometry description and construction for the blade surface and the surrounding fluid domain volume were presented in Section 2.2.

3.3.1 Rotationally Periodic Boundary Conditions

As mentioned earlier in Section 2.2.2, to enhance the efficiency of the simulations, we take advantage of the problem rotational symmetry and only construct a 120° slice of the computational domain. However, to carry out the simulations, rotationally periodic boundary conditions must be imposed. Let $\mathbf{u}_l^h$ and $\mathbf{u}_r^h$ denote the discrete fluid velocities at the left and right boundary, respectively, and p_l^h and p_r^h denote the corresponding pressures (see Figure 3.3), we set

$$\mathbf{u}_l^h = \mathbf{R}\left(\frac{2}{3}\pi\right)\mathbf{u}_r^h, \tag{3.18}$$

and

$$p_l^h = p_r^h, \tag{3.19}$$

where $\mathbf{R}(2/3\pi)$ is the rotation matrix evaluated at $\alpha = 2/3\pi$. That is, while the pressure degrees-of-freedom take on the same values, the fluid velocity degrees-of-freedom

are related through a linear transformation corresponding to a rotation by $2/3\pi$ radians. Note that the transformation matrix is independent of the current domain position. Rotationally-periodic boundary conditions are implemented through standard master-slave relationships. During the discrete residual assembly, the nodal or control-point momentum residual at the slave boundary is rotated using $\mathbf{R}^T(2/3\pi)$ and added to the master residual (assumed to reside at the right boundary). Once the fluid solution update is computed, the velocity solution at the master boundary is rotated using $\mathbf{R}(2/3\pi)$ prior to communicating it to the slave boundary.

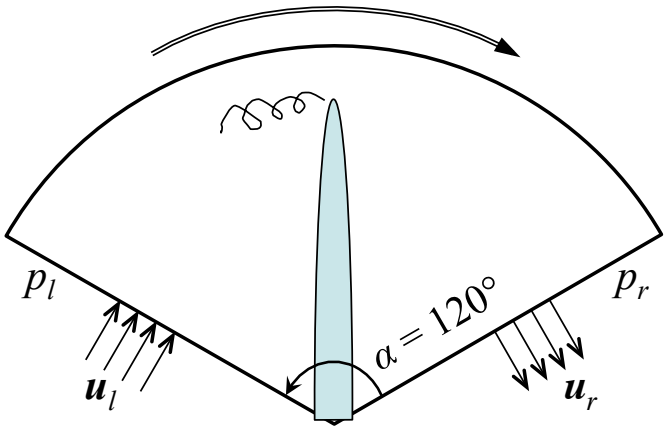


Figure 3.3: Rotationally periodic boundary conditions.

3.3.2 Simulations with Prescribed Rotor Speed

Case I: 9 m/s Wind Speed and 1.08 rad/s Rotor Speed

We compute the aerodynamics of the wind turbine rotor with prescribed speed using a rotating mesh. The wind speed is uniform at 9 m/s and the rotor speed is 1.08 rad/s, giving a tip speed ratio of 7.55 (see Spera [106] for wind-turbine terminology). The air density and viscosity are 1.2 kg/m^3 and 2.0×10^{-5} kg/(m s), respectively. At the inflow boundary the velocity is set to the wind velocity, at the outflow boundary the stress vector is set to zero, and at the radial boundary the radial component of the velocity is set to zero. We start from a flow field where the velocity is equal to the inflow velocity everywhere in the domain except on the rotor surface, where the velocity

matches the rotor velocity. We carry out the computations at a constant time-step size of 4.67×10^{-4} s.

The chosen wind velocity and rotor speed correspond to one of the cases given in Jonkman *et al.* [4], where the aerodynamics simulations were performed using FAST [3]. FAST (Fatigue, Aerodynamics, Structures, and Turbulence) is an aeroelastic code developed by National Renewable Energy Laboratory (NREL) for predicting the dynamics and performance of the wind turbine system. The aerodynamic loads are computed using the blade element momentum (BEM) method. The code is based on look-up tables for airfoil cross-sections, which give planar, steady-state lift and drag data for a given wind speed and angle of attack. The effects of trailing edge turbulence, hub, and tip are incorporated through empirical models, and therefore the use of FAST in predictive simulations is restricted to a narrow class of cases.

It was reported in Jonkman *et al.* [4] that at these wind conditions and rotor speed, no blade pitching takes place and the rotor develops a favorable aerodynamic torque (i.e., torque in the direction of the rotation) of 2500 kN m. Although this value is used for comparison with our simulations, the exact match is not expected, as our computational modeling is very different than the one in Jonkman *et al.* [4]. Nevertheless, we feel that this value of the aerodynamic torque is close to what is expected in reality, given the vast experience of NREL with wind turbine rotor simulations employing FAST.

The volumetric NURBS mesh of the computational domain and the problem setup are shown in Figures 2.5 and 3.4, respectively. The mesh of the 120° slice of the domain is comprised of 1,449,000 quadratic NURBS elements, which yields about the same number of mesh control points (analogues of nodes in finite elements). Figure 3.5 shows 2D blade cross-sections at 60% and 75% spanwise stations to illustrate the boundary layer mesh used in this computation. Near the blade surface, the size of the first element in the wall-normal direction is about 2 cm.

The computation is carried out on 240 compute cores on Ranger Linux Cluster at the Texas Advanced Computing Center [107]. The GMRES search technique [102] is used with a block-diagonal preconditioner. Each nodal block consists of a 3×3 and 1×1 matrices, corresponding to the discrete momentum and continuity equations, respec-

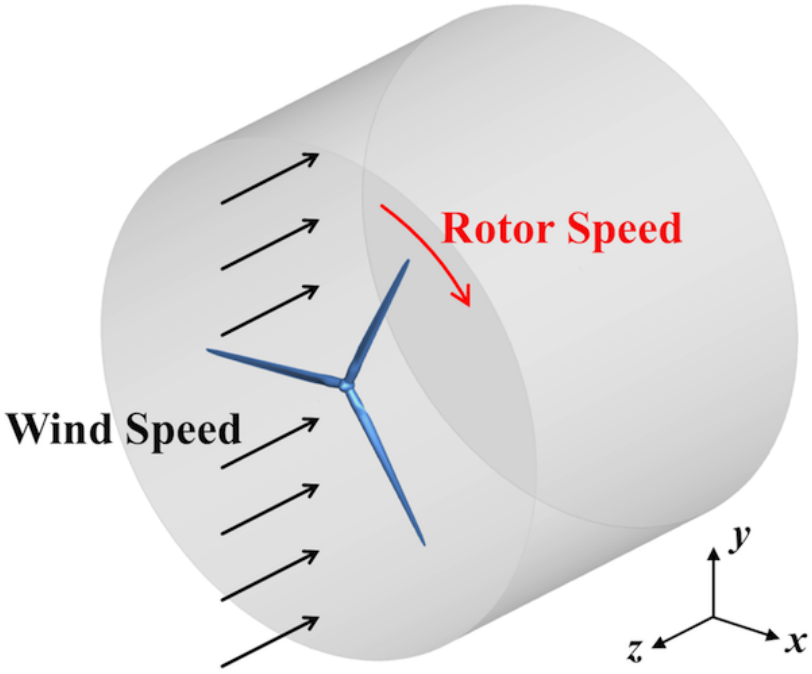
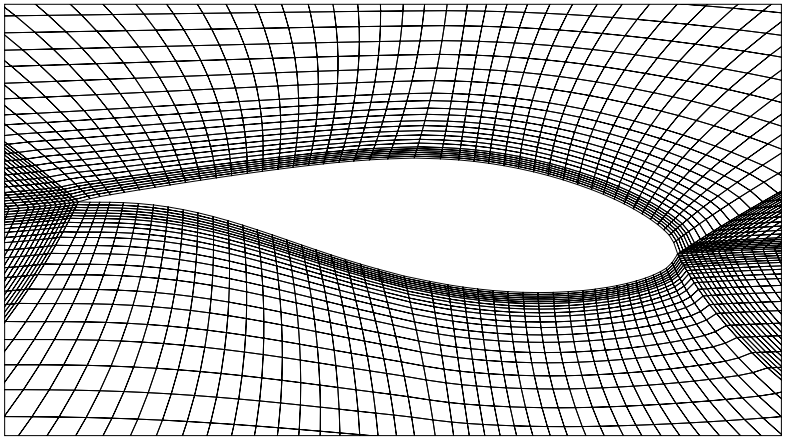


Figure 3.4: Problem setup with prescribed inflow wind velocity and rotor speed.

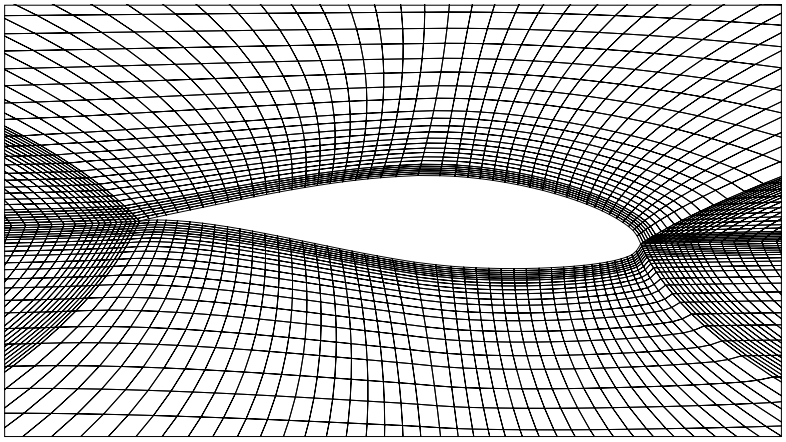
tively. The number of nonlinear iterations per time step is 4 and the number of GMRES iterations is 200 for the first Newton iteration, 300 for the second, and 400 for the third and fourth.

Figure 3.6 shows the air speed at $t = 0.8$ s at 1 m behind the rotor plane. Note the fine-grained turbulent features at the trailing edge of the blade, which require enhanced mesh resolution for accurate representation. Figure 3.7 shows the air pressure contours at the suction side (i.e., the back side) of the wind turbine blade. It is precisely the large negative pressure that creates the desired lift. The fluid traction vectors projected to the plane of rotation are shown in Figure 3.8. The traction vectors point in the direction of rotation and grow in magnitude toward the blade tip, creating favorable aerodynamic torque. However, at the blade tip the traction vectors rapidly decay to zero and even change sign, which introduces a small amount of inefficiency.

The time history of the aerodynamic torque is shown in Figure 3.9, where the steady-state result from Jonkman *et al.* [4] is also shown for reference. The figure shows that in less than 0.8 s the torque settles at a statistically-stationary value of 2,670 kN m, which is within 6.4% of the reference. Given the significant differences in the com-



(a) at 60% spanwise station



(b) at 75% spanwise station

Figure 3.5: 2D blade cross-sections at 60% and 75% spanwise stations to illustrate the boundary layer mesh used in our computation. For the purpose of visualization, each quadratic NURBS element is interpolated with $2 \times 2 \times 2$ bilinear elements.

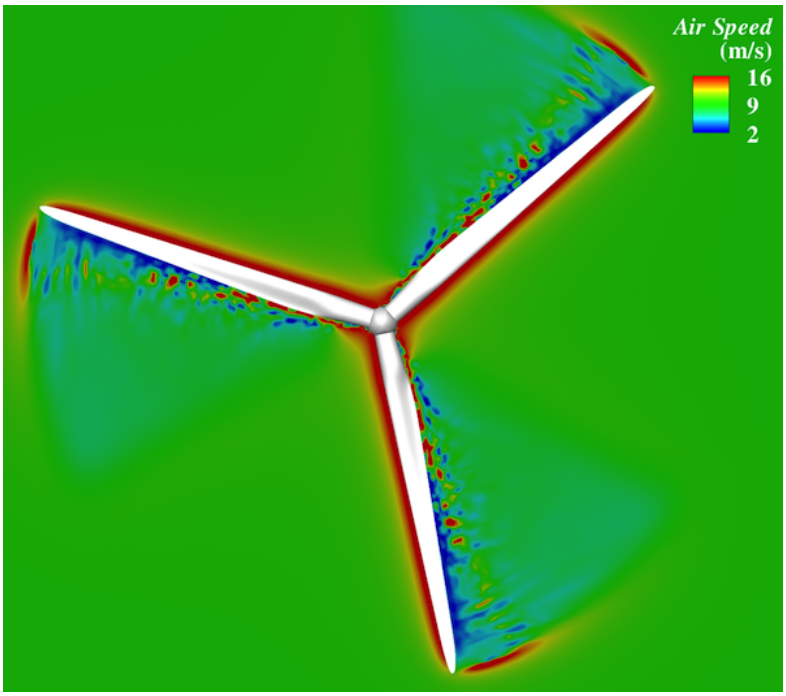


Figure 3.6: Air speed at $t = 0.8$ s.

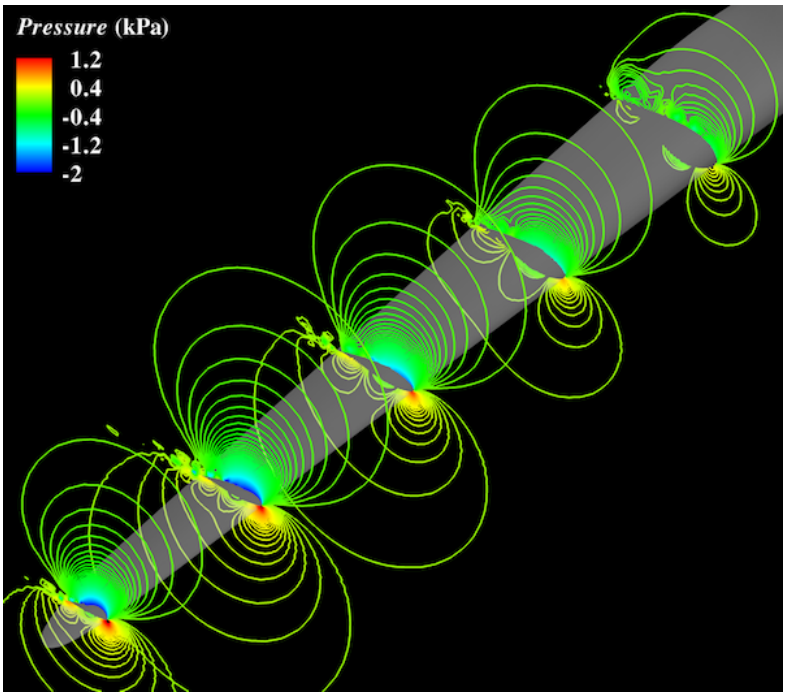


Figure 3.7: Pressure contours at several blade cross-sections at $t = 0.8$ s viewed from the back of the blade. The large negative pressure at the suction side of the airfoil creates the desired lift.

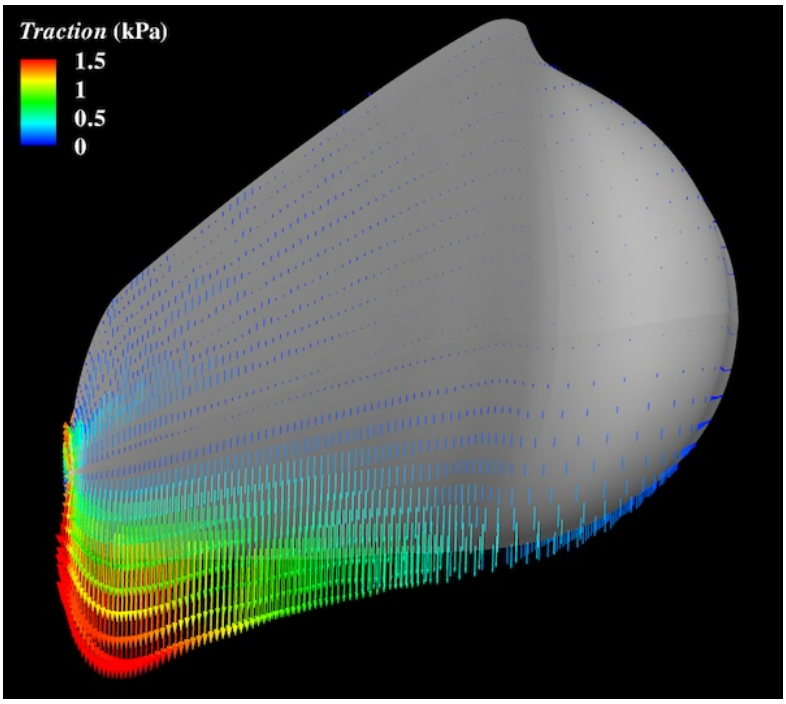


Figure 3.8: Fluid traction vectors at $t = 0.8$ s viewed from the back of the blade. The traction vectors, colored by magnitude, are projected to the rotor plane and illustrate the mechanism by which the favorable aerodynamic torque is created.

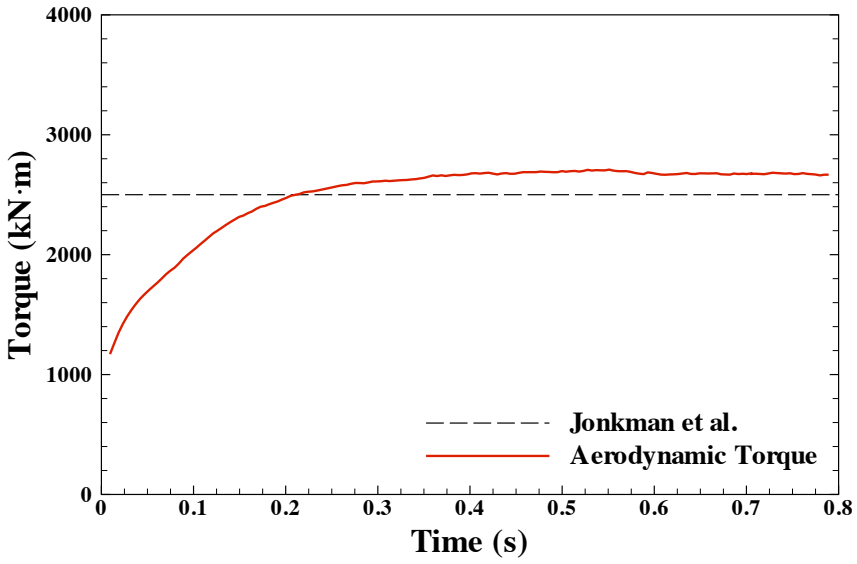


Figure 3.9: Time history of the aerodynamic torque. Statistically-stationary torque is attained in less than 0.8 s. The reference steady-state result from NREL is also shown for comparison.

putational modeling approaches, the two values are remarkably close. This result is encouraging in that 3D time-dependent simulation with a manageable number of degrees of freedom and without any empiricism is able to predict important quantities of interest for wind-turbine rotors simulated at full scale. This result also gives us confidence that our procedures are accurate and may be applied to simulations cases where 3D, time-dependent modeling is indispensable (e.g., simulation of wind gusts or blade pitching).

Given the aerodynamic torque and the rotor speed, the power extracted from the wind with these wind conditions (based on our simulations) is

$$P = T_f \dot{\theta} \approx 2.88 \text{ MW}. \tag{3.20}$$

According to the Betz law (see, e.g., Hau [108]), the maximum power that a horizontal-axis wind turbine is able to extract from the wind is

$$P_{max} = \frac{16}{27} \frac{\rho A \|\mathbf{u}_{in}\|^3}{2} \approx 3.23 \text{ MW}, \tag{3.21}$$

where $A = \pi R^2$ is the cross-sectional area swept by the rotor, and $\|\mathbf{u}_{in}\|$ is the inflow speed. From this we conclude that the wind turbine aerodynamic efficiency at the simulated wind conditions is

$$\frac{P}{P_{max}} \approx 89\%, \tag{3.22}$$

which is quite high even for modern wind turbine designs.

The blade is segmented into 18 spanwise "patches" to investigate how the aerodynamic torque distribution varies along the blade span. The patch-wise torque distribution is shown in Figure 3.10. The torque is nearly zero in the cylindrical section of the blade. A favorable aerodynamic torque is created on Patch 4 and its magnitude continues to increase until Patch 15. The torque magnitude decreases rapidly after Patch 15, however, the torque remains favorable all the way to the last patch.

The importance of 3D modeling and simulation is further illustrated in Figure 3.11, where the axial component of the flow velocity is displayed at a blade cross-section located at $0.889R$ (56 m above the rotor center). The magnitude of the axial velocity component exceeds 15 m/s in the boundary layer, showing that 3D effects are important, especially in the regions of the blade with the largest contribution to the aerodynamic torque.

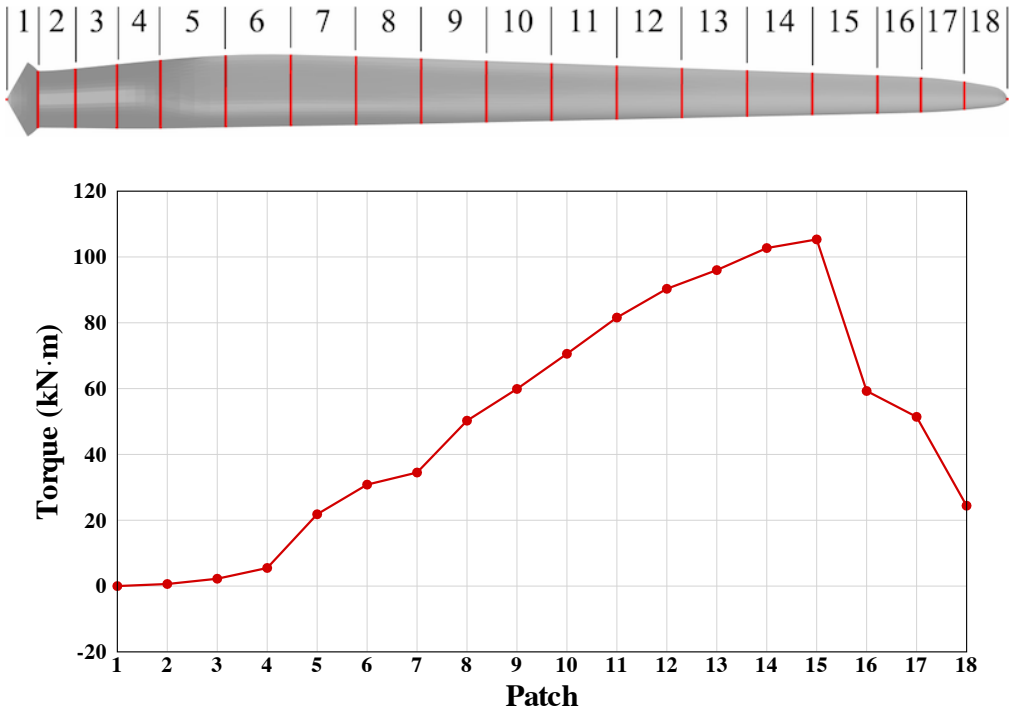


Figure 3.10: Patches along the blade (top) and the aerodynamic torque contribution from each patch (bottom) at $t = 0.8$ s.

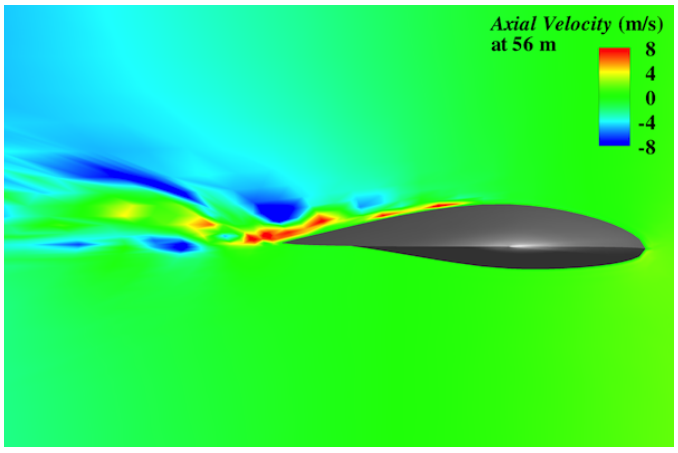


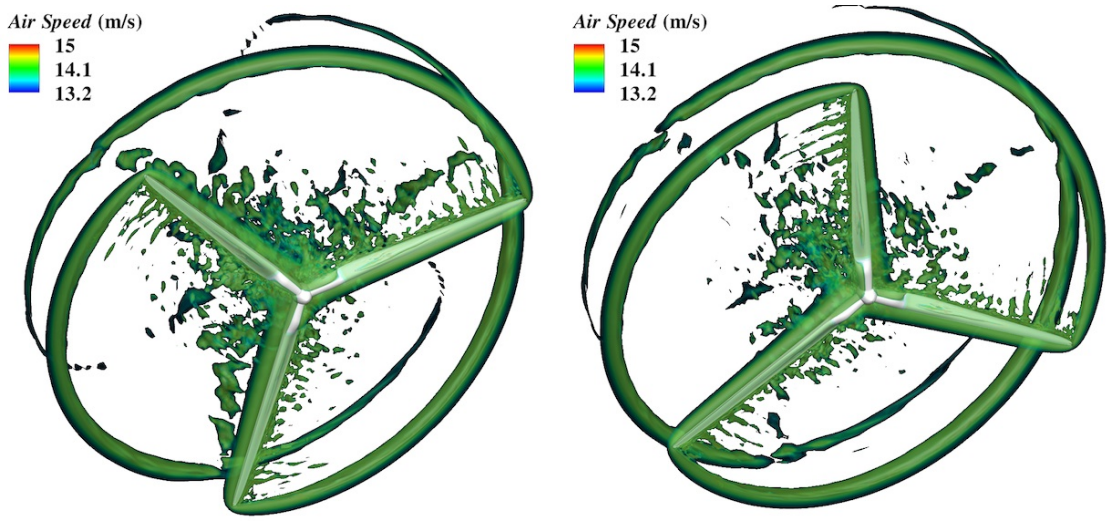
Figure 3.11: Axial flow velocity over the blade cross-section at 56 m at $t = 0.8$ s. The level of axial flow in the boundary layer is significant, which illustrates the importance of 3D modeling.

Case II: 11.4 m/s Wind Speed and 1.267 rad/s Rotor Speed

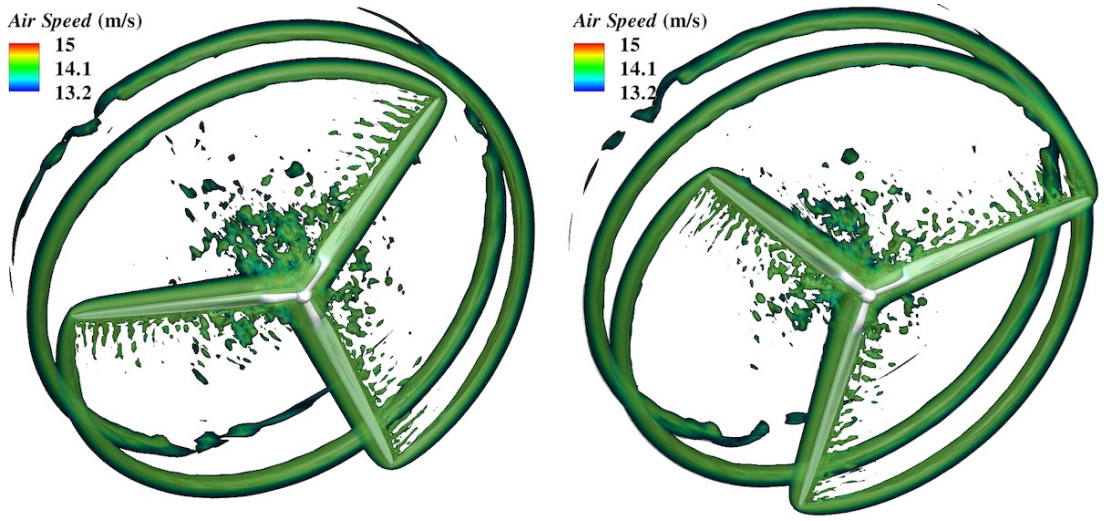
For the second case, the wind turbine rotor is simulated at prescribed steady inlet wind speed of 11.4 m/s and rotor angular velocity of 1.267 rad/s. This setup corresponds to the case of the highest wind speed under 0° blade pitch reported in Jonkman *et al.* [4]. When the wind speed is higher than 11.4 m/s, the blade is pitched such that the rotor rotates at a constant rotational speed of 12.1 rpm (see Jonkman *et al.* [4] for more detail).

Isosurfaces of the air speed at a time instant is shown in Figure 3.12. The vortex forming at the tip of the blades is convected downstream of the rotor with little decay, which is attributable to the use of NURBS functions. The time history of the aerodynamic torque is plotted in Figure 3.13. The result compares favorably to the data reported by Jonkman *et al.* [4] obtained using FAST [3]. As mentioned earlier, computations using FAST highly rely on the choice of airfoil lift and drag data, and empirical corrections of 2D airfoil results to account for 3D effects. In our simulations we are able to capture the aerodynamic torque using the proposed procedures, which do not rely on empiricism and are 100% predictive.

We also assess the scalability of our aerodynamics code on the wind turbine simulation. The parallel scalability tests were carried out on Ranger, a Sun Constellation Linux Cluster at the Texas Advanced Computing Center (TACC) [107]. The Ranger system is comprised of 3936 16-way SMP compute nodes providing 15744 AMD Opteron^{TM} processors for a total of 62976 compute cores, 123 TB of total memory and 1.7 PB of raw global disk space. It has a theoretical peak performance of 579 TFLOPS. All Ranger nodes are interconnected using InfiniBand technology in a full-CLOS topology providing a 1GB/sec point-to-point bandwidth [109]. The wind turbine mesh is decomposed into 60, 120, 240 and 480 subdomains as shown in Figure 3.14. Five and six partitions were created in the downstream and radial directions, respectively, for all cases. In the azimuthal direction the mesh is decomposed into 2, 4, 8 or 16 partitions. The scalability test results are shown in Figure 3.15. Near-perfect linear parallel scaling is achieved.



(a) t = 1.345 s (b) t = 1.895 s



(c) t = 2.445 s (d) t = 2.995 s

Figure 3.12: Isosurfaces of air speed at several instants in the wind turbine simulation. The flow exhibits complex behavior. The vortical feature generated at the blade tip is convected downstream of the rotor with very little decay.

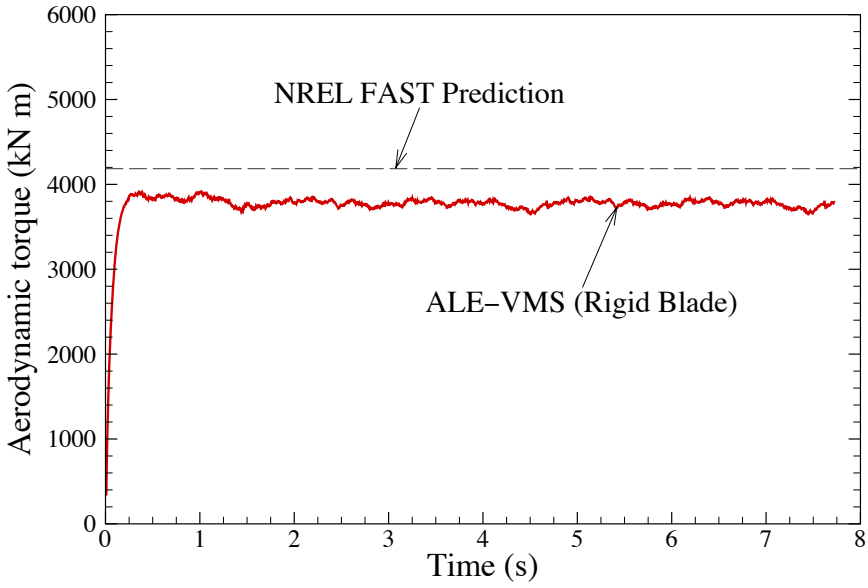
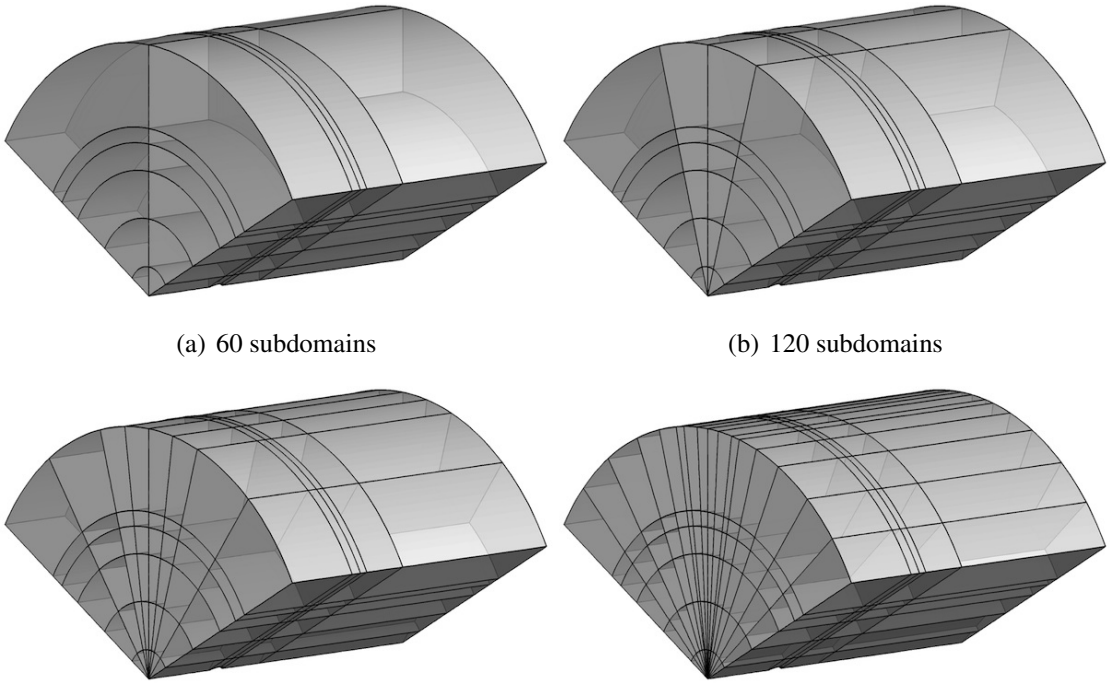


Figure 3.13: Time history of the aerodynamic torque. The reference result reported by Jonkman *et al.* [4] is also shown for comparison.



(a) 60 subdomains

(b) 120 subdomains

(c) 240 subdomains

(d) 480 subdomains

Figure 3.14: Scalability study for wind turbine rotor simulation. The computational domain is decomposed into 60, 120, 240 and 480 subdomains.

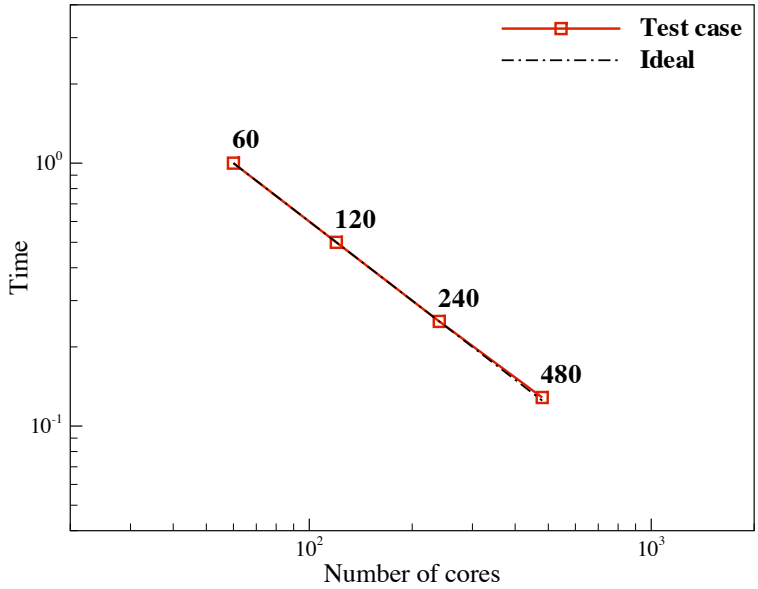


Figure 3.15: Scalability study for wind turbine rotor simulation. The computation time is normalized by the result of 60-processor case.

3.3.3 Coupling of a Rigid Rotor with Incompressible Flow

In this section we present the formulation of the coupled problem that involves incompressible fluid and a rigid body that undergoes rotational motion. We specialize the formulation to the case of a wind turbine rotor whose axis of rotation coincides with the z-direction. We denote by θ the angle of rotation of the wind turbine rotor about the z-axis from the reference configuration. The rotor angular velocity and acceleration are found by differentiating θ with respect to time, and are denoted by $\dot{\theta}$ and $\ddot{\theta}$, respectively. The coupled problem amounts to finding the triple $\{\mathbf{u}^h, p^h, \theta\}$ such that for all test functions $\{\mathbf{w}^h, q^h\}$

$$B^{\mathrm{VMS}}(\{\mathbf{w}^h, q^h\}, \{\mathbf{u}^h, p^h\}) - F(\{\mathbf{w}^h, q^h\}) = 0, \tag{3.23}$$

$$I\ddot{\theta} + f(\dot{\theta}, t) - T_f(\mathbf{u}^h, p^h) = 0, \tag{3.24}$$

where I is the rotor moment of inertia, $f(\dot{\theta}, t)$ may represent forcing due to friction in the system or interaction with the generator, and $T_f(\mathbf{u}^h, p^h)$ is the torque exerted on the rotor by the fluid, i.e. the aerodynamic torque. This is a two-way coupled problem: the fluid solution depends on the rotor position and speed, while the rotor equation is driven by the fluid torque.

The fluid domain rotates with the same angular speed as the rotor. This means the fluid domain current configuration is defined as

$$\Omega = \{\mathbf{x} \mid \mathbf{x} = \mathbf{R}(\theta)[\mathbf{X} - \mathbf{X}_0] + \mathbf{X}_0,\ \mathbf{X} \in \Omega_0\}, \tag{3.25}$$

where Ω_0 is the fluid domain in the reference configuration, $\mathbf{X}_0$ is the rotor center of mass, and $\mathbf{R}(\theta)$ is the rotation matrix given by

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.26}$$

With the above definition of the fluid domain motion, the fluid domain velocity becomes

$$\hat{\mathbf{u}} = \frac{\mathrm{d}\mathbf{R}(\theta)}{\mathrm{d}\theta}[\mathbf{X} - \mathbf{X}_0]\ \dot{\theta}. \tag{3.27}$$

The aerodynamic torque is given by

$$T_f(\mathbf{u}, p) = \int_{\Gamma_{\mathrm{rot}}} \mathbf{r} \times (p\mathbf{n} - 2\mu\boldsymbol{\varepsilon}(\mathbf{u})\,\mathbf{n})\ \mathrm{d}\Gamma, \tag{3.28}$$

where Γ_{rot} is the rotor surface, $\mathbf{r} = \mathbf{x} - \mathbf{x}_0$ is the radius-vector from the rotor center of mass in the current configuration, and $\mathbf{n}$ is the unit normal to the rotor surface. In the computations, instead of using Eq. (3.28) directly, we opt for a conservative definition of the aerodynamic torque. For this, we first define the discrete momentum equation residual at every finite element node or isogeometric control point A and cartesian direction i as

$$R_{A,i} = B^{\mathrm{VMS}}(\{N_A\mathbf{e}_i, 0\}, \{\mathbf{u}^h, p^h\}) - F(\{N_A\mathbf{e}_i, 0\}), \tag{3.29}$$

where N_A's are the velocity basis functions and $\mathbf{e}_i$'s are the Cartesian basis vectors. The residual $R_{A,i}$ vanishes everywhere in the domain except at the rotor surface control points (or nodes). The conservative aerodynamic torque along the z-axis may now be computed as

$$T_f(\boldsymbol{v}^h, p^h) = \sum_{A,j,k} \epsilon_{3jk} r_{A,j} R_{A,k}, \tag{3.30}$$

where ϵ_{ijk}'s are the cartesian components of the alternator tensor and $r_{A,j}$ are the nodal or control point coordinates.

The coupled system given by Eqs. (3.23) and (3.24) is advanced in time using the Generalized-α method (see Chung and Hulbert [110], Jansen *et al.* [111], and Bazilevs *et al.* [56]) for both the fluid and rotor equations. Within each time step, the coupled equations are solved using an inexact Newton approach. For every Newton iteration we obtain the fluid solution increment, update the fluid solution, compute the aerodynamic torque, obtain the rotor solution increment, and update the rotor solution and the fluid domain velocity and position. The iteration is repeated until convergence. Because the wind turbine rotor is a relatively heavy structure, the proposed approach, also referred to as "block-iterative" (see Tezduyar *et al.* [112] for the terminology), is stable.

Remark 3.5. The case of prescribed angular velocity is also of interest and can be obtained by providing a time history of θ, which obviates Eq. (3.24). We make use of this in the sequel, where we perform a validation study of our wind turbine rotor configuration.

Remark 3.6. We note that the torque definition given by Eq. (3.30) makes use of the isoparametric property of the finite element and isogeometric methods. Besides the possible accuracy benefits associated with a conservative definition of torque, the formulation avoids performing boundary integration and makes use of the same "right-hand-side" routine as the fluid assembly itself, which simplifies implementation.

Remark 3.7. The conservative torque definition given by Eq. (3.30) is only applicable for strongly enforced velocity boundary conditions, which are employed in this work for wind turbine simulations. For the case of weak boundary conditions, a conservative definition of the torque given in Bazilevs and Akkerman [46] may be used instead.

We perform a coupled aerodynamic analysis of the same NREL 5 MW offshore baseline wind turbine using a NURBS discretization. The wind speed at the inflow is now set to 12 m/s. Jonkman *et al.* [4] reports that at this wind speed the rotor blades are pitched to prevent over-rotation. Our interest was to study a scenario in which no blade pitching controls are applied and the rotor is free to spin under the influence of incoming wind. This situation may occur due to the failure of the rotor control system, in which case, under the prescribed wind condition, the rotor will spin at speeds much faster than the limits set by the design strength of the rotor structure.

The simulation is started with the rotor spinning at a prescribed angular velocity of 0.8 rad/s. Then the rotor is released to spin freely under the action of wind forces. We neglect the frictional losses and interaction with the generator, that is, we set $f(\dot{\theta}, t) = 0$ in Eq. (3.24). Figure 3.16 shows the time history of the rotor angular speed and acceleration. Within 1 s the rotor attains an angular velocity of 3 rad/s. The computation was stopped at this point, however, the rotor continues to accelerate. Due to the size and the dimension of the blade, such rotor speeds lead to unsustainable structural loads and, ultimately, wind turbine failure.

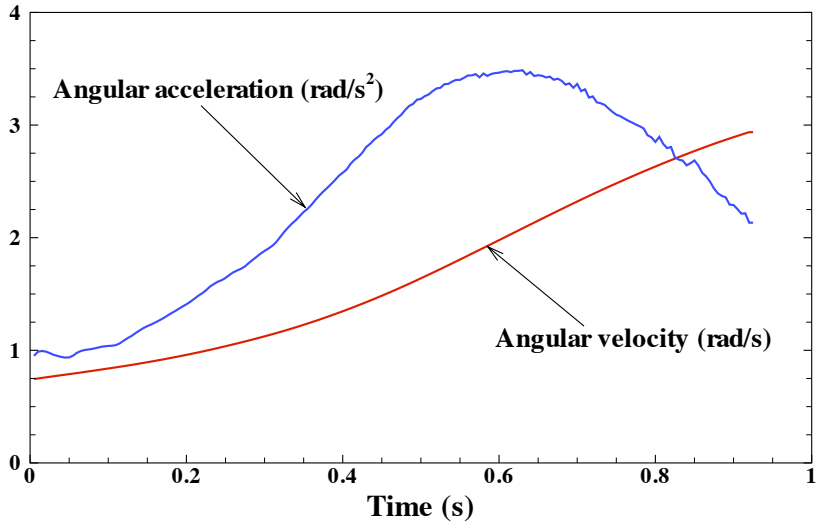


Figure 3.16: Time history of the rotor angular velocity and acceleration.

Figure 3.17 shows a snapshot of the rotor spinning at nearly 3 rad/s and the isosurfaces of the air speed. The trailing edge of the blade and especially the tip create significant amount of turbulence. Snapshots of air pressure on the front and back of the blade are shown in Figure 3.18. Note that, compared to the previous case of 9 m/s wind and 1.08 rad/s rotation, the pressure magnitude is significantly higher. Furthermore, the pressure field at the trailing edge and the blade tip exhibits fluctuating behavior, which leads to high-frequency loading and aerodynamic noise generation in these areas.

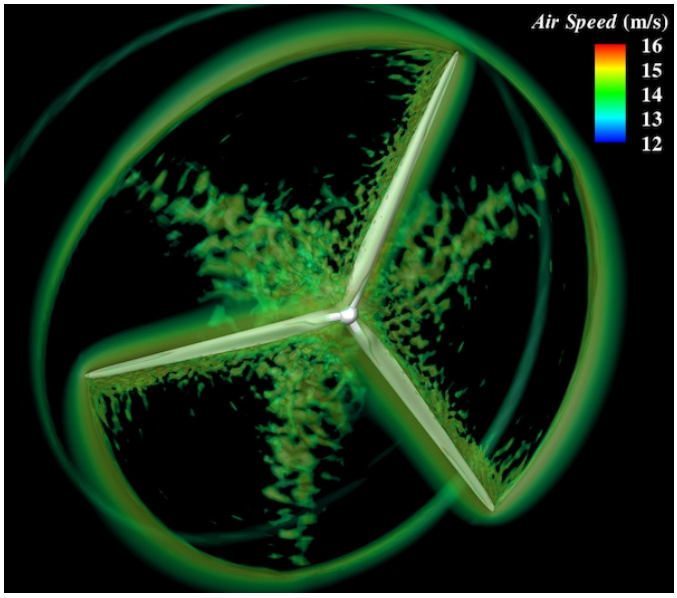
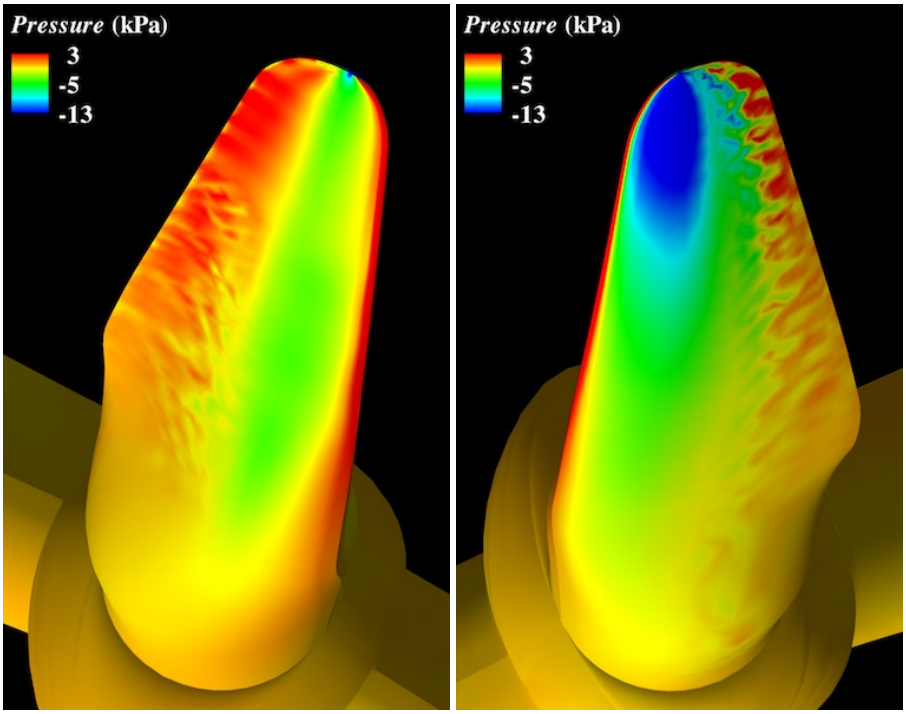


Figure 3.17: Isosurfaces of air speed at t = 0.8 s.



(a) Front (b) Back

Figure 3.18: Air pressure distribution on the blade surface.

3.4 Validation Study using NREL Phase VI Experiment

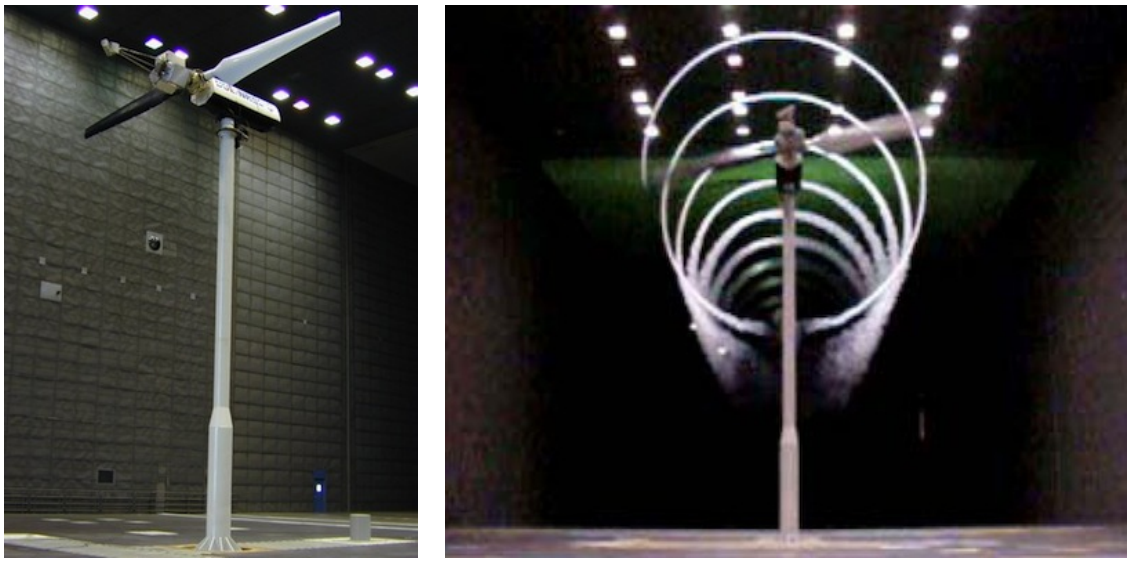
We simulated the NREL 5 MW offshore baseline wind turbine rotor in Section 3.3, and compared the results with those reported in Jonkman *et al.* [4]. Although good matching of the aerodynamic torque was obtained in the case of steady inflow and rotor speed conditions, this was not, strictly speaking, validation, since the comparison was made with another numerical prediction.

In this section, we present a true validation study where we simulate the NREL Unsteady Aerodynamics Experiment (UAE) Phase VI wind turbine [113], and compare our results with the existing experimental data for this case. This two-bladed experimental wind turbine has a rotor diameter of 10.058 m, hub height of 12.192 m, and a rated power of 19.8 kW. It was tested in the NASA Ames 80 ft $\times$ 120 ft wind tunnel in 2000 (see Figure 3.19), and is considered to be one of the most comprehensive, accurate, and reliable experiments carried out on a full-scale wind turbine. The measured quantities were studied by many computational researchers (see, e.g., [5–10, 15, 114–121] for the purposes of validating their simulation software and improving their ability to predict wind turbine aerodynamics.

The Phase VI rotor geometry makes use of a single NREL S809 airfoil [122]. Selected blade cross-section geometry data are summarized in Table 3.1, and the detailed documentation of the rotor configuration and its technical specifications are available in Hand *et al.* [113].

Table 3.1: Selected blade cross-section geometry data for the NREL UAE Phase VI rotor. For the complete table, see Hand *et al.* [113] and Jonkman [116].

Radial Distance r (m)	Span Station (r/5.029 m)	Chord Length (m)	Twist Angle (degrees)	Twist Axis (% chord)	Airfoil Profile (-)
0.508	0.100	0.218	0.0	50	Cylinder
1.510	0.300	0.711	14.292	30	NREL S809
2.343	0.466	0.627	4.715	30	NREL S809
3.185	0.633	0.542	1.115	30	NREL S809
4.023	0.800	0.457	-0.381	30	NREL S809
4.780	0.950	0.381	-1.469	30	NREL S809
5.029	1.000	0.355	-1.815	30	NREL S809



(a) (b)

Figure 3.19: (a) NREL UAE Phase VI wind turbine in the NASA Ames 80 ft × 120 ft wind tunnel. (b) Wake flow visualization of the operating turbine in the wind tunnel. The images are taken from Hand *et al.* [113].

3.4.1 Finite element simulation of wind turbine aerodynamics

In contrast to Section 3.3, where isogeometric analysis was used for aerodynamics computations, the ALE–VMS equations are discretized using linear finite elements in space for the purpose of assessing the accuracy of the ALE–VMS methodology in predicting wind turbine aerodynamics. Also note that a full domain of this two-bladed rotor without rotationally periodic boundary conditions is considered. This choice was motivated by the fact that we are moving toward simulating the full wind turbine assembly, including the tower and nacelle, for which rotationally periodic boundary conditions are no longer applicable.

For this validation study, wind speeds of 5, 7, 10, 15, 20 and 25 m/s were selected from the sequence S of the NREL Phase VI experiment [113]. The experimental sequence S setup consists the wind turbine rotor in the upwind configuration, 0° yaw angle, 0° cone angle, rotational speed of 72 rpm, and blade tip pitch angle of 3°.

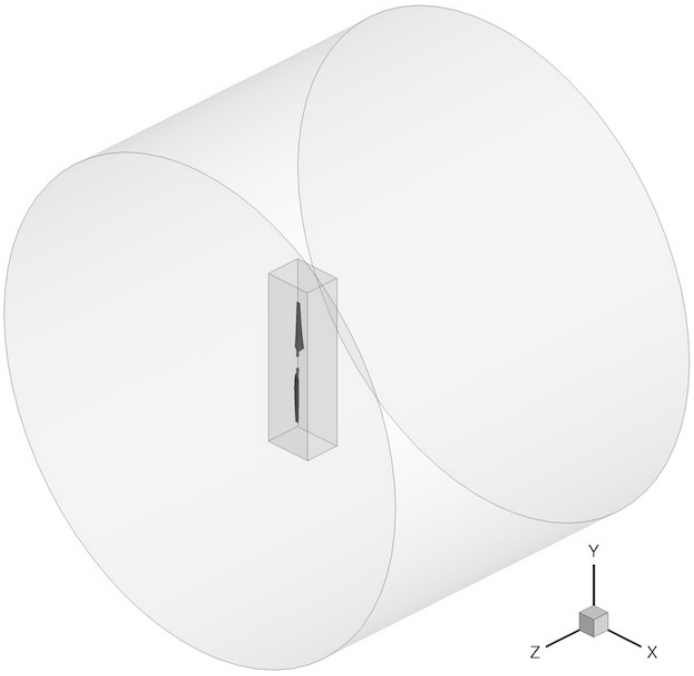
Figure 3.20 shows the computational setup and mesh resolution used in this study. The wind turbine rotor radius R is 5.029 m and the blade is assumed to be rigid.

The aerodynamic computation with prescribed wind and rotor speeds is carried out on a rotating mesh. The mesh is refined in the inner region for better resolving the flow near the rotor. At the inflow boundary the wind speed is set to 5, 7, 10, 15, 20 or 25 m/s. The traction vector at the outflow boundary and the radial component of the velocity at the radial boundary are set to zero. The air density and viscosity are 1.23 kg/m^3 and 1.78×10^{-5} kg/(m s), respectively. The mesh is comprised of 3,795,447 linear elements and 1,126,615 nodes. Figure 3.21 shows a 2D blade cross-section at 80% spanwise station to illustrate the boundary-layer mesh used in the computations. The size of the first element in the wall-normal direction is 0.002 m, and 15 layers of prismatic elements were generated with a growth ratio of 1.2. The Reynolds number based on the chord length and relative speed at this location is $O(10^6)$ and the corresponding y^+ is $O(10^2)$.

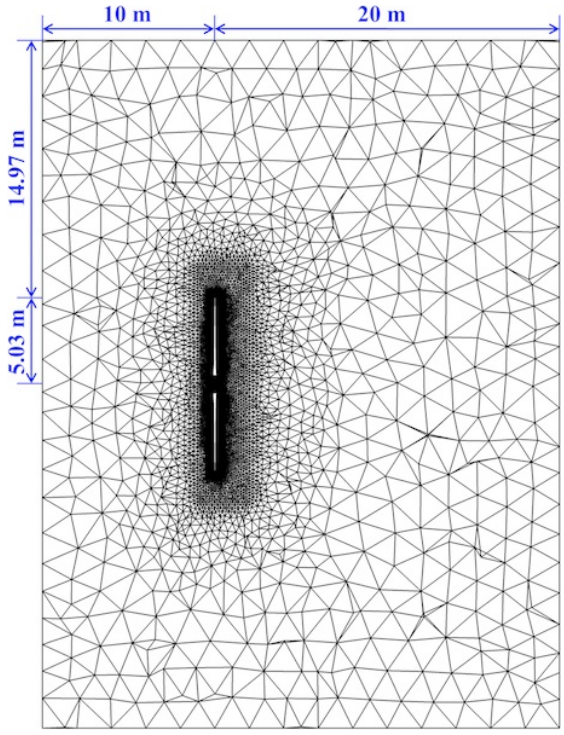
The computations were carried out in a parallel computing environment on the Lonestar Linux Cluster at the Texas Advanced Computing Center (TACC) [107]. The mesh is partitioned into subdomains using METIS [123], and each subdomain is assigned to a compute core. The ALE–VMS equations are integrated in time using the Generalized-α method [57, 110, 111]. The linear system is solved using a GMRES method [102, 124] with diagonal preconditioning. The time-step size is 1.0×10^{-5} s for 5 to 10 m/s cases, and 5.0×10^{-6} s for the rest of the cases. We typically use three Newton iterations per time step, with 50 GMRES iterations for the first and second, and 50 to 80 GMRES iterations for the third Newton iteration.

Air speed contours and velocity streamlines at 80% spanwise station are shown in Figure 3.22 for all cases. The flow is fully attached for the 5 to 10 m/s cases, and is separated for the 15 to 25 m/s cases. The 20 and 25 m/s cases are under heavily stalled conditions and the flow separates at the leading edge. The flow separation occurs due to the increased angle of attack created by the increased wind speed relative to the constant rotational speed. The pressure contours at 80% spanwise station are shown in Figure 3.23 for all cases. The differences in the pressure contours between the attached and separated flow regimes are evident in the figure.

The time-averaged low-speed shaft (aerodynamic) torque and root flap bending moment are shown in Figure 3.24 for all wind speeds. Overall, the simulation results and experimental data match remarkably well. Although, there is a slight under-prediction



(a) Computational domain



(b) Problem mesh

Figure 3.20: The computational domain and problem mesh. The mesh is refined in the inner region for better flow resolution near the rotor.

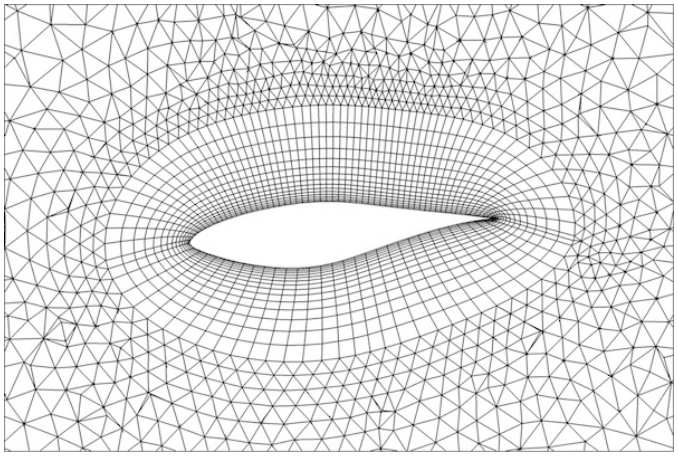


Figure 3.21: A 2D blade cross-section at 80% spanwise station to illustrate the boundary-layer mesh.

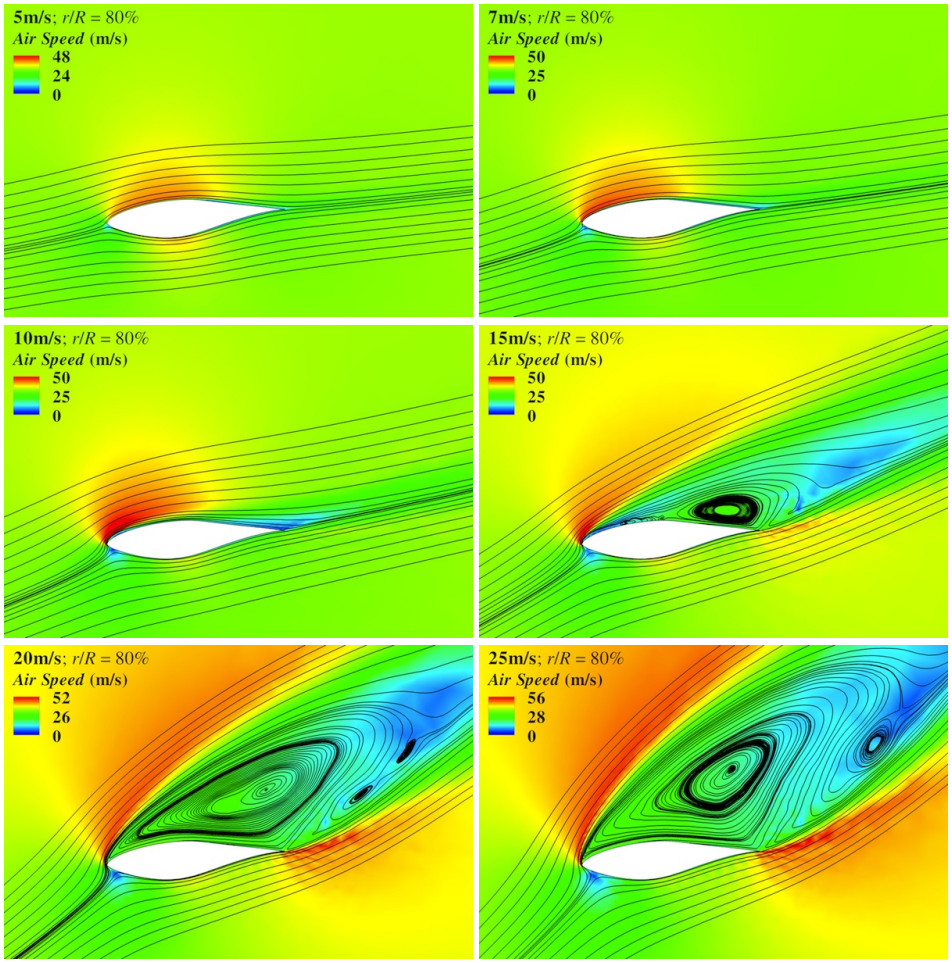


Figure 3.22: Air speed contours and velocity streamlines at 80% spanwise station for all case.

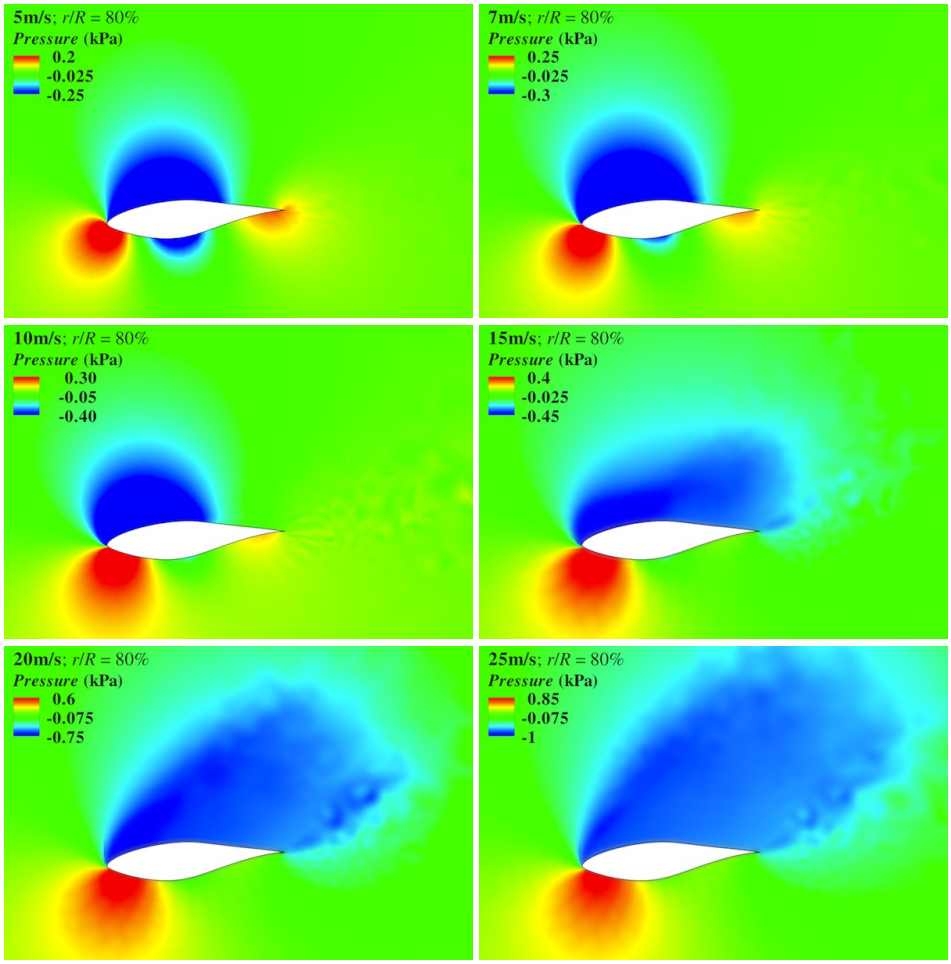
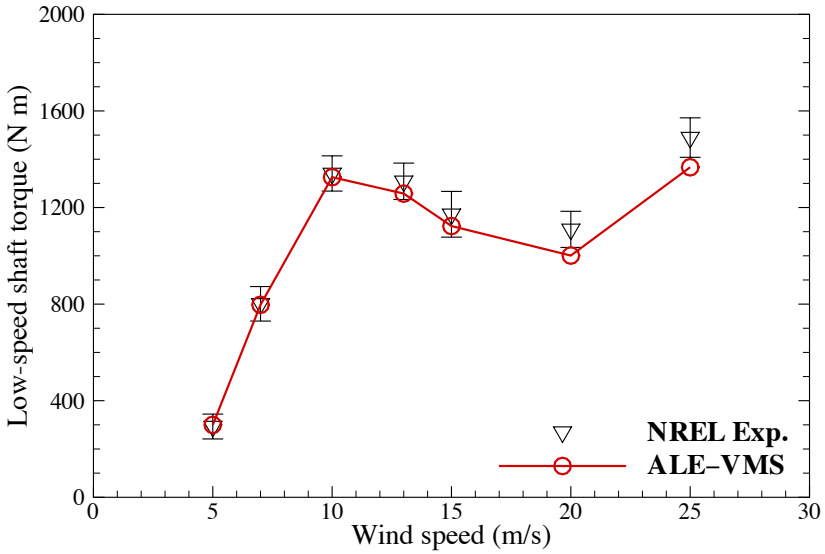
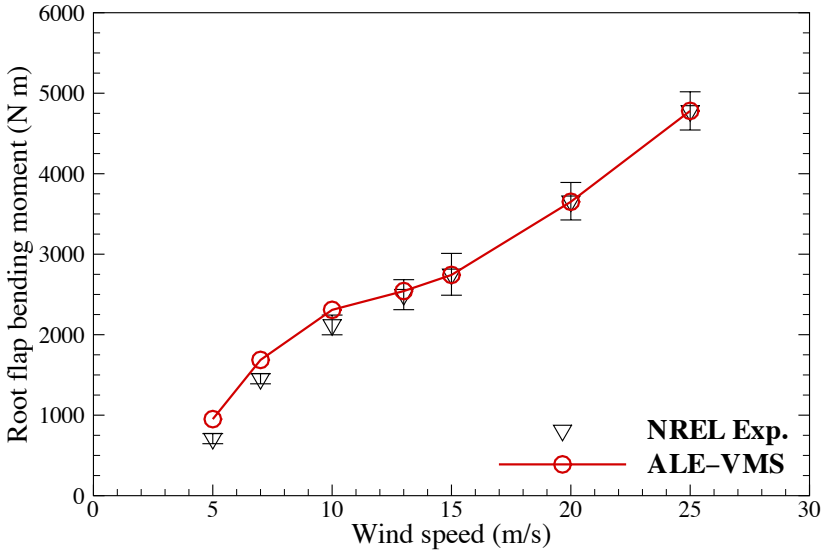


Figure 3.23: Pressure contours at 80% spanwise station for all cases.



(a)



(b)

Figure 3.24: (a) The low-speed shaft (aerodynamic) torque and (b) the root flap bending moment for all cases. The simulation results are compared with the NREL experimental data. The vertical bars represent ±1 standard deviation.

of the aerodynamic torque for the high wind speed cases, and a slight over-predictions of the root flap bending moment for the low wind speed cases.

The low-speed shaft torque and root flap bending moment represent the integrated effect of the aerodynamic loads acting on the rotor blades. It is also of interest to assess the local flow behavior by examining a distribution of the pressure coefficient over the blade surface. Figures 3.25 and 3.26 show the pressure coefficient at 30%, 46.6%, 63.3%, 80% and 95% spanwise stations of the blade for all wind speeds. The predicted values are plotted together with the experimental data. While most of the simulation results compared very well against experimental data at different radial locations, noticeable differences are found for 10 m/s wind at location $r/R = 46.6\%$, and 15 m/s wind at location $r/R = 30\%$. The same discrepancies, however, were found in the computational results of several other researchers (see, e.g., Sørensen *et al.* [5], Duque *et al.* [6], and Li *et al.* [15]). As a result, we conducted a mesh refinement study for the 10 m/s wind speed case to see if the discrepancies are due to the lack of mesh resolution. The study is presented in the next section.

Remark 3.8. Note that the same finite element ALE–VMS formulation given by Eq. (3.4) is employed to compute the entire range of wind speeds. The numerical results show that the formulation automatically adapts to, and accurately captures a full range of flow regimes present in this application, from the laminar and attached flow, to the turbulent and separated flow. There is no need to select an "appropriate turbulence model" for a given flow regime to better match the experimental data. This feature underscores the predictive capability of the finite element ALE–VMS method for this class of problems, as well as other applications.

3.4.2 Mesh refinement study

To show the convergence of the ALE–VMS method, a refinement study based on three different meshes was carried out for the 10 m/s wind case. The mesh size on the surface of the rotor and the bounding box that enclosed the rotor was reduced by a factor of $\sqrt{2}$ with each refinement. The mesh statistics are summarized in Table 3.2. We note that Mesh 2 was used for all computations in the previous section. Figures 3.27 and 3.28 show the results for the aerodynamic torque and pressure coefficient,

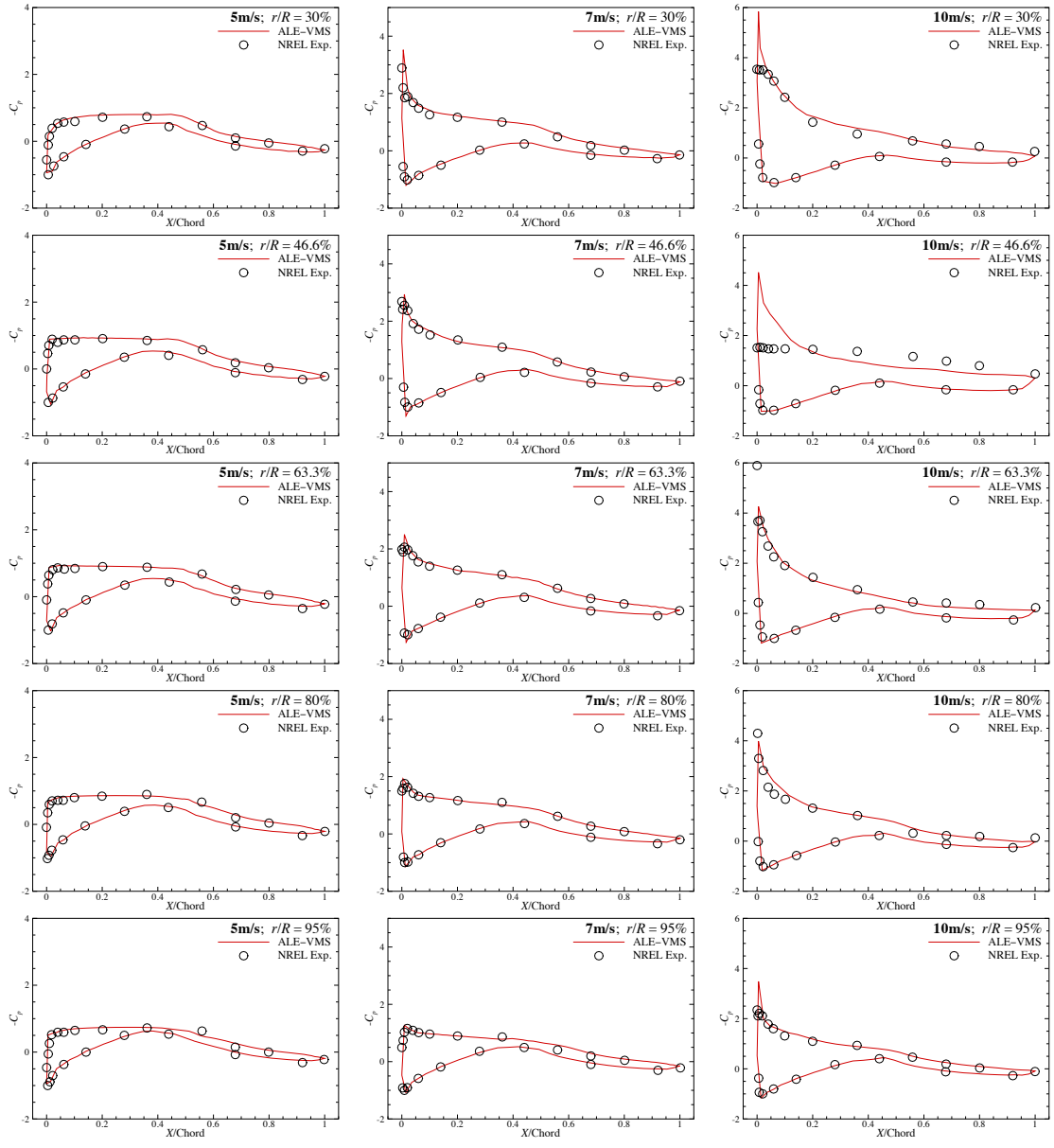


Figure 3.25: Pressure coefficient at 30%, 46.6%, 63.3%, 80% and 95% spanwise stations for 5, 7, and 10 m/s cases. The predicted values are plotted with the experimental data.

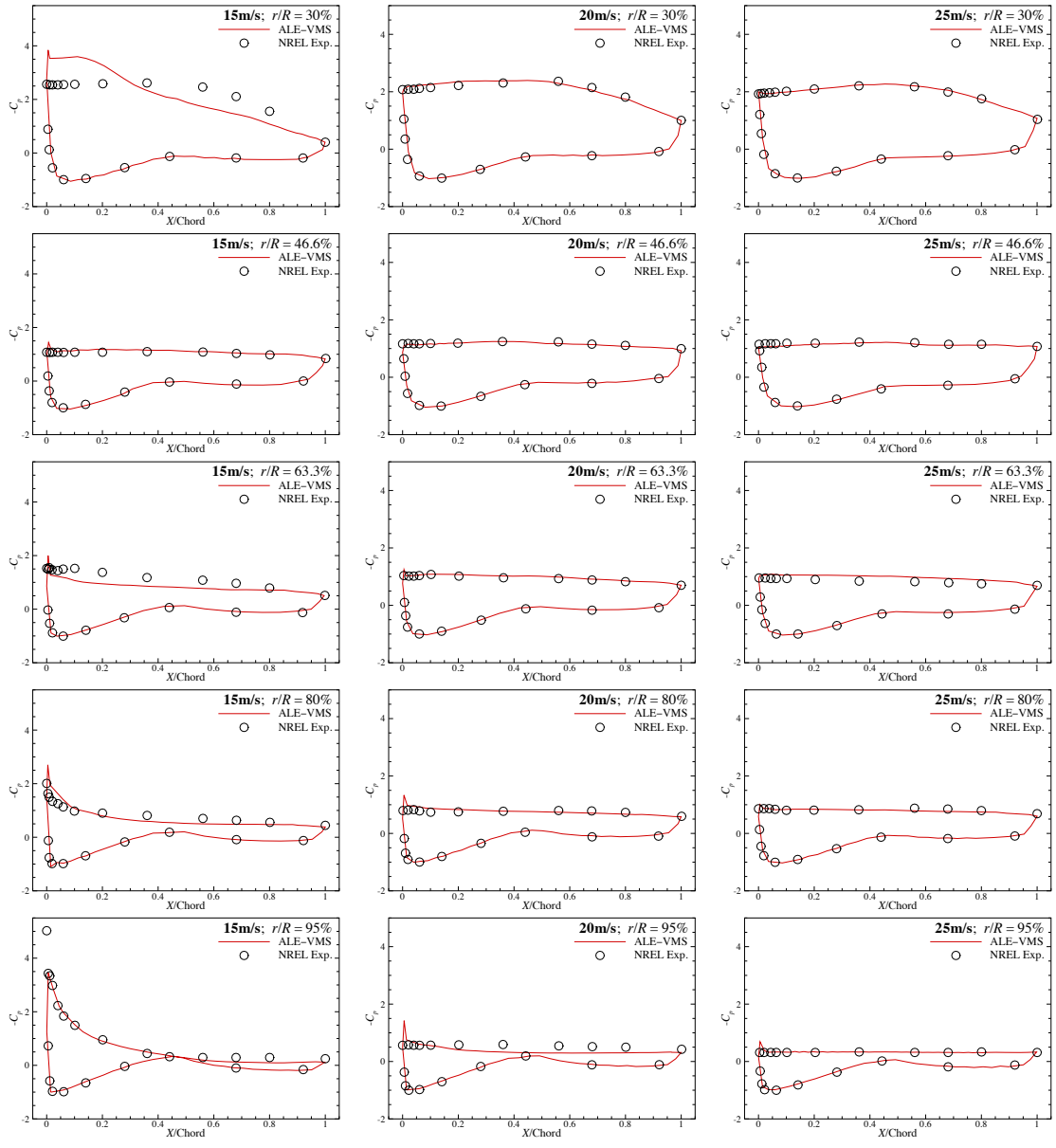


Figure 3.26: Pressure coefficient at 30%, 46.6%, 63.3%, 80% and 95% spanwise stations for 15, 20, and 25 m/s cases. The predicted values are plotted with the experimental data.

respectively. Convergence under mesh refinement is achieved for both quantities. Note that the discrepancy between the computational results and experimental data at location $r/R = 46.6\%$ is still present.

Table 3.2: Mesh statistics for the refinement study.

	Number of Nodes	Number of Elements
Mesh 1	583,627	1,991,891
Mesh 2	1,126,615	3,795,447
Mesh 3	2,178,147	7,328,867

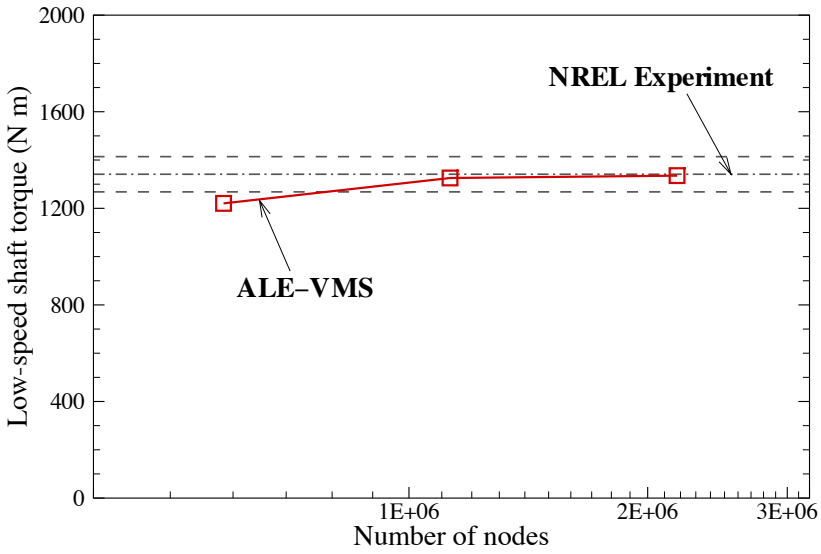


Figure 3.27: The aerodynamic torque for the 10 m/s case. The results show convergence with mesh refinement.

3.5 Simulation of the Full Wind Turbine

The results presented so far were rotor-only computations, where the wind turbine rotor is housed in a cylindrical domain and the rotation is applied to the entire computational domain. However, to simulate the full wind turbine configuration and investigate the blade–tower interaction this approach is no longer suitable. Instead, we consider an approach that makes use of a rotating subdomain, which encloses the entire wind turbine rotor, and a stationary subdomain that contains the rest of the wind turbine

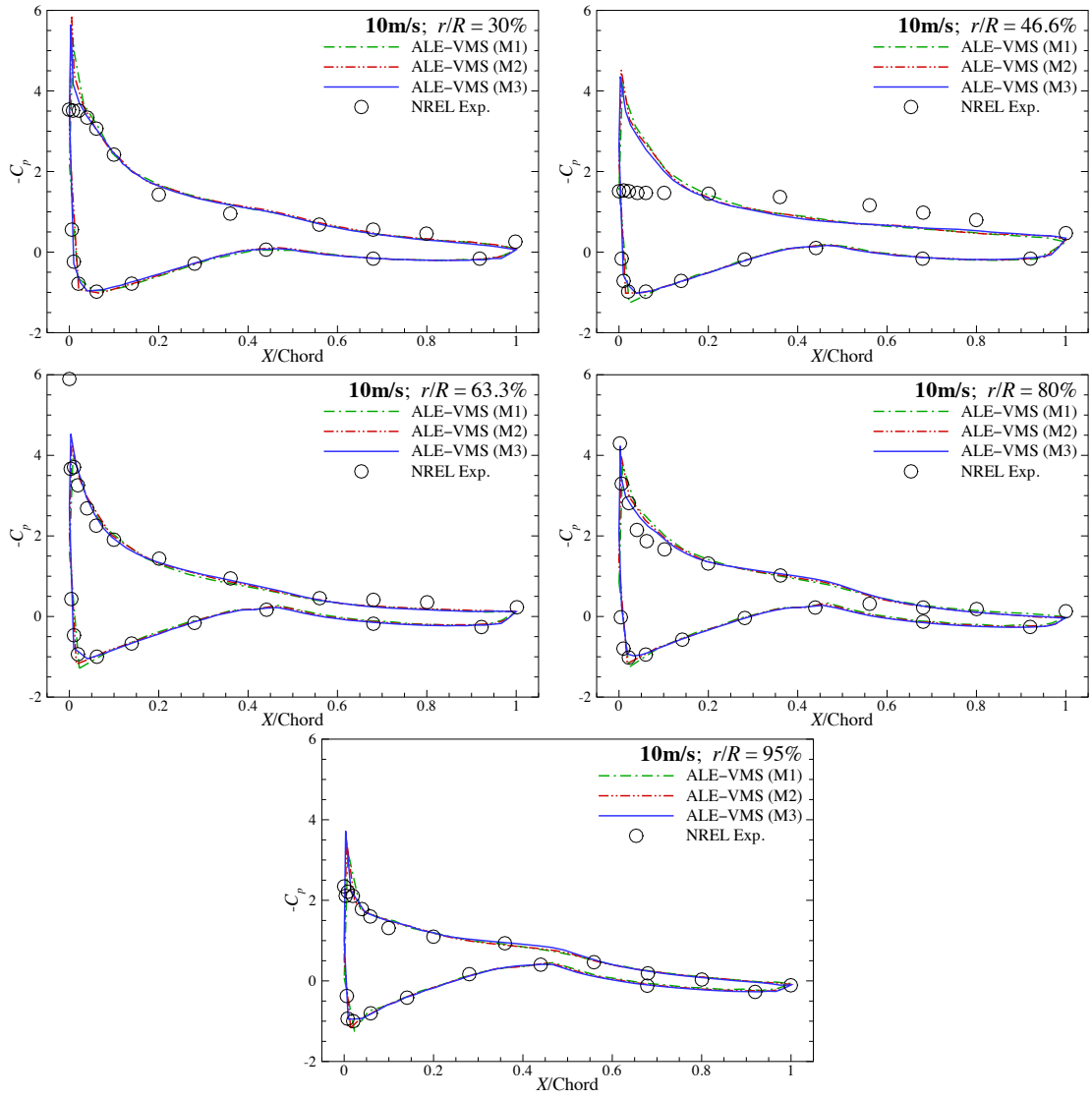


Figure 3.28: Pressure coefficient at different spanwise stations for the 10 m/s case. The results show convergence with mesh refinement.

(see Figure 3.29). The two domains are in relative motion and share a sliding cylindrical interface. Because of the relative motion, the meshes on each side of the interface are non-matching (see Figure 3.30). As a result, a numerical procedure is needed that imposes the continuity of the kinematic and traction fields despite the fact that the interface discretizations of the stationary and rotating subdomains are incompatible. Such a procedure is presented in what follows.

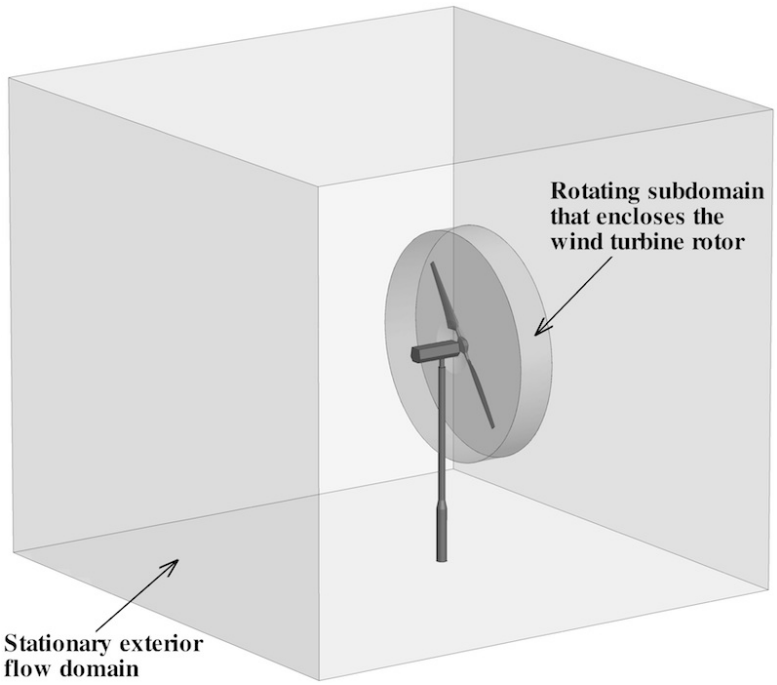


Figure 3.29: Setup for a full-wind-turbine simulation. An interior rotating subdomain, which encloses the wind-turbine rotor, and an exterior stationary subdomain, which houses the nacelle and tower, are employed.

3.5.1 The Sliding Interface Formulation

The sliding interface formulation presented here was first proposed by Bazilevs and Hughes [61] in the context of Isogeometric Analysis (IGA) [28] of flows about rotating components. The advantage of IGA for rotating-component flows is that the cylindrical sliding interfaces are represented exactly, and no geometry errors are incurred. In the case of standard FEM, the geometric compatibility is only approximate.

Let subscripts 1 and 2 denote the quantities pertaining to the fluid mechanics

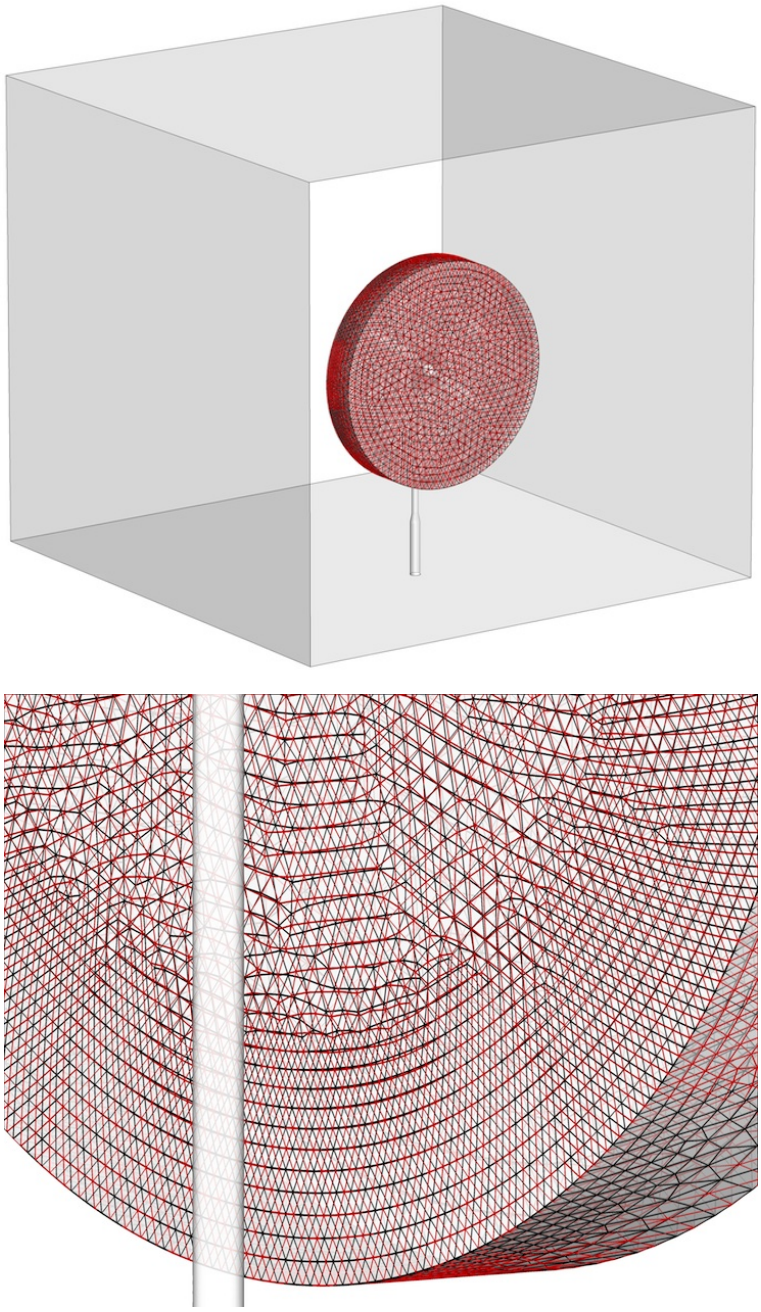


Figure 3.30: Non-matching meshes at the sliding interface between the stationary and rotating subdomains.

problem on the stationary and rotating subdomains, respectively. The semi-discrete formulation that weakly imposes the continuity of the fluid velocity and traction vectors at the sliding interface is stated as: find $\{\mathbf{u}_1^h, p_1^h\}$ and $\{\mathbf{u}_2^h, p_2^h\}$, such that $\forall\{\mathbf{w}_1^h, q_1^h\}$ and $\forall\{\mathbf{w}_2^h, q_2^h\}$,

$$
\begin{aligned}
&B_1^{\mathrm{VW}}(\{\mathbf{w}_1^h, q_1^h\}, \{\mathbf{u}_1^h, p_1^h\}; \hat{\mathbf{u}}_1^h) - F_1(\{\mathbf{w}_1^h, q_1^h\}) + B_2^{\mathrm{VW}}(\{\mathbf{w}_2^h, q_2^h\}, \{\mathbf{u}_2^h, p_2^h\}; \hat{\mathbf{u}}_2^h) - F_2(\{\mathbf{w}_2^h, q_2^h\}) \\
&- \sum_{b=1}^{N_{\mathrm{eb}}} \int_{\Gamma_t^b \cap (\Gamma_t)_{\mathrm{I}}} \left(\mathbf{w}_1^h - \mathbf{w}_2^h\right) \cdot \frac{1}{2} \left(\boldsymbol{\sigma}_1\left(\mathbf{u}_1^h, p_1^h\right)\mathbf{n}_1 - \boldsymbol{\sigma}_2\left(\mathbf{u}_2^h, p_2^h\right)\mathbf{n}_2\right) \, \mathrm{d}\Gamma \\
&- \sum_{b=1}^{N_{\mathrm{eb}}} \int_{\Gamma_t^b \cap (\Gamma_t)_{\mathrm{I}}} \frac{1}{2} \left(\delta\boldsymbol{\sigma}_1\left(\mathbf{w}_1^h, q_1^h\right)\mathbf{n}_1 - \delta\boldsymbol{\sigma}_2\left(\mathbf{w}_2^h, q_2^h\right)\mathbf{n}_2\right) \cdot \left(\mathbf{u}_1^h - \mathbf{u}_2^h\right) \, \mathrm{d}\Gamma \\
&- \sum_{b=1}^{N_{\mathrm{eb}}} \int_{\Gamma_t^b \cap (\Gamma_t)_{\mathrm{I}}} \mathbf{w}_1^h \cdot \rho \left\{\left(\mathbf{u}_1^h - \hat{\mathbf{u}}_1^h\right) \cdot \mathbf{n}_1\right\}_- \left(\mathbf{u}_1^h - \mathbf{u}_2^h\right) \, \mathrm{d}\Gamma \\
&- \sum_{b=1}^{N_{\mathrm{eb}}} \int_{\Gamma_t^b \cap (\Gamma_t)_{\mathrm{I}}} \mathbf{w}_2^h \cdot \rho \left\{\left(\mathbf{u}_2^h - \hat{\mathbf{u}}_2^h\right) \cdot \mathbf{n}_2\right\}_- \left(\mathbf{u}_2^h - \mathbf{u}_1^h\right) \, \mathrm{d}\Gamma \\
&+ \sum_{b=1}^{N_{\mathrm{eb}}} \int_{\Gamma_t^b \cap (\Gamma_t)_{\mathrm{I}}} \frac{C_I^B \mu}{h_n} \left(\mathbf{w}_1^h - \mathbf{w}_2^h\right) \cdot \left(\mathbf{u}_2^h - \mathbf{u}_1^h\right) \, \mathrm{d}\Gamma = 0,
\end{aligned}
\tag{3.31}
$$

where

$$
\begin{aligned}
B^{\mathrm{VW}}(\{\mathbf{w}^h, q^h\}, \{\mathbf{u}^h, p^h\}; \hat{\mathbf{u}}^h) = {} & B^{\mathrm{VMS}}(\{\mathbf{w}^h, q^h\}, \{\mathbf{u}^h, p^h\}; \hat{\mathbf{u}}^h) \\
& + B^{\mathrm{WBC}}(\{\mathbf{w}^h, q^h\}, \{\mathbf{u}^h, p^h\}; \hat{\mathbf{u}}^h)
\end{aligned}
\tag{3.32}
$$

and

$$
\delta\boldsymbol{\sigma}\left(\mathbf{w}^h, q^h\right) = q^h \mathbf{I} + 2\mu\boldsymbol{\varepsilon}\left(\mathbf{w}^h\right). \tag{3.33}
$$

The terms on the first line of Eq. (3.31) correspond to the ALE–VMS formulation with weak boundary conditions in the stationary and rotating subdomains. The rest of the terms in Eq. (3.31) are associated with the weak enforcement of the kinematic and traction compatibility at the sliding interface. Here, $(\Gamma_t)_{\mathrm{I}}$ denotes the interface between the stationary and rotating subdomains, and $\{\mathcal{A}\}_-$ denotes the negative part of $\mathcal{A}$, that is, $\{\mathcal{A}\}_- = \mathcal{A}$ if $\mathcal{A} < 0$ and $\{\mathcal{A}\}_- = 0$ if $\mathcal{A} \geq 0$. The formulation may be see as a Discontinuous Galerkin method (see, e.g., Arnold *et al.* [68]), where the continuity of the basis function is enforced everywhere in the interior of the two subdomains, but not at

the sliding interface between them. Note that, in this case, $\hat{\mathbf{u}}_1^h = \mathbf{0}$, because domain 1 is stationary. However, the formulation is able to handle situations where both subdomains 1 and 2 are in motion.

The structure of the terms on the sliding interface is similar to that of the weak enforcement of essential boundary conditions (see Eq. (3.14)). The significance of each term is as follows: The term on the second line of Eq. (3.31) is the consistency term. It is necessary to ensure that the discrete formulation is identically satisfied by the exact solution of the Navier–Stokes equations, which implies the accuracy of the discrete formulation. Also note that when deriving the Euler–Lagrange equations corresponding to Eq. (3.31), this term cancels with the terms that come from the integration-by-parts of the stress terms, thus correctly removes traction boundary conditions from the sliding interface. The term on the third line is the so-called adjoint consistency term. It ensures that the analytical solution of the adjoint equations, when introduced in place of the linear momentum and continuity equation test functions, also satisfies the discrete formulation. Adjoint consistency is linked to optimal convergence of the discrete solution in lower-order norms (see, e.g., Arnold *et al.* [68]). The next two terms are the upwinding, which lead to better satisfaction of inflow conditions. The last term is penalty-like, in that it penalizes the discrete version of the kinematic compatibility at the sliding interface. It is necessary to ensure the stability of the discrete formulation, which may be lost due to the introduction of consistency and adjoint consistency terms.

3.5.2 Simulation Results

The sliding interface formulation is applied to the simulation of the full NREL Phase VI wind turbine configuration, including the rotor (blades and hub), nacelle and tower. The tower is composed of two cylinders with diameters of 0.6096 m and 0.4064 m, that are connected with a short conical section. The tower height is 11.144 m above the wind tunnel floor. The detailed geometry and configuration of the tower and nacelle can be found in Hand *et al.* [113]. In this study, we considered the 7 and 10 m/s wind cases from the same experimental sequence S mentioned in Section 3.4.1. Figures 3.31 and 3.32 show the flow visualization of the full-wind-turbine simulations. The flow structures are different between the two cases. The tip vortex for the 7 m/s case decays

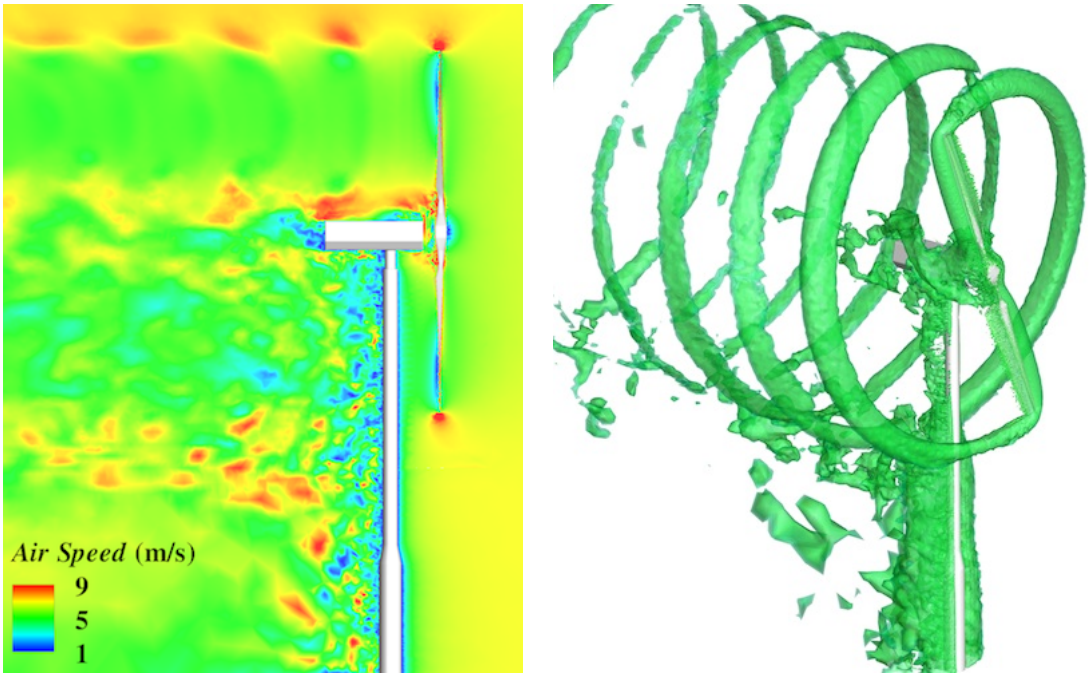


Figure 3.31: Air speed contours at a planar cut (left), and isosurfaces of air speed (right) at an instant for the 7 m/s case.

very slowly as it is convected downstream, while the tip vortex breaks down quickly for the 10 m/s case. Figure 3.33 shows the flow field when the blade passes in front of the tower for the 7 m/s case. Note that no visible discontinuities are seen in the flow field at the sliding interface, which indicates that the method correctly handles the kinematic compatibility in this location.

To see the influence of the tower, the single-blade aerodynamic torque over a full revolution is plotted in Figure 3.34 for both 7 and 10 m/s cases. The results of the full-wind-turbine computations are compared with the experimental data, as well as with the results of the rotor-only computations.

Remark 3.9. The experimental data plotted in Figure 3.34 corresponds to the "estimated aerodynamic torque" data adjusted to match the mean of the "corrected low-speed shaft torque" data. The former is obtained by integrating the experimental pressure measurements along the span of the blade, while the latter is obtained from strain-gauge data corrected to remove the gravity effect. Because only five spanwise cross-sections were employed in the calculation of the "estimated aerodynamic torque", we feel that the

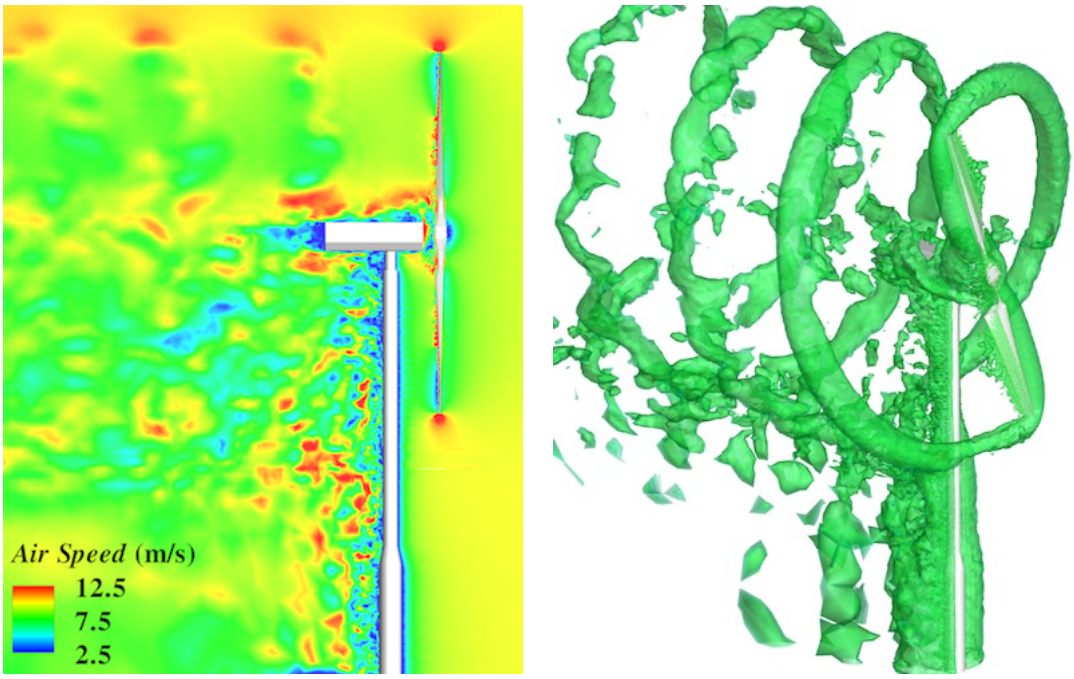


Figure 3.32: Air speed contours at a planar cut (left), and isosurfaces of air speed (right) at an instant for the 10 m/s case.

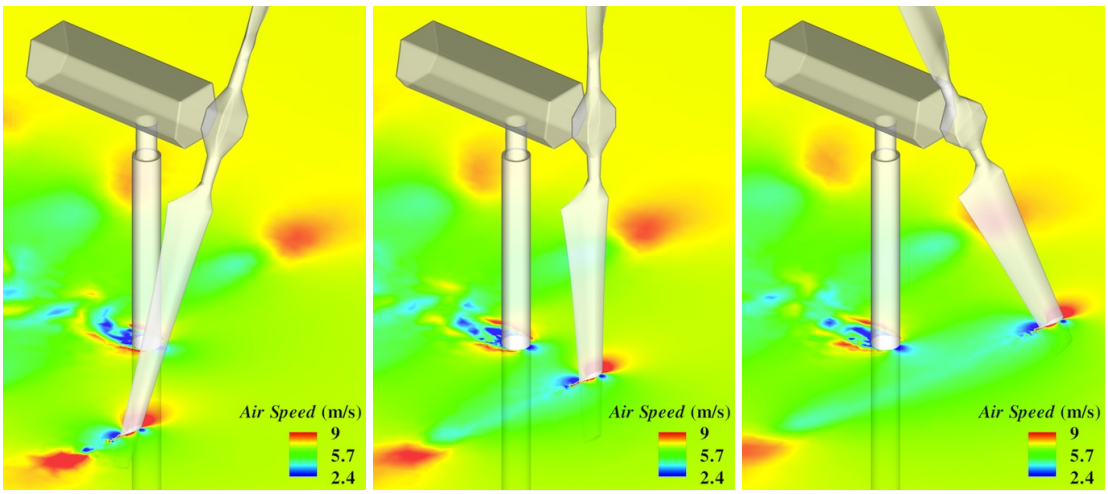
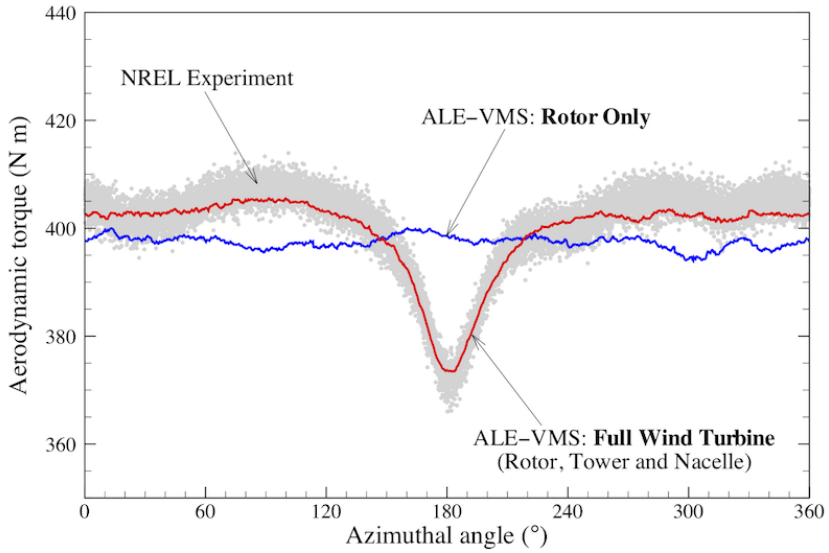
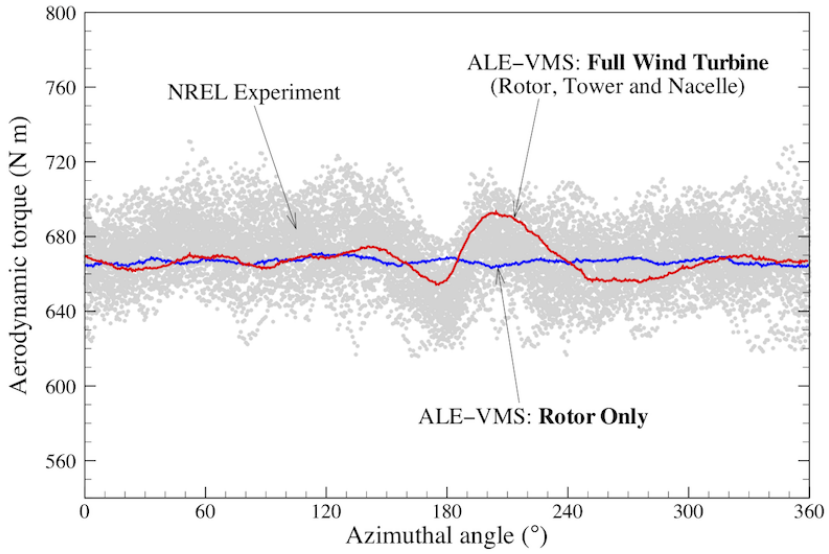


Figure 3.33: Air speed contours at 80% spanwise station for the 7 m/s case.



(a) 7 m/s case



(b) 10 m/s case

Figure 3.34: The single-blade aerodynamic torque over a full revolution for 7 and 10 m/s cases. The 180° azimuthal angle corresponds to the instant when the blade passes in front the tower. The tower effect is clearly pronounced in the 7 m/s case. It is also present in the 10 m/s case, but is not as significant. The results in both cases are in very good agreement with the experimental data.

mean value produced by the strain-gauge measurement is more reliable. On the other hand, because the gravity effect is removed in an approximate manner, we feel that the fluctuating component of the data is more accurately captured using spanwise pressure integration. For more discussion on this see Hand *et al.* [113].

For the full-wind-turbine simulation of the 7 m/s case, Figure 3.34(a) clearly shows the drop in the aerodynamic torque at an instant when the blade passes in front of the tower, which corresponds to the azimuthal angle of 180°. The drop in the torque is about 8% relative to its value when the blade is away from the tower. These results are in good agreement with the experimental data. The rotor-only computation is obviously unable to predict this feature, which may be important for the transient structural response of the blades (not considered in this work). It should be noted, however, that the cycle-averaged aerodynamic torque is nearly identical for the full-wind-turbine and the rotor-only simulations. The picture is completely different for the 10 m/s case, where the influence of the tower, although clearly present, is a lot less pronounced.

3.6 The Role of Weakly Enforced Essential Boundary Conditions

To assess the role of weakly enforced essential boundary conditions in predicting wind turbine aerodynamics using the proposed ALE–VMS method, in this section we perform the simulations for both NREL Phase VI and NREL 5 MW offshore baseline wind turbine rotors on relatively coarse boundary layer meshes. Both weakly and strongly enforced no-slip and no-penetration boundary conditions are employed in these simulations.

3.6.1 NREL Phase VI Wind Turbins Rotor

Two cases from the sequence S of NREL Phase VI experiment [113] were selected for this study. The first case has the wind speed of 5 m/s and the second case has the wind speed of 25 m/s. The two cases we considered here present very different flow conditions. For the 5 m/s case the flow is fully attached for the entire blade. On

the contrary, stall occurred for most of the blade for the case of 25 m/s wind, and it is considered to be more challenging for simulation (see, e.g., Li *et al.* [15]).

The mesh resolution and computational domain are shown in Figure 3.35. The ratio of the rotor radius to the radius and the axial length of the computational domain are approximately 1/4 and 1/6, respectively. The mesh is refined in the inner region for better flow resolution near the rotor, and is comprised of 8,494,182 linear tetrahedral elements and 1,508,983 nodes. Figure 3.36 shows a 2D blade cross-section at 80% spanwise station to illustrate the type of mesh near the boundary that we used in our computation. Near the blade surface the size of the first element in the wall-normal direction is about 0.008 m, the Reynolds number based on the chord length and relative speed at this location is $O(10^6)$, and the corresponding y^+ is $O(10^3)$. No special boundary layer meshing was used in this study, in part to test the ability of the ALE–VMS method to deal with coarse boundary layer meshes.

The computations were carried out on 128 compute cores on a Dell Cluster [125] at TACC. The system consists of 256 dual-socket nodes, each with two Intel Nehalem 2.53 GHz quad-core processors. The time-step size is 0.0001 s. The number of Newton's iterations per time step is three with 50 GMRES iterations for the first and second nonlinear iterations, and 50 to 80 GMRES iterations for the third nonlinear iteration. In general, it takes about 20 to 24 hours to compute 1 s of real time, which is generally sufficient to predict the aerodynamic torque.

The time history of aerodynamic torque is shown in Figure 3.37. Good agreement of the aerodynamic torque is found between the weak boundary condition simulations and experimental data for both flow conditions. However, the results for the strongly enforced boundary condition simulations are not at all accurate.

Pressure and air speed contours and velocity streamlines at 80% spanwise station at an instant for 5 and 25 m/s cases are shown in Figures 3.38 and 3.39, respectively. Figure 3.38(a) shows the weak boundary condition prediction of the air flow for the 5 m/s case. Here, the flow is fully attached, and the torque is correctly predicted. However, the strong boundary condition simulation predicts flow separation at the trailing edge (see Figure 3.38(b)). The blade stalls and, as a result, the torque is underpredicted by 126% (see Figure 3.37(a)). For the 25 m/s case, small differences are found in the

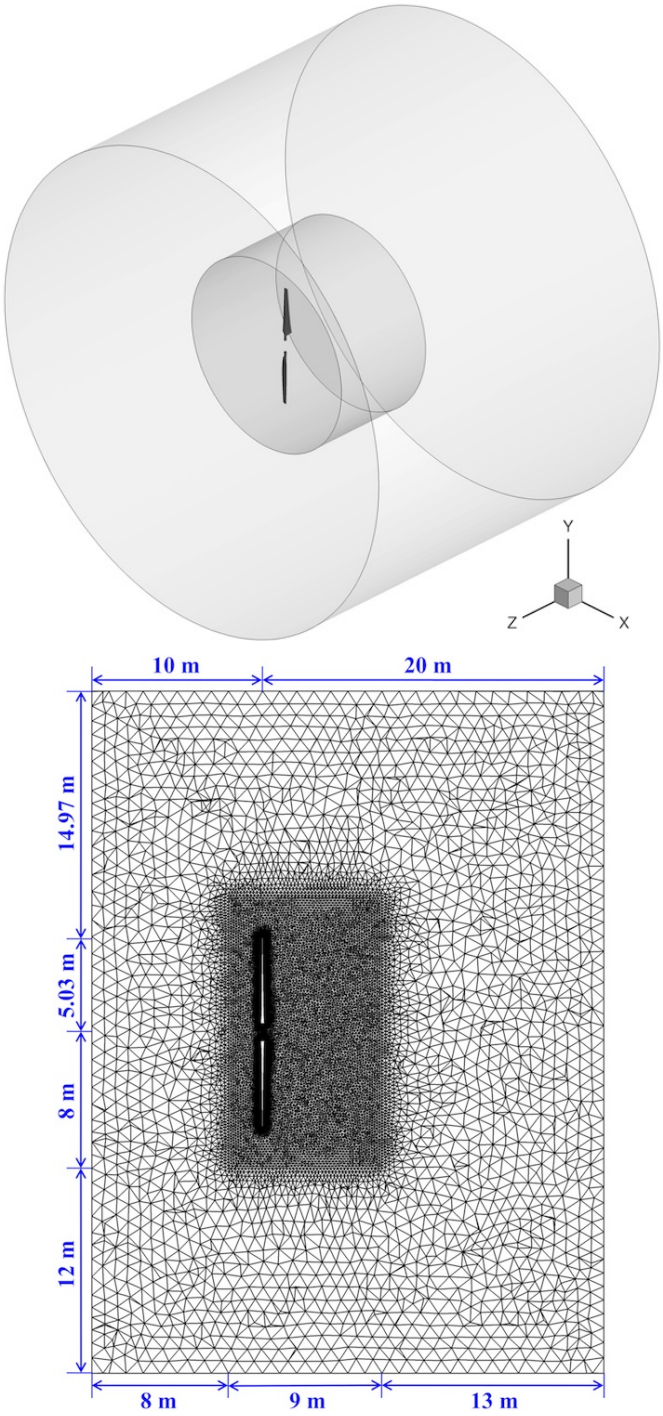


Figure 3.35: NREL Phase VI wind turbine. The problem mesh and computational domain. The ratio of the rotor radius to the radius and the axial length of the computational domain are approximately 1/4 and 1/6, respectively. The mesh is refined in the inner region for better flow resolution near the rotor.

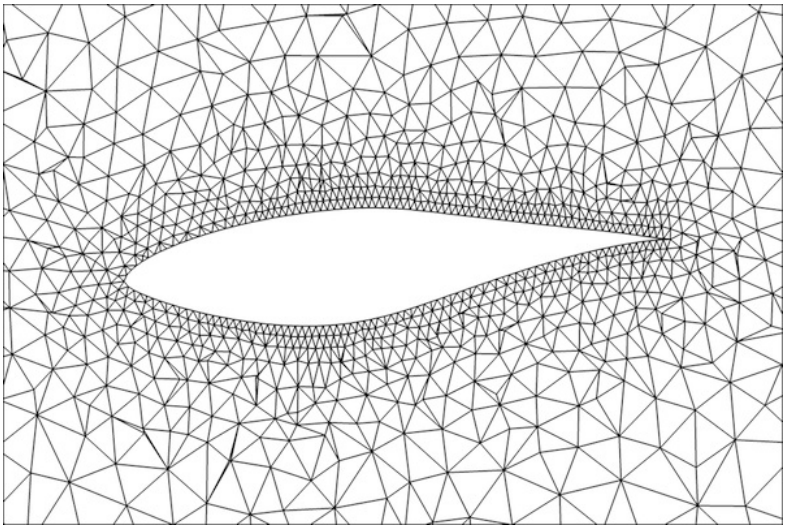
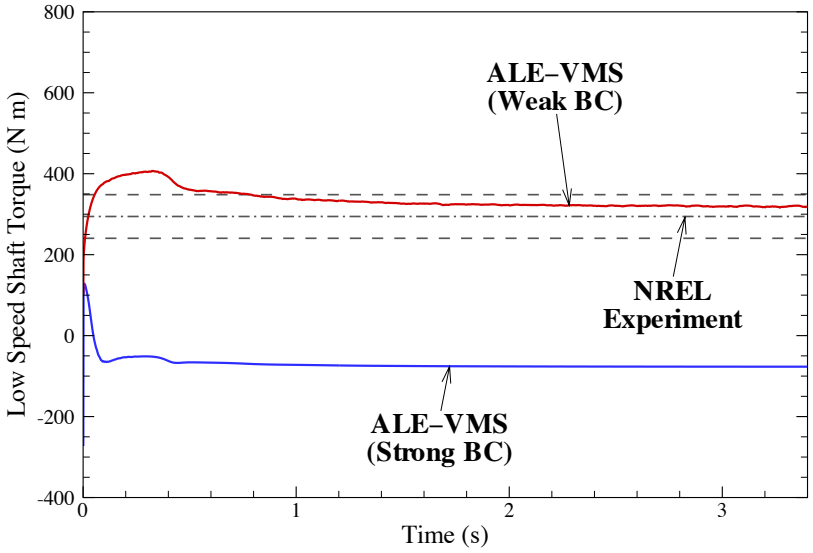


Figure 3.36: NREL Phase VI wind turbine. A 2D blade cross-section at 80% spanwise stationto illustrate the type of mesh near the boundary that we used in our computation. The size of the first element in the wall-normal direction is about 0.008 m.

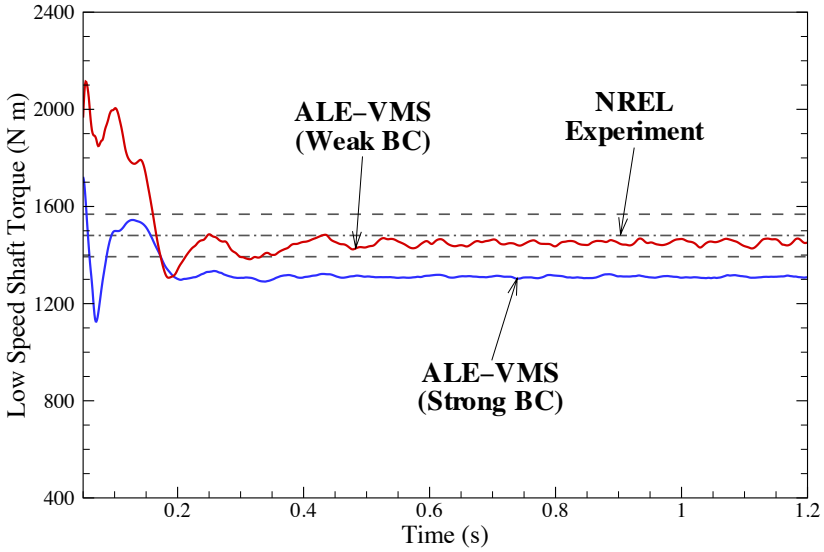
pressure contours and air flow patterns between the weak and strong boundary condition computations. This is due to the fact that the flow is already separated at the edges, the entire airfoil is stalled, and the boundary layer resolution is not so important for these type of flow conditions. In this case, the weak boundary condition again correctly predicts the torque, while the strong boundary condition underpredicts the torque, but only by 11% (see Figure 3.37(b)).

These results are not surprising. In the case of strongly enforced boundary conditions, the coarse boundary layer discretization gives rise to artificially "thick" boundary layers, which retard the flow and lead to non-physical aerodynamics, such as premature flow separation. In the case of weakly enforced boundary conditions, the flow is allowed to slip on the solid surface without forming these undesired thick boundary layers. Of course, with sufficient boundary layer mesh refinement, both approaches will capture the boundary layer, and the strongly enforced boundary condition approach will also produce the correct result (see Bazilevs *et al.* [42]).

Figure 3.40 shows the flow visualization (isosurfaces of air speed) of the 5 m/s case. The tip vortex generated by the blade is carried downstream of the rotor with little decay. Pressure contours are also plotted on the rotor surface in the figure.

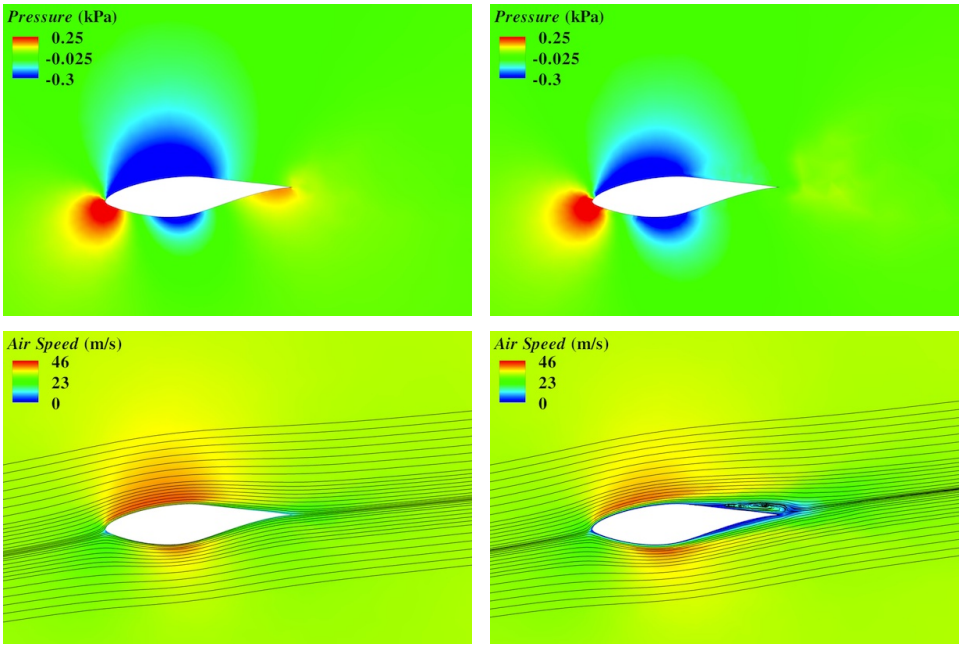


(a) 5 m/s case



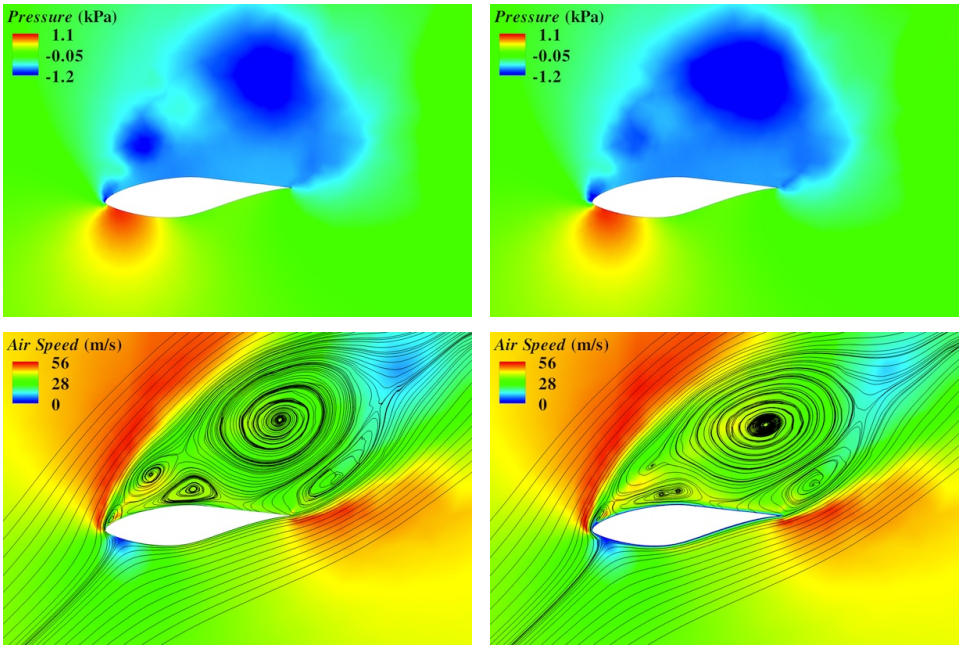
(b) 25 m/s case

Figure 3.37: NREL Phase VI wind turbine. The time history of the aerodynamic (low-speed shaft) torque for both weak and strong boundary condition simulations for (a) 5 m/s and (b) 25 m/s cases. The results are compared to the NREL experimental data reported in [5, 126]. Dashed line represents the experimental standard deviation to indicated the variation over one revolution



(a) Weakly enforced boundary condition (b) Strongly enforced boundary condition

Figure 3.38: NREL Phase VI wind turbine. Pressure and air speed contours and velocity streamlines at 80% spanwise station at an instant for 5 m/s case.



(a) Weakly enforced boundary condition (b) Strongly enforced boundary condition

Figure 3.39: NREL Phase VI wind turbine. Pressure and air speed contours and velocity streamlines at 80% spanwise station at an instant for 25 m/s case.

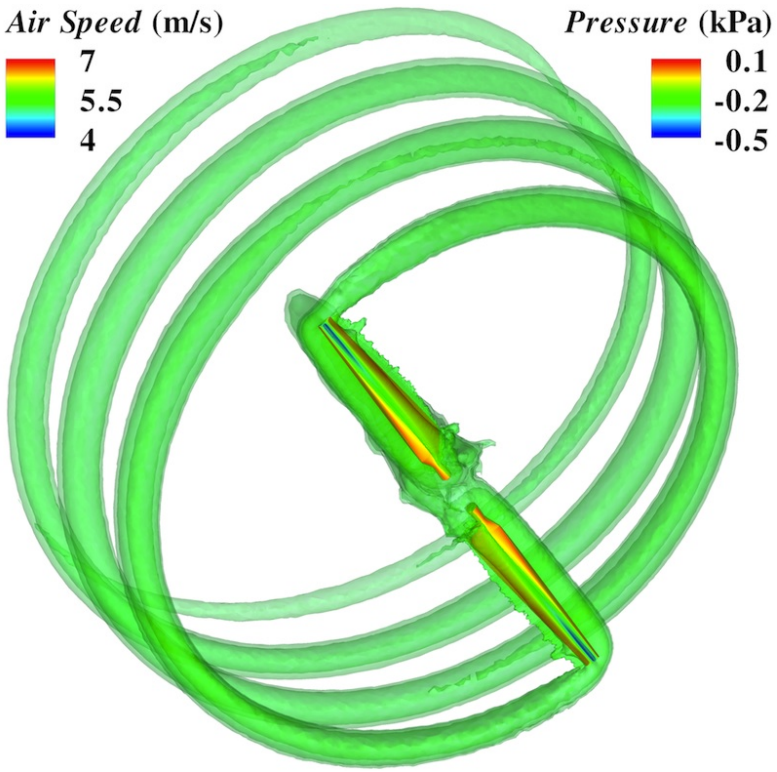


Figure 3.40: NREL Phase VI wind turbine. Isosurfaces of air speed at an instant for 5 m/s case. The tip vortex generated by the blade is carried downstream of the rotor with little decay. Pressure contours are also plotted on the rotor surface.

3.6.2 NREL 5 MW Offshore Baseline Wind Turbine

In this section, we present our computations of the NREL 5 MW offshore baseline wind turbine rotor [4] and compare the relative performance of the weak and strong boundary conditions for this significantly larger diameter wind turbine rotor design. We compute the case corresponding to wind speed of 11.4 m/s and rotor speed of 12.1 rpm.

In the previous studies of this problem (see Section 3.3.2), NURBS-based isogeometric analysis with strongly enforced boundary conditions were employed. Also, a 120° slice of the computational domain with rotationally periodic boundary conditions was used. Here, motivated by the fact that we are also moving toward simulating the full wind turbine assembly of this design, we consider a full domain of this three-bladed rotor without rotationally periodic boundary conditions.

The aerodynamics volume mesh is comprised of 6,309,349 linear tetrahedral elements and 1,193,404 nodes. Figure 3.41 shows a 2D blade cross-section at 75% spanwise station to illustrate the coarse boundary layer mesh used in our computations. Near the blade surface, the size of the first element in the wall-normal direction is about 0.075 m. The Reynolds number based on the chord length and relative speed at this location is $O(10^7)$, and the corresponding y^+ is $O(10^4)$.

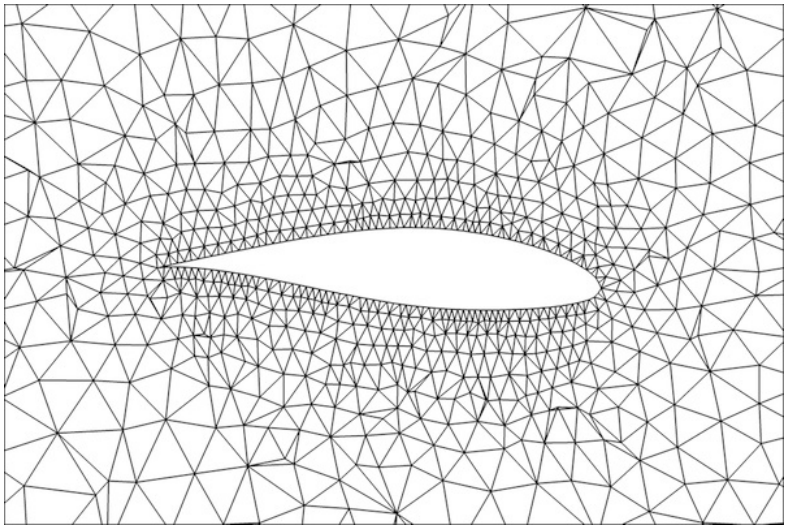


Figure 3.41: A 2D blade cross-section cut at 75% spanwise station to illustrate the coarse boundary layer mesh used in our computations. Near the blade surface, the size of the first element in the wall-normal direction is about 0.075 m.

The time history of the aerodynamic torque is plotted in Figure 3.42 for both weak and strong boundary condition simulations. The results are compared to the NURBS-based simulation from Section 3.3.2 and the NREL prediction from Jonkman *et al.* [4] obtained using FAST [3]. The weak boundary condition result is nearly identical to the NURBS-based simulation except for low-amplitude, high-frequency fluctuations present in the NURBS-based simulation, which better captures the fine-scale turbulence due to higher-order functions and better boundary-layer resolution employed. The strong boundary condition result in this study is, however, completely inaccurate. Nevertheless, as evident in Section 3.3.2, with sufficient boundary layer mesh refinement, strongly enforced boundary condition approach will capture the boundary layer and will also produce the correct result.

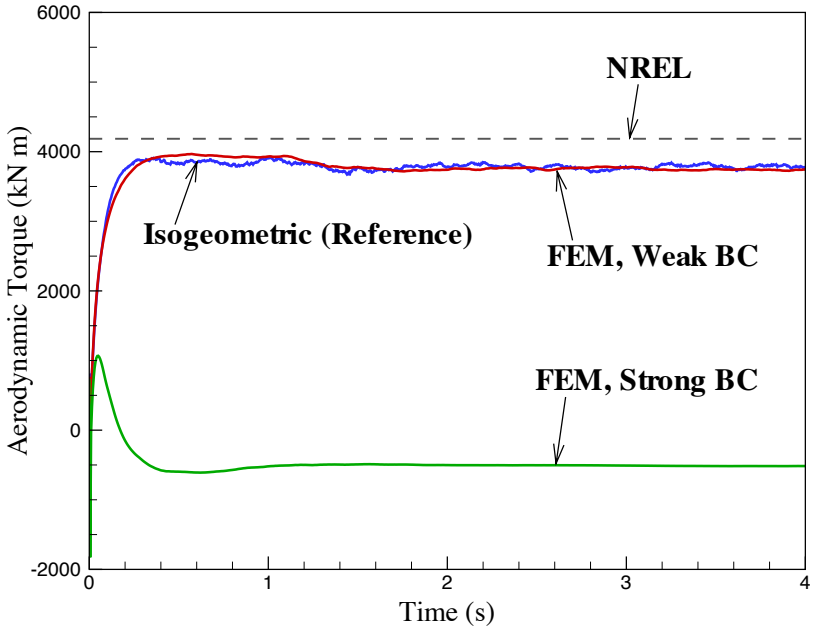
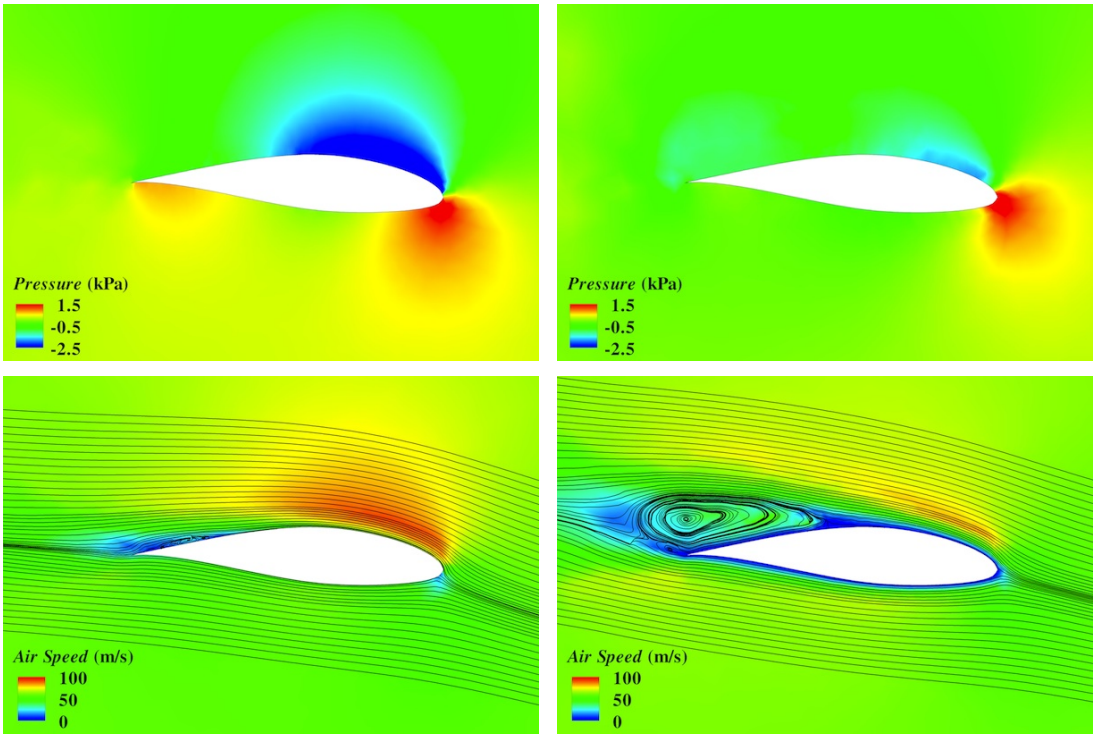


Figure 3.42: The time history of the aerodynamic torque for both weak and strong boundary condition simulations. The results are compared to the NURBS-based simulation from Section 3.3.2 and the NREL prediction [4] obtained using FAST [3].

The comparison of the air speed and pressure contours at the blade cross-section corresponding to 75% spanwise station for weakly and strongly enforced boundary condition simulations is shown in Figure 3.43. As in the case of the Phase VI rotor, the strong boundary condition simulation produces a thick boundary layer due to the lack



(a) Weakly enforced boundary condition (b) Strongly enforced boundary condition

Figure 3.43: Pressure and air speed contours and velocity streamlines at 75% spanwise station for weakly and strongly enforced boundary condition simulations.

of mesh refinement, which leads to unphysical flow separation, incorrect pressure distribution around the airfoil, and, as a result, inaccurate torque prediction.

Figure 3.44 shows the air speed contours (using the weak boundary condition formulation) in the rotor plane. As expected, the large scales of the air flow are very similar between the three blades. However, due to the fine-scale turbulence, the small-scale features show some differences and the flow is not fully periodic. Of course, the flow rotational periodicity will be further reduced in the presence of the tower.

3.7 Acknowledgements

Chapter 3, in part, is a reprint of the material as it appears in: "Wind turbine aerodynamics using ALE–VMS: Validation and the role of weakly enforced boundary conditions," (with I. Akkerman and Y. Bazilevs), *Computational Mechanics*, 2012; "ALE–

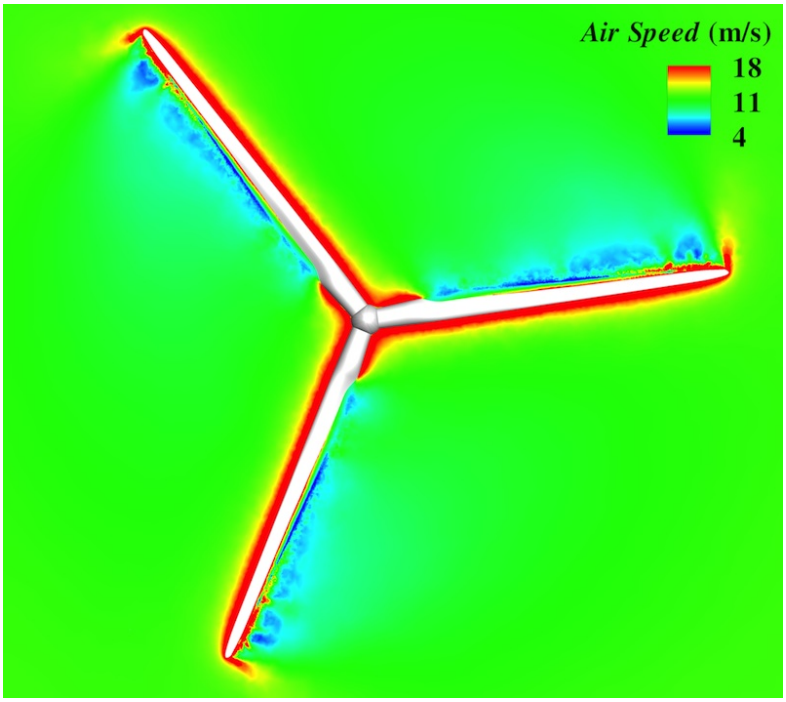


Figure 3.44: NREL 5 MW offshore baseline wind turbine. Air speed contours (using the weak boundary condition formulation) in the rotor plane. The large scales of the air flow are very similar between the three blades. However, the small-scale features show some differences and the flow is not fully periodic.

VMS and ST–VMS methods for computer modeling of wind-turbine rotor aerodynamics and fluid–structure interaction," (with Y. Bazilevs, K. Takizawa and T.E. Tezduyar), *Mathematical Models and Methods in Applied Sciences*, 2012; "High-performance computing of wind turbine aerodynamics using isogeometric analysis," (with I. Akkerman and Y. Bazilevs), *Computers & Fluids*, 2011. This chapter, in part, has also been submitted for publication of the material as it may appear in: "Finite element simulation of wind turbine aerodynamics: Validation study using NREL Phase VI experiment," (with I. Akkerman and Y. Bazilevs), *Wind Energy*, 2012. The dissertation author was the primary investigator and author of these papers.

Chapter 4

Structural Modeling of Wind Turbine Blades

4.1 Isogeometric Kirchhoff–Love Composite Shell and the Bending Strip Method

4.1.1 Kirchhoff–Love Shells

In this section we present the governing equations of the Kirchhoff–Love shell theory. The theory is appropriate for thin shell structures and requires no rotational degrees of freedom. The variational formulation of a Kirchhoff–Love shell is based on the principle of virtual work expressed as

$$\delta W = \delta W_{\text{int}} + \delta W_{\text{ext}} = 0\,, \tag{4.1}$$

where W, W_{int}, and W_{ext} denote the total, internal, and external work, respectively, and δ denotes a variation with respect to the virtual displacement variables $\delta\mathbf{d}$, that is

$$\delta W = \frac{\partial W}{\partial \mathbf{d}}\delta\mathbf{d}\,. \tag{4.2}$$

The internal virtual work is defined by (see, e.g., Belytschko *et al.* [127])

$$\delta W_{\text{int}} = -\int_{\Omega_0^s} \delta\mathbf{E} : \mathbf{S}\, \mathrm{d}\Omega\,, \tag{4.3}$$

where Ω_0^s is the shell volume in the reference configuration (the total Lagrangian approach is adopted in this work), $\mathbf{E}$ is the Green–Lagrange strain tensor, $\delta\mathbf{E}$ is its variation with respect to virtual displacements $\delta\mathbf{d}$, and $\mathbf{S}$ is the energetically conjugate second Piola–Kirchhoff stress tensor.

In the case of shells, the 3D continuum description is reduced to that of the shell midsurface and the transverse normal stress is neglected. Moreover, straight lines normal to the shell midsurface remain straight and unstretched during the deformation, which implies that the transverse displacement is independent of the transverse (or through-thickness) coordinate and the transverse normal strain is zero. Furthermore, the Kirchhoff–Love theory assumes that straight lines normal to the midsurface remain normal to the midsurface during the deformation, which implies that the transverse shear strains are zero, and the deformation is due entirely to bending and in-plane stretching.

According to these assumptions, only in-plane stress and strain tensors are considered, and the indices $\alpha = 1, 2$ and $\beta = 1, 2$ are employed to denote their components. We denote by Γ_0^s the shell midsurface in the undeformed reference configuration, h_{th} is the shell thickness, and $\xi_3 \in [-0.5h_{\text{th}}, 0.5h_{\text{th}}]$ is the through-thickness coordinate. We introduce the following standard shell kinematic quantities and relationships (see, e.g., Bischoff *et al.* [69] and Kiendl [128] for more details):

$$E_{\alpha\beta} = \varepsilon_{\alpha\beta} + \xi_3 \kappa_{\alpha\beta}, \tag{4.4}$$

$$\varepsilon_{\alpha\beta} = \frac{1}{2}\left(\mathbf{g}_\alpha \cdot \mathbf{g}_\beta - \mathbf{G}_\alpha \cdot \mathbf{G}_\beta\right), \tag{4.5}$$

$$\kappa_{\alpha\beta} = \frac{\partial \mathbf{G}_\alpha}{\partial \xi_\beta} \cdot \mathbf{G}_3 - \frac{\partial \mathbf{g}_\alpha}{\partial \xi_\beta} \cdot \mathbf{g}_3, \tag{4.6}$$

$$\mathbf{g}_\alpha = \frac{\partial \mathbf{x}}{\partial \xi_\alpha}, \quad \mathbf{g}_\beta = \frac{\partial \mathbf{x}}{\partial \xi_\beta}, \tag{4.7}$$

$$\mathbf{G}_\alpha = \frac{\partial \mathbf{X}}{\partial \xi_\alpha}, \quad \mathbf{G}_\beta = \frac{\partial \mathbf{X}}{\partial \xi_\beta}, \tag{4.8}$$

$$\mathbf{g}_3 = \frac{\mathbf{g}_1 \times \mathbf{g}_2}{\|\mathbf{g}_1 \times \mathbf{g}_2\|}, \tag{4.9}$$

$$\mathbf{G}_3 = \frac{\mathbf{G}_1 \times \mathbf{G}_2}{\|\mathbf{G}_1 \times \mathbf{G}_2\|}, \tag{4.10}$$

$$\mathbf{G}^\alpha = (\mathbf{G}_\alpha \cdot \mathbf{G}_\beta)^{-1} \mathbf{G}_\beta. \tag{4.11}$$

Here, $E_{\alpha\beta}$ are the covariant components of the in-plane Green–Lagrange strain, and $\varepsilon_{\alpha\beta}$ and $\kappa_{\alpha\beta}$ are the covariant components of the membrane strain and curvature tensors, respectively, of the shell midsurface. The shell continuum is represented by the membrane and bending actions of the shell midsurface, and the strain components vary linearly through the thickness. The spatial coordinates of the shell midsurface in the current and reference configurations are $\mathbf{x} = \mathbf{x}(\xi_1, \xi_2)$ and $\mathbf{X} = \mathbf{X}(\xi_1, \xi_2)$, respectively, parameterized by ξ_1 and ξ_2. The covariant surface basis vectors in the current and reference configurations are $\mathbf{g}_\alpha$ and $\mathbf{G}_\alpha$, respectively. The unit outward normal vectors to the shell midsurface in the current and reference configurations are $\mathbf{g}_3$ and $\mathbf{G}_3$, respectively. The contravariant surface basis vectors in the reference configuration are denoted by $\mathbf{G}^\alpha$.

We select the local Cartesian basis vectors as follows:

$$\bar{\mathbf{e}}_1 = \frac{\mathbf{G}_1}{\|\mathbf{G}_1\|}, \tag{4.12}$$

$$\bar{\mathbf{e}}_2 = \frac{\mathbf{G}_2 - (\mathbf{G}_2 \cdot \bar{\mathbf{e}}_1)\bar{\mathbf{e}}_1}{\|\mathbf{G}_2 - (\mathbf{G}_2 \cdot \bar{\mathbf{e}}_1)\bar{\mathbf{e}}_1\|}, \tag{4.13}$$

that is, the first local basis vector is the normalized first covariant basis vector in the reference configuration. The local Cartesian basis vectors $\bar{\mathbf{e}}_\alpha$ are used in expressing a constitutive relationship for the shell. Because the local basis is orthonormal, we make no distinction between covariant and contravariant quantities, which are expressed with respect to it.

Remark 4.1. Alternatively, one can select the first local basis vector to be the normalized second covariant basis vector in the reference configuration. The choice of one or the other definition makes no difference in the modeling of isotropic materials. However, for the anisotropic cases, the two definitions must be distinguished.

With the above definitions, we calculate the components of the Green–Lagrange strain tensor and its variation in the local coordinate system as

$$\overline{E}_{\alpha\beta} = \overline{\varepsilon}_{\alpha\beta} + \xi_3 \overline{\kappa}_{\alpha\beta}, \tag{4.14}$$

$$\delta\overline{E}_{\alpha\beta} = \delta\overline{\varepsilon}_{\alpha\beta} + \xi_3 \delta\overline{\kappa}_{\alpha\beta}, \tag{4.15}$$

where

$$\overline{\varepsilon}_{\alpha\beta} = \varepsilon_{\gamma\delta}(\mathbf{G}^{\gamma} \cdot \overline{\mathbf{e}}_{\alpha})(\mathbf{G}^{\delta} \cdot \overline{\mathbf{e}}_{\beta}), \tag{4.16}$$

$$\overline{\kappa}_{\alpha\beta} = \kappa_{\gamma\delta}(\mathbf{G}^{\gamma} \cdot \overline{\mathbf{e}}_{\alpha})(\mathbf{G}^{\delta} \cdot \overline{\mathbf{e}}_{\beta}), \tag{4.17}$$

$$\delta\overline{\varepsilon}_{\alpha\beta} = \delta\varepsilon_{\gamma\delta}(\mathbf{G}^{\gamma} \cdot \overline{\mathbf{e}}_{\alpha})(\mathbf{G}^{\delta} \cdot \overline{\mathbf{e}}_{\beta}), \tag{4.18}$$

$$\delta\overline{\kappa}_{\alpha\beta} = \delta\kappa_{\gamma\delta}(\mathbf{G}^{\gamma} \cdot \overline{\mathbf{e}}_{\alpha})(\mathbf{G}^{\delta} \cdot \overline{\mathbf{e}}_{\beta}). \tag{4.19}$$

The variations $\delta\varepsilon_{\gamma\delta}$ and $\delta\kappa_{\gamma\delta}$ may be computed directly by taking the variational derivatives of the expressions given by Eqs. (4.5) and (4.6) with respect to the displacement vector.

Using the Voigt notation, we define the vectors of membrane strain and curvature components in the local coordinate system as

$$\overline{\boldsymbol{\varepsilon}} = \begin{bmatrix} \overline{\varepsilon}_{11} \\ \overline{\varepsilon}_{22} \\ \overline{\varepsilon}_{12} \end{bmatrix} \tag{4.20}$$

and

$$\overline{\boldsymbol{\kappa}} = \begin{bmatrix} \overline{\kappa}_{11} \\ \overline{\kappa}_{22} \\ \overline{\kappa}_{12} \end{bmatrix}, \tag{4.21}$$

together with a Green–Lagrange strain vector

$$\overline{\mathbf{E}} = \overline{\boldsymbol{\varepsilon}} + \xi_3 \overline{\boldsymbol{\kappa}}. \tag{4.22}$$

In this work, we use St. Venant–Kirchhoff material model and write the following linear stress–strain relationship in the local coordinate system:

$$\overline{\mathbf{S}} = \overline{\mathbf{C}}\,\overline{\mathbf{E}}, \tag{4.23}$$

where $\overline{\mathbf{S}}$ is a vector of components of the second Piola–Kirchhoff stress tensor in the local coordinate system, and $\overline{\mathbf{C}}$ is a constitutive material matrix, which is symmetric. Introducing Eqs. (4.22) and (4.23) into the expression for the internal virtual work given

by Eq. (4.3), we obtain

$$\delta W_{\text{int}} = -\int_{\Omega_0^s} \delta\overline{\mathbf{E}} \cdot \overline{\mathbf{S}}\, \mathrm{d}\Omega \tag{4.24}$$

$$= -\int_{\Gamma_0^s} \left(\int_{h_{\text{th}}} \delta\overline{\mathbf{E}} \cdot \overline{\mathbf{C}}\, \overline{\mathbf{E}}\, d\xi_3 \right) \mathrm{d}\Gamma \tag{4.25}$$

$$\begin{aligned} = &-\int_{\Gamma_0^s} \delta\overline{\boldsymbol{\varepsilon}} \cdot \left[\left(\int_{h_{\text{th}}} \overline{\mathbf{C}}\, d\xi_3 \right) \overline{\boldsymbol{\varepsilon}} + \left(\int_{h_{\text{th}}} \overline{\mathbf{C}}\, \xi_3 d\xi_3 \right) \overline{\boldsymbol{\kappa}} \right] \mathrm{d}\Gamma \\ &-\int_{\Gamma_0^s} \delta\overline{\boldsymbol{\kappa}} \cdot \left[\left(\int_{h_{\text{th}}} \overline{\mathbf{C}}\, \xi_3 d\xi_3 \right) \overline{\boldsymbol{\varepsilon}} + \left(\int_{h_{\text{th}}} \overline{\mathbf{C}}\, \xi_3^2 d\xi_3 \right) \overline{\boldsymbol{\kappa}} \right] \mathrm{d}\Gamma. \end{aligned} \tag{4.26}$$

For a general orthotropic material,

$$\overline{\mathbf{C}}_{\text{ort}} = \begin{bmatrix} \dfrac{E_1}{(1-\nu_{12}\nu_{21})} & \dfrac{\nu_{21}E_1}{(1-\nu_{12}\nu_{21})} & 0 \\ \dfrac{\nu_{12}E_2}{(1-\nu_{12}\nu_{21})} & \dfrac{E_2}{(1-\nu_{12}\nu_{21})} & 0 \\ 0 & 0 & G_{12} \end{bmatrix}, \tag{4.27}$$

where E_1 and E_2 are the Young's moduli in the directions defined by the local basis vectors, ν_{12} and ν_{21} are the Poisson ratios, G_{12} is the shear modulus, and $\nu_{21}E_1 = \nu_{12}E_2$ to ensure the symmetry of the constitutive material matrix $\overline{\mathbf{C}}_{\text{ort}}$. In the case of an isotropic material, $E_1 = E_2 = E$, $\nu_{21} = \nu_{12} = \nu$, and $G_{12} = E/(2(1+\nu))$.

4.1.2 Laminated Shells

In the case of composite materials, which are used in the manufacturing of modern wind turbine blades, we assume that the structure is composed of a set of plies (or laminae), each modeled as an orthotropic material. We use the classical laminated plate theory (see, e.g., Reddy [129]), and homogenize the material through-thickness constitutive behavior for a given composite laminate lay-up. Let k denote the kth ply and n be the total number of plies (see Figure 4.1). Pre-integrating through the shell thickness in Eq. (4.26) and assuming each ply has the same thickness h_{th}/n, the extensional

(membrane) stiffness **A**, coupling stiffness **B**, and bending stiffness **D** are given by

$$\mathbf{A} = \int_{h_{\text{th}}} \overline{\mathbf{C}}\, d\xi_3 = \frac{h_{\text{th}}}{n} \sum_{k=1}^{n} \overline{\mathbf{C}}^{(k)}, \tag{4.28}$$

$$\mathbf{B} = \int_{h_{\text{th}}} \overline{\mathbf{C}}\, \xi_3 d\xi_3 = \left(\frac{h_{\text{th}}}{n}\right)^2 \sum_{k=1}^{n} \overline{\mathbf{C}}^{(k)} \left(k - \frac{n}{2} - \frac{1}{2}\right), \tag{4.29}$$

$$\mathbf{D} = \int_{h_{\text{th}}} \overline{\mathbf{C}}\, \xi_3^2 d\xi_3 = \left(\frac{h_{\text{th}}}{n}\right)^3 \sum_{k=1}^{n} \overline{\mathbf{C}}^{(k)} \left(\left(k - \frac{n}{2} - \frac{1}{2}\right)^2 + \frac{1}{12}\right), \tag{4.30}$$

where

$$\overline{\mathbf{C}}^{(k)} = \mathbf{T}^T(\phi^{(k)})\, \overline{\mathbf{C}}^{(k)}_{\text{ort}}\, \mathbf{T}(\phi^{(k)}), \tag{4.31}$$

$$\mathbf{T}(\phi) = \begin{bmatrix} \cos^2\phi & \sin^2\phi & \sin\phi\cos\phi \\ \sin^2\phi & \cos^2\phi & -\sin\phi\cos\phi \\ -2\sin\phi\cos\phi & 2\sin\phi\cos\phi & \cos^2\phi - \sin^2\phi \end{bmatrix}, \tag{4.32}$$

and ϕ is the fiber orientation angle in each ply. Eq. (4.31) transforms $\overline{\mathbf{C}}^{(k)}_{\text{ort}}$ from the principal material coordinates to the laminate coordinates (defined by the local Cartesian basis) for each ply. Superscript "(k)" indicates that the orthotropic material matrix, in general, can be different from ply to ply. The transformed local material matrix $\overline{\mathbf{C}}^{(k)}$ is constant within each ply.

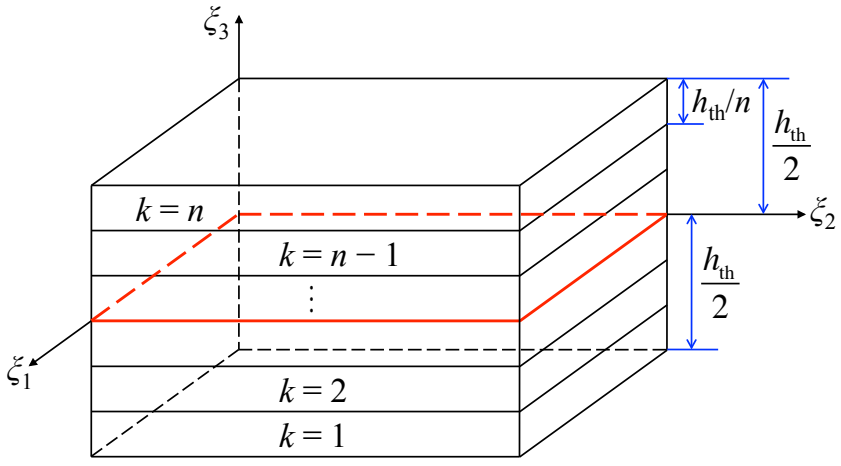


Figure 4.1: Schematic of a composite laminate.

With the above definitions, the expression for the internal virtual work for a composite shell may now be compactly written as

$$\delta W_{\text{int}} = -\int_{\Gamma_0^s} \delta\overline{\boldsymbol{\varepsilon}} \cdot (\mathbf{A}\,\overline{\boldsymbol{\varepsilon}} + \mathbf{B}\,\overline{\boldsymbol{\kappa}})\,\mathrm{d}\Gamma - \int_{\Gamma_0^s} \delta\overline{\boldsymbol{\kappa}} \cdot (\mathbf{B}\,\overline{\boldsymbol{\varepsilon}} + \mathbf{D}\,\overline{\boldsymbol{\kappa}})\,\mathrm{d}\Gamma. \tag{4.33}$$

Remark 4.2. Setting $n = 1$ and $\overline{\mathbf{C}}^{(k)} = \overline{\mathbf{C}}_{\text{ort}}$ in Eqs. (4.28)–(4.30), we get $\mathbf{B} = \mathbf{0}$ and

$$\mathbf{A} = h_{\text{th}}\overline{\mathbf{C}}_{\text{ort}}, \tag{4.34}$$

$$\mathbf{D} = \frac{h_{\text{th}}^3}{12}\overline{\mathbf{C}}_{\text{ort}}, \tag{4.35}$$

which are the classical membrane and bending stiffnesses for an orthotropic shell.

4.1.3 The Bending Strip Method and the Complete Variational Statement of the Structural Mechanics Problem

The expression for the internal virtual work given by Eq. (4.33) is only meaningful when the shell midsurface is described using a smooth geometrical mapping. In the case when the regularity of the mapping reduces to the C^0 level, the terms involving the curvature tensors, which rely on the second derivatives of the geometrical mapping, lead to non-integrable singularities, and the formulation may not be used as is. However, for complex structures, the geometry definition often requires that the continuity of the geometrical mapping is reduced to the C^0 level (e.g., the trailing edge of an airfoil or an I-beam, the latter being a non-manifold surface).

A "bending strip method" was purposed[1] to handle complex multi-patch shell structures in the context of the rotation-free Kirchhoff–Love theory. The main idea behind the method, illustrated in Figure 4.2, consists of the following. It is assumed that the shell structure is comprised of smooth subdomains, such as NURBS patches, that are joined with C^0-continuity. In addition, thin strips of fictitious material, also modeled as surface NURBS patches, are placed at patch intersections. The triples of control points at the patch interface, consisting of a shared control point and one on each side, are

[1]The method was implemented and tested with J. Kiendl, at the time a Ph.D. student in the group of Prof. K.-U. Bletzinger at the Technische Universität München, during his visit to the research group of Prof. Y. Bazilevs at the University of California, San Diego.

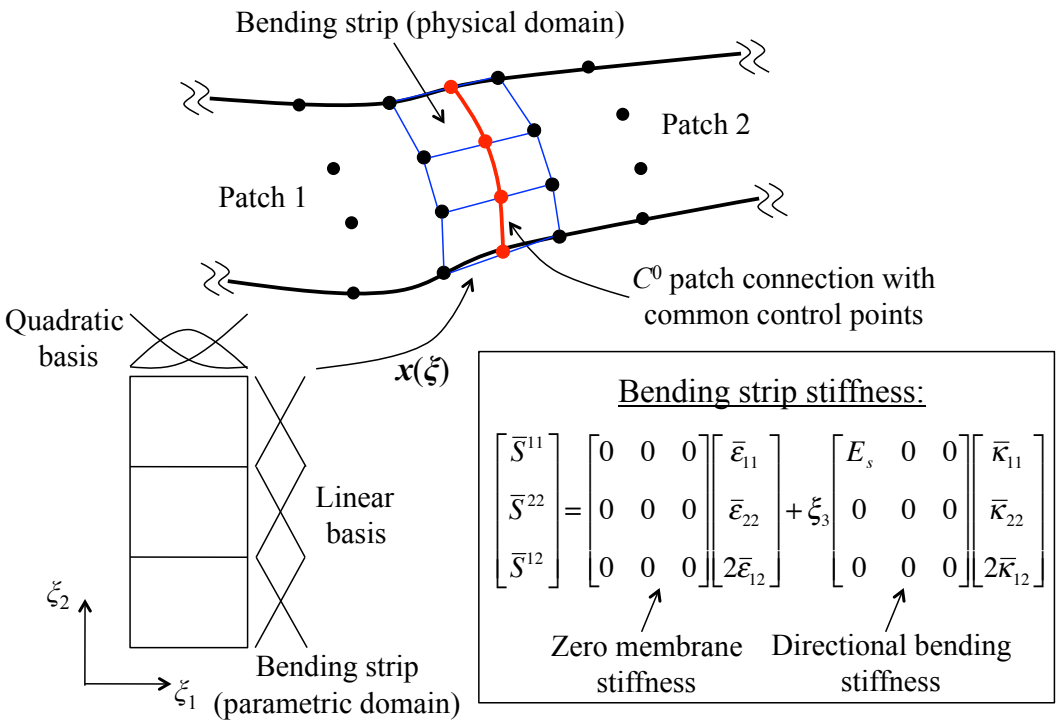


Figure 4.2: Schematic of the bending strip method.

extracted and used as a control mesh for the bending strips. The parametric domain of each bending strip consists of one quadratic element in the direction transverse to the interface and, for simplicity and computational efficiency, of as many linear elements as necessary to accommodate all the control points along the length of the strip. The material is assumed to have zero mass, zero membrane stiffness, and non-zero bending stiffness only in the direction transverse to the interface. The transverse direction may be obtained using the local basis construction given by Eqs. (4.12) and (4.13), however, other options may be explored.

Let Γ_0^s and Γ_t^s denote the structure midsurface in the reference and deformed configurations, respectively, and let Γ_0^b denote the bending strip domain, where Γ_0^b is a union of the bending strip patch subdomains. Let $\mathcal{S}_d^h$ and $\mathcal{V}_d^h$ denote the discrete trial and test function spaces for the structural problem. We seek the displacement of the shell midsurface $\mathbf{d}^h \in \mathcal{S}_d^h$, such that $\forall \delta\mathbf{d}^h \in \mathcal{V}_d^h$:

$$B(\delta\mathbf{d}^h, \mathbf{d}^h) - F(\delta\mathbf{d}^h) = 0, \tag{4.36}$$

where

$$B(\delta\mathbf{d}^h, \mathbf{d}^h) = \int_{\Gamma_t^s} \delta\mathbf{d}^h \cdot \rho h_{\text{th}} \left.\frac{\partial^2 \mathbf{d}^h}{\partial t^2}\right|_{\mathbf{X}} \mathrm{d}\Gamma + \int_{\Gamma_0^s} \delta\overline{\boldsymbol{\varepsilon}}^h \cdot \left(\mathbf{A}\,\overline{\boldsymbol{\varepsilon}}^h + \mathbf{B}\,\overline{\boldsymbol{\kappa}}^h\right) \mathrm{d}\Gamma + \int_{\Gamma_0^s} \delta\overline{\boldsymbol{\kappa}}^h \cdot \left(\mathbf{B}\,\overline{\boldsymbol{\varepsilon}}^h + \mathbf{D}\,\overline{\boldsymbol{\kappa}}^h\right) \mathrm{d}\Gamma + \int_{\Gamma_0^b} \delta\overline{\boldsymbol{\kappa}}^h \cdot \mathbf{D}_{\text{b}}\,\overline{\boldsymbol{\kappa}}^h \,\mathrm{d}\Gamma \tag{4.37}$$

and

$$F(\delta\mathbf{d}^h) = \int_{\Gamma_t^s} \delta\mathbf{d}^h \cdot \rho h_{\text{th}} \mathbf{f} \,\mathrm{d}\Gamma + \int_{(\Gamma_t^s)_{\text{h}}} \delta\mathbf{d}^h \cdot \mathbf{h}^h \,\mathrm{d}\Gamma. \tag{4.38}$$

In the above, the superscript h denotes all the discrete quantities, ρ is the structural mass density in the deformed configuration, $\mathbf{f}$ is the body force (e.g., gravity), and $\mathbf{h}$ is the prescribed surface traction on $(\Gamma_t^s)_{\text{h}}$. The first term on the right-hand-side of Eq. (4.37) represents the inertial force. The terms on the second line of Eq. (4.37) were derived in the previous section. The last term is penalty-like and represents the contribution of the bending strips to the structural formulation given by Eq. (4.36). Here $\mathbf{D}_{\text{b}}$ is the bending stiffness of the bending strips:

$$\mathbf{D}_{\text{b}} = \frac{h_{\text{th}}^3}{12}\mathbf{C}_{\text{b}}, \tag{4.39}$$

where

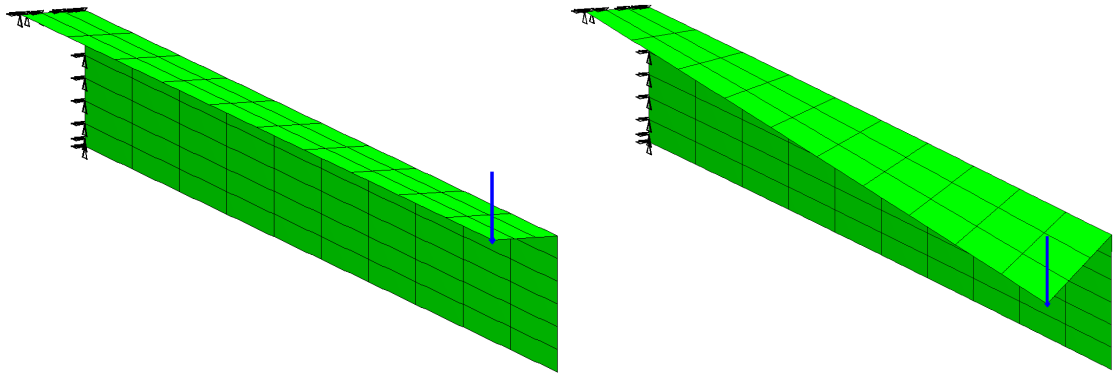
$$\mathbf{C}_{\text{b}} = \begin{bmatrix} E_{\text{b}} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{4.40}$$

and E_{b} is the scalar bending strip stiffness, typically chosen as a multiple of the local Young's modulus of the shell. This design of the material constitutive matrix ensures that the bending strips add no extra stiffness to the structure. They only penalize the change in the angle during the deformation between the triples of control points at the patch interface. The stiffness E_{b} must be high enough so that the change in angle is within an acceptable tolerance. However, if E_{b} is chosen too high, the global stiffness matrix becomes badly conditioned, which may lead to divergence in the computations.

Remark 4.3. Because of the structure of the bending strip term in Eq. (4.37), the method may be interpreted as a physically-motivated penalty formulation.

Remark 4.4. In IGA, the possibility to employ smooth surface descriptions directly in analysis has led to the development of new shell element formulations. Besides the references cited in this section, the reader is referred to Cirak *et al.* [71–73] and Benson *et al.* [50, 51] for relevant work on shells.

The method is illustrated by an example of the L-shape cantilever with a point load. Figure 4.3(a) shows the geometry consisting of two rectangular patches meeting at a 90° angle, which corresponds to the connection with a kink. Figure 4.3(b) shows the deformed configuration for the case when the two patches are connected with C^0-continuity and the Kirchhoff–Love shell formulation is used without additional treatment (i.e., without using the bending strip). The result shows that the C^0-continuous connection acts like a hinge between the patches and no bending moment is transferred.

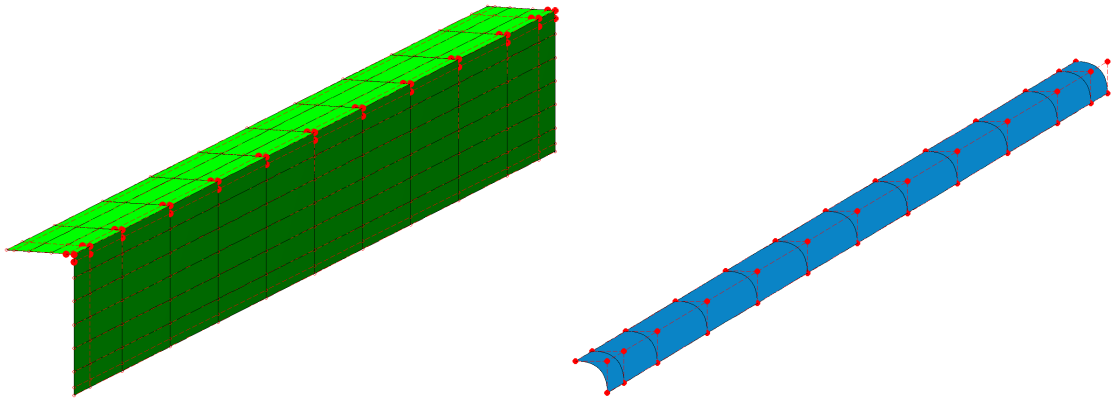


(a) (b)

Figure 4.3: L-shaped cantilever with a point load. (a) Two rectangular patches meeting at a 90° angle. (b) Deformed configuration with no bending strip. The connection acts like a hinge.

To correctly transfer the bending moments between the patches, a bending strip patch is added into the problem. Figure 4.4(a) shows the control points to be coupled by the bending strip and Figure 4.4(b) shows the bending strip patch built using these control points. The resulting deformation with the bending strip is shown in Figure 4.5 and the angle between the patches remains nearly constant during deformation.

To examine the effect on the solution for different choice of bending strip stiffness, we compute the L-shape cantilever with bending strip for a range of stiffness values and record the change in the angle between the two patches. The bending strip stiffness



(a) (b)

Figure 4.4: L-shaped cantilever with a point load. (a) Structure with control points coupled to a bending strip highlighted. (b) Bending strip corresponding to an L-shaped configuration.

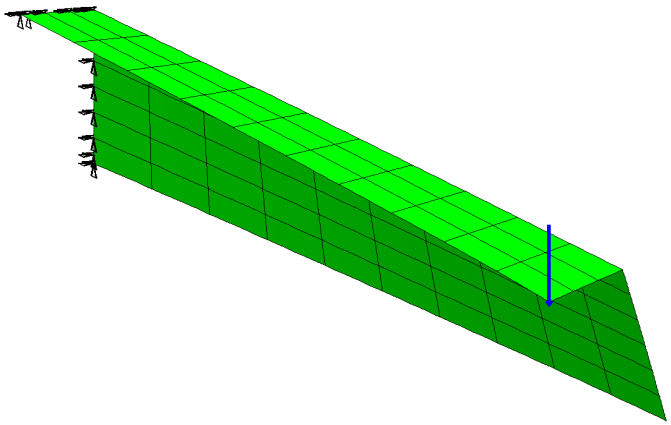


Figure 4.5: L-shaped cantilever with a point load. Deformed configuration with a bending strip. Angle between the patches remains nearly constant throughout the deformation.

is chosen as $E_b = E \times 10^{\alpha}$, where $\alpha = 0, 1, 2, ..., 10$. The results are shown in Figure 4.6. As expected, the change in the angle decreases when E_b is increased. For stiffness ratio $E_b/E = 10^3$, the relative error is less than 10^{-3}. Further computations reveal that the solution becomes unstable for stiffness ratio $E_b/E > 10^{13}$. This suggests that there is a relatively large range of E_b for which the bending strip method yields correct and stable solutions.

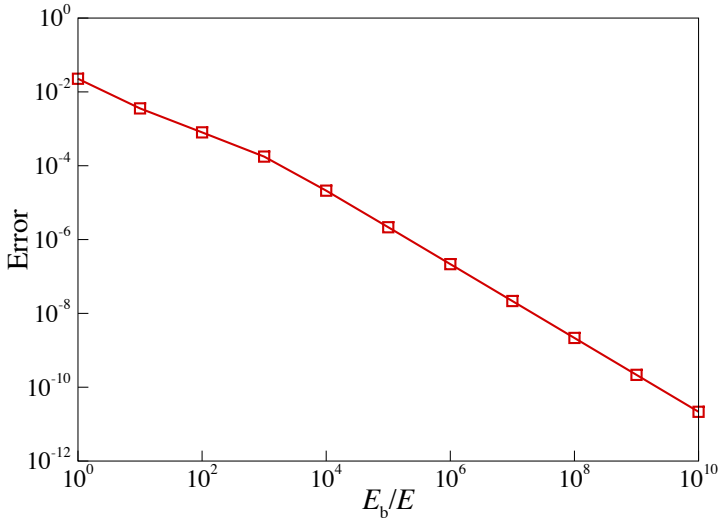


Figure 4.6: L-shape cantilever. Relative change in the angle between two patched at the cantilever tip as a function of the bending strip stiffness.

4.2 Numerical Examples

4.2.1 Scordelis–Lo Roof

This exmaple is taken from the shell obstacle course proposed by Belytschko *et al.* [130]. The Scordelis–Lo roof problem consists of a section of a cylindrical shell subjected to a uniform gravity load. It is supported by rigid diaphragms at its ends and free at the side edges. Figure 4.7 shows the problem geometry and material parameters. Linearized theory is assumed in this example, and the vertical displacement at the midpoint of the side edge reported by Belytschko *et al.* [130] is taken as a reference solution.

In this work the problem is modeled using multi-patch shells with bending strips. Two lines of C^0-continuity are created by repeated knot insertions. Figure 4.8(a) shows

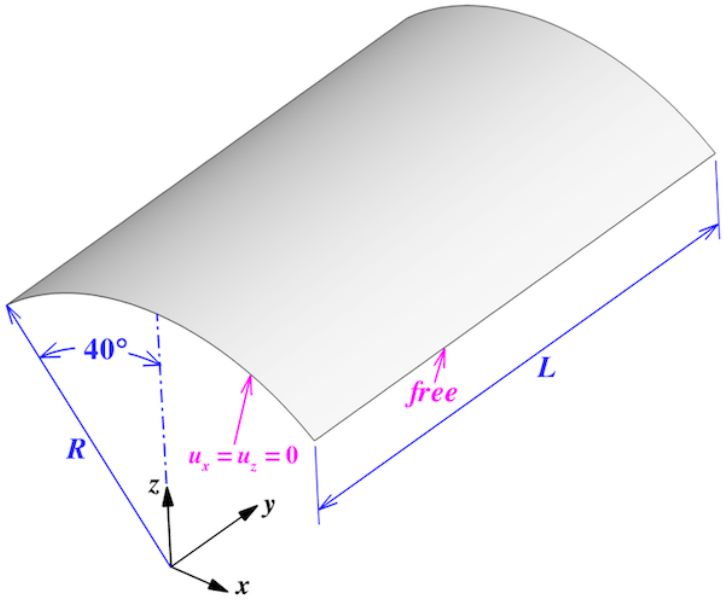
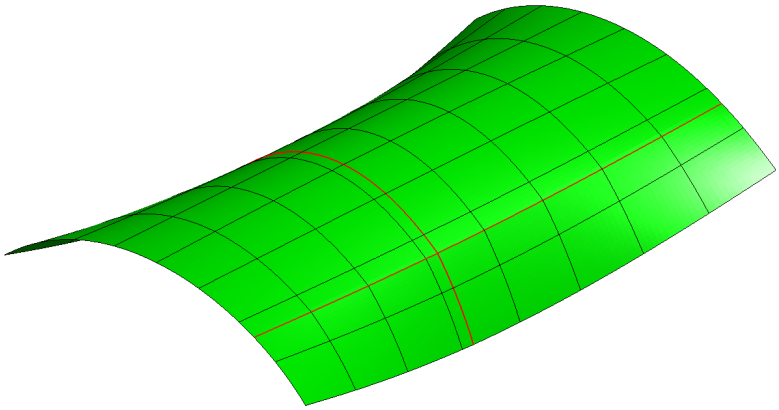


Figure 4.7: Scordelis–Lo roof. Problem description. The roof is subjected to a uniform vertical gravity load of 90.0 per unit area. The ends are supported by rigid diaphragms.

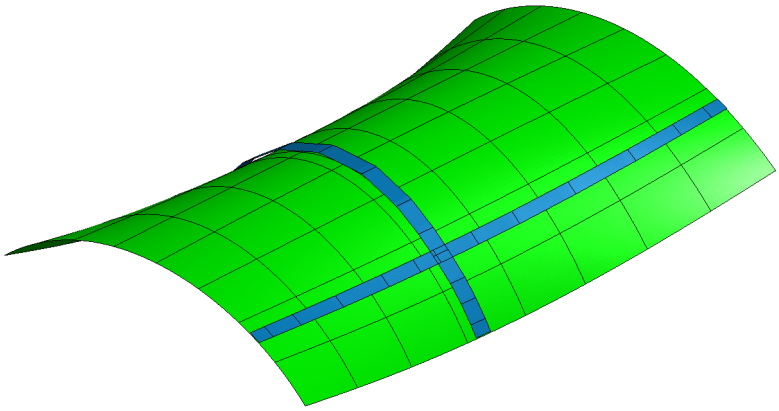
the roof with C^0-continuity marked by red lines. Bending strips with $E_b = E \times 10^3$ are added along the C^0-continuity lines and the two strips overlap with each other (see Figure 4.8(b)). A k-refinement study of the roof with the bending strips is performed and the convergence of the vertical displacement is shown in Figure 4.9.

Remark 4.5. The bending strips are re-created for every new mesh in the refinement sequence according to its design. The cost of creating the new strips is negligible in comparison with the costs associated with mesh refinement. Due to the rectangular topology of NURBS patches, the bending strip control points may be easily extracted from the adjoining patches. The polynomial order of the strip is fixed at quadratic and linear in the transverse and longitudinal directions, respectively, independent of the polynomial order of the structural model. The width of the bending strip tends to zero as the mesh is h-refined.

Remark 4.6. Comparing the convergence results with the single-patch computations reported by Kiendl *et al.* [52] shows that similar convergence behavior is observed for both single- and multi-patch computations, indicating that the analysis results are not sensitive to the presence of the bending strips.



(a)



(b)

Figure 4.8: Scordelis–Lo roof. (a) Decomposition of the problem domain into four patches. (b) Two overlapping bending strips used in the computation. Both are plotted in the deformed configuration.

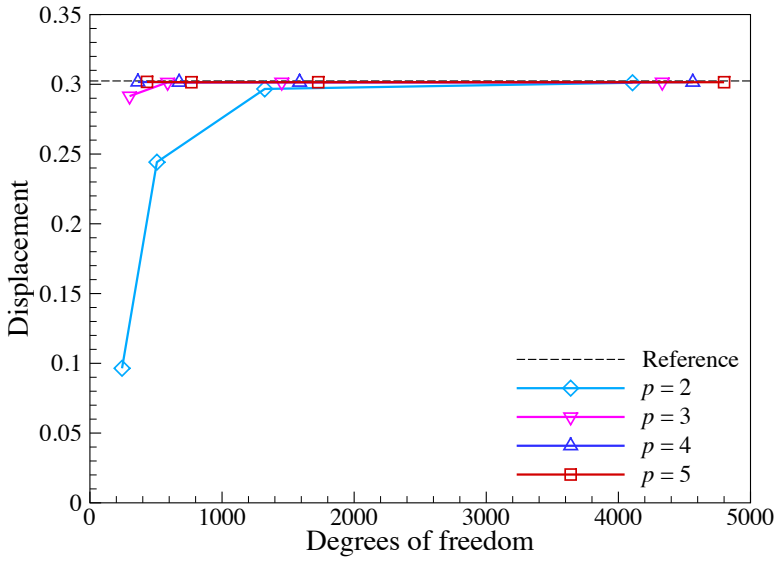


Figure 4.9: Scordelis–Lo roof. k-refinement study with two overlapping bending strips.

Remark 4.7. An important feature of the proposed approach is that no interaction between the overlapping strips is introduced, making the method practical for large structures where more than two patches may meet at a point.

4.2.2 Geometrically Nonlinear Structures

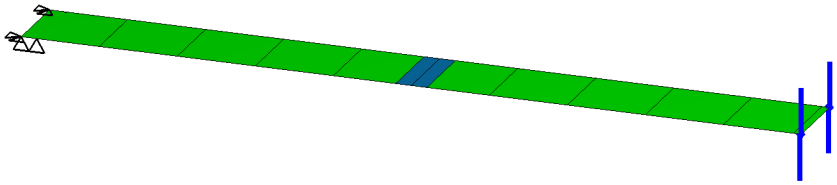
In this section we apply the developed method to two geometrically nonlinear benchmark problems. Both examples involve a cantilever plate under constant moment loading. In the first case the plate is under pure bending and the second case the plate is under pure twisting moment.

Cantilever Plate under Bending

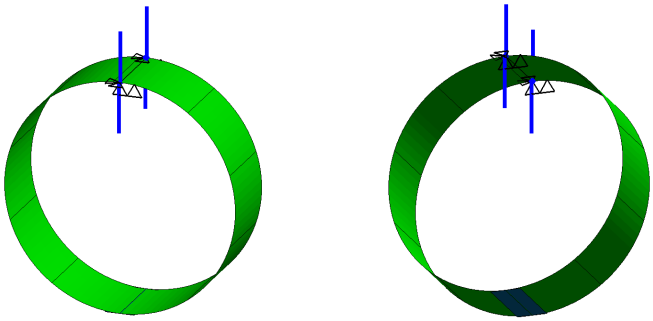
A cantilever plate is used to test the formulation in the geometrically nonlinear regime. The problem setup is illustrated in Figure 4.10(a). An external moment $M = 2\pi EI/L$ is applied at the tip of the plate. E, I and L are the Youngs modulus, moment of inertia and length, respectively, of the plate. The expected plate deformation is such that the final configuration is a circular shape. The moment is modeled by two pairs of forces in opposite directions, orthogonal to the shell, following the deformation. The plate is

discretized by ten quintic NURBS elements of full continuity and split into two patches by introducing a C^0 line in the middle of the plate. The bending strip with stiffness $E_b = E \times 10^3$ is added to the structure.

Figure 4.10(b) shows that the cantilever plate is deformed into a circular ring. The correct result is obtained and the applicability of the bending strips for large displacement and rotation is demonstrated.



(a) Initial configuration



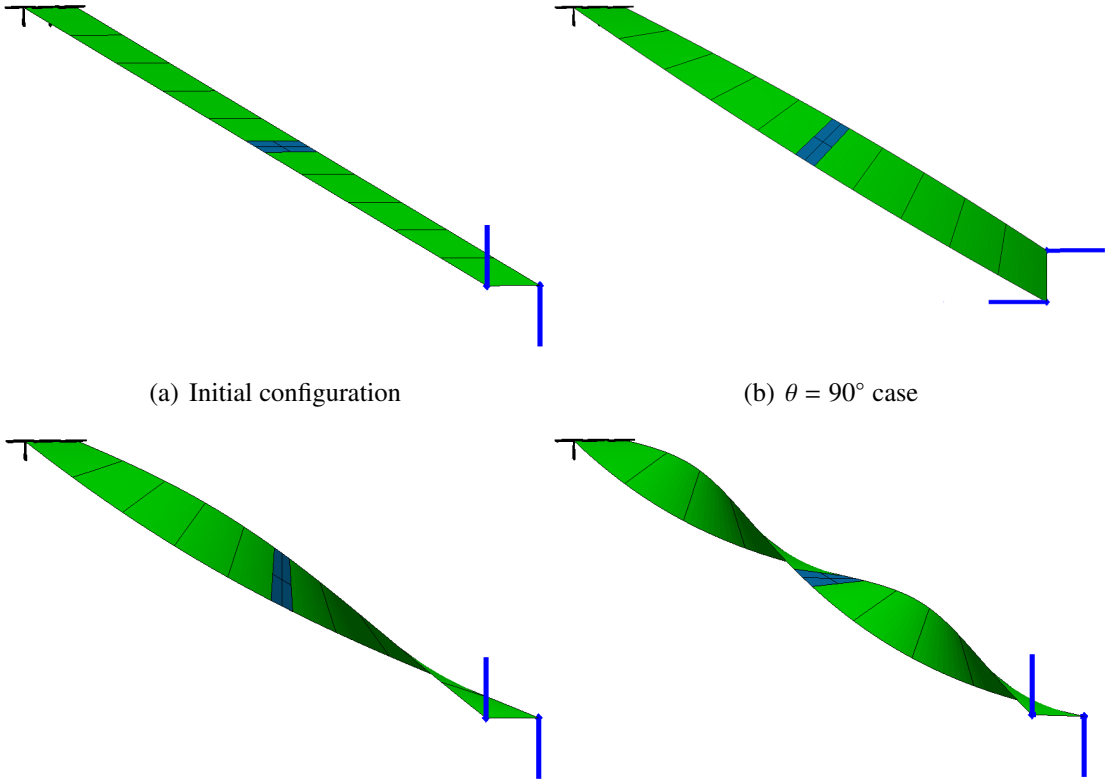
(b) Deformed configuration

Figure 4.10: Cantilever plate under bending. The bending moment is represented by two pairs of follower forces acting in the opposite directions.

Cantilever Plate under Torsion

Here the cantilever plate is subjected to a pure twisting moment, which is represented by a pair of follower forces in opposite directions. From the beam theory, $T = \theta GJ/L$, where T is the twisting moment (torsion), θ is the twist angle at the tip, G is the shear modulus, J is the torsion constant, and L is the plate length. The torsion constant is defined as $J = bt^3/3$, where b and t are the width and thickness of the plate, respectively. We consider $\theta = 90°$, $180°$ and $360°$ and apply the corresponding torsions at the plate tip. The twist angle of the deformed shape is compared with the expected

values. The results are shown in Figure 4.11 and the correct deformations are obtained for all cases.



(a) Initial configuration

(b) $\theta = 90°$ case

(c) $\theta = 180°$ case

(d) $\theta = 360°$ case

Figure 4.11: Cantilever plate under torsion. The twisting moment is represented by a pair of follower forces acting in opposite directions.

4.2.3 Application to the Wind Turbine Blade

The following illustrates the use of the bending strip method applied to the construction of the structural model of the NREL 5 MW offshore baseline wind turbine blade. A symmetric fiberglass/epoxy composite with $[\pm 45/0/90_2/0_3]_s$ lay-up, which enhances the flapwise and edgewise stiffness, is considered for the wind turbine blade material. The 0° fiber points in the direction of a tangent vector to the airfoil cross-section curve. The orthotropic elastic moduli for each ply are given in Table 4.1. For simplicity, the entire blade is assumed to have the same lay-up. The resulting **A**, **B**, and

D matrices from Eqs. (4.28)–(4.30) are

$$\mathbf{A} = h_{\text{th}} \begin{bmatrix} 26.315 & 4.221 & 0 \\ 4.221 & 18.581 & 0 \\ 0 & 0 & 5.571 \end{bmatrix} \times 10^9 \,(\text{N/m})\,, \tag{4.41}$$

$$\mathbf{B} = \mathbf{0}, \tag{4.42}$$

$$\mathbf{D} = h_{\text{th}}^3 \begin{bmatrix} 1.727 & 0.545 & 0.053 \\ 0.545 & 1.627 & 0.053 \\ 0.053 & 0.053 & 0.658 \end{bmatrix} \times 10^9 \,(\text{N m})\,, \tag{4.43}$$

and the total laminate thickness distribution is shown in Figures 4.12 and 4.13(a). The thickness varies linearly between eight blade radial locations defined in Table 4.2. The blade shell model together with the bending strips covering the regions of C^0-continuity is shown in Figure 4.13(b).

We perform the elastostatic analysis of the wind turbine blade using the proposed isogeometric Kirchhoff–Love composite shell formulation. The blade is clamped at the root and subjected to the gravity load in the flapwise direction. We consider the anisotropic case corresponds to the aforementioned composite, as well as an isotropic case, where the Young's modulus E and the Poisson ratio ν are set to 19 GPa and 0.29, respectively. Both geometrically linear and nonlinear computations are performed and the results are shown in Table 4.3. Figure 4.13(c) shows the deformed shape of the blade for the nonlinear anisotropic (composite) case.

The FSI simulations with this blade model will be presented in the next chapter. Further examples of calculations using the bending strip method, which include the possibility of using the approach for simple coupling of solids and shells, may be found in Kiendl *et al.* [53].

Table 4.1: Material properties of a unidirectional fiberglass/epoxy composite taken from Daniel and Ishai [131].

E_1 (GPa)	E_2 (GPa)	G_{12} (GPa)	ν_{12} (–)	ρ (g/cm^3)
39	8.6	3.8	0.28	2.1

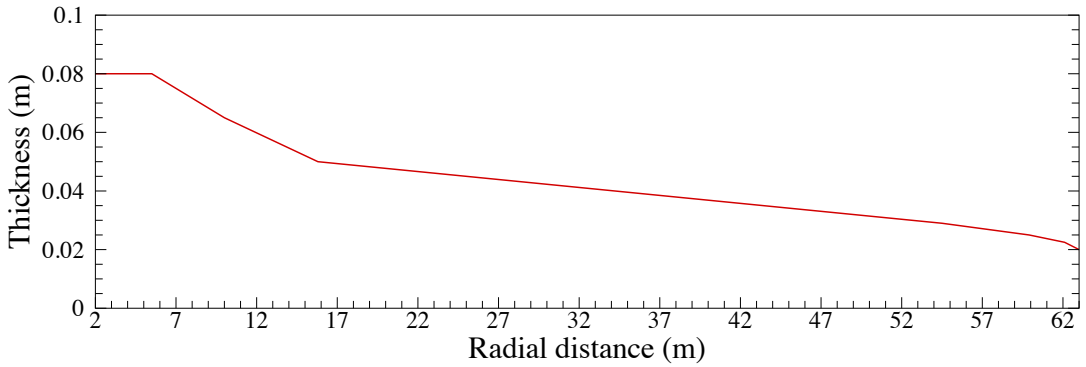


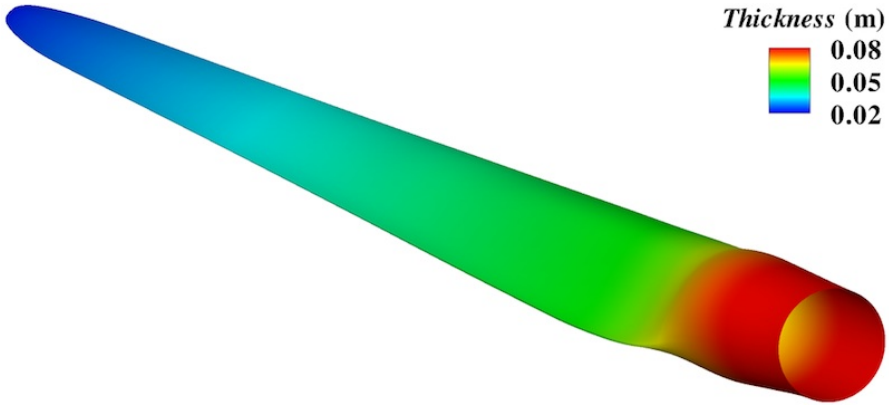
Figure 4.12: Shell thickness of the wind turbine blade. The thickness varies linearly between eight blade radial locations defined in Table 4.2.

Table 4.2: Shell thickness of the wind turbine blade. The thickness is defined at eight blade radial locations and varies linearly between them.

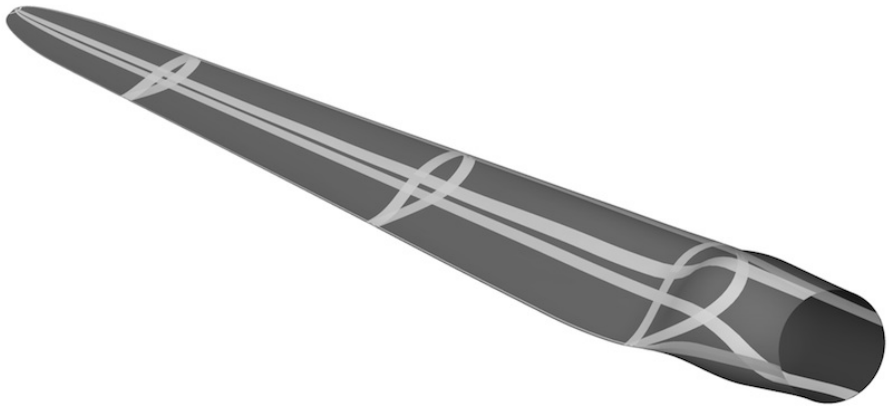
	Radial distance (m)	Thickness (cm)
Root	2.0	8.00
	5.5	8.00
	10.0	6.50
	15.8	5.00
	54.5	2.90
	59.9	2.50
	62.1	2.25
Tip	63.0	2.00

Table 4.3: Results of the blade tip deflection under gravity load.

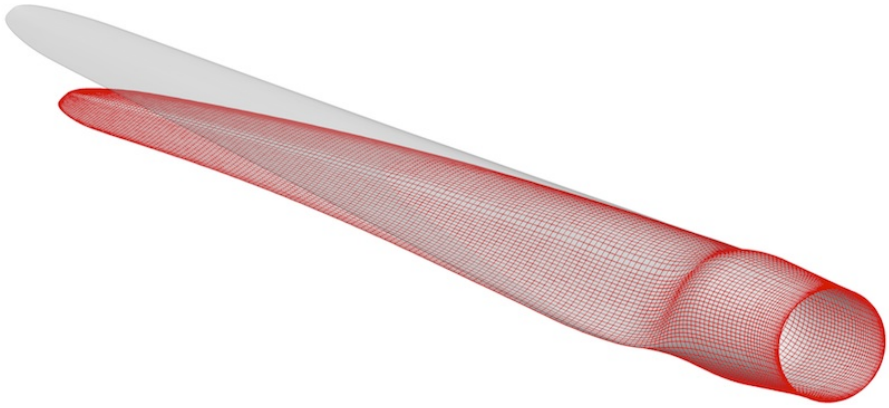
	Isotropic	Anisotropic
Linear	3.010 m	3.194 m
Nonlinear	3.045 m	3.235 m



(a) Shell thickness distribution



(b) A 20-patch NURBS model with bending strips



(c) Reference and deformed configurations

Figure 4.13: NREL 5 MW offshore baseline wind turbine rotor blade model.

4.2.4 Effect of the Transverse Shear

For thin shell structures the transverse shear stresses are much smaller than the in-plane stresses. However, they play an important role in the onset and growth of damage and delamination in composite structures (see, e.g., Reddy [129]). To study the effect of transverse shear, we perform elastostatic analysis of the wind turbine blade using the isogeometric Kirchhoff–Love shell formulation, where the transverse shear is zero, and compare our results with an LS-DYNA [132] FEM simulation that makes use of a Reissner–Mindlin shell with rotational degrees of freedom (see, e.g., Benson *et al.* [50]). In the latter case the transverse shear effect is taken into account.

We consider two cases that one uses isotropic material and the other uses orthotropic material. For the isotropic case, the Young's modulus E and the Poisson ratio ν are set to 19 GPa and 0.29, respectively. The anisotropic case corresponds to a symmetric fiberglass/epoxy composite with $[\pm 45/90/0_2]_s$ lay-up. The orthotropic elastic moduli for each ply are given in Table 4.1. The entire blade is assumed to have the same lay-up and a variable composite thickness is considered from root to tip (see Figure 4.12 and Table 4.2). The blade aerodynamic loads are obtained from a separate CFD simulation, reported in Section 3.3.2, for which the inflow wind speed is assumed uniform at 11.4 m/s, and the rotor speed is 1.267 rad/s.

Comparison of the tip deflection between the two shell formulations is shown in Table 4.4. The difference in tip deflection is less than 1%. Figure 4.14 shows the overlapped deformed configuration obtained using the rotation-free shell formulation and that with rotational degrees of freedom. Both deformed configurations are nearly indistinguishable. These results suggest that the use of a thin shell theory is suitable for this application, and that the transverse shear effect is not as important for accurate prediction of composite blade deformation under the action of wind and inertial (centripetal) loads.

Table 4.4: Comparison of the blade tip deflection between isogeometric Kirchhoff–Love (IGA-KL) and LS-DYNA Reissner–Mindlin shell analysis.

	Isotropic	Anisotropic
IGA-KL shell	3.48 m	3.73 m
LS-DYNA	3.51 m	3.76 m

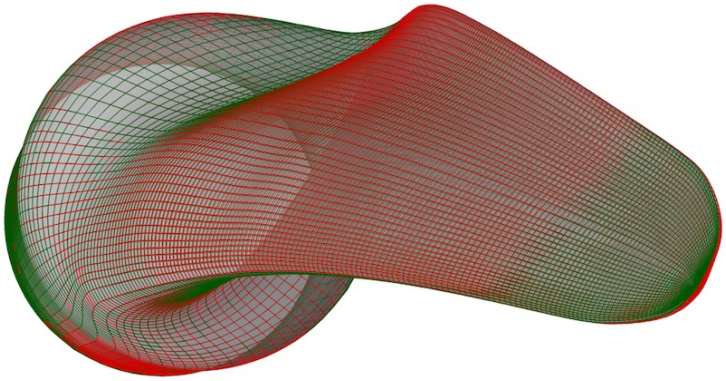


Figure 4.14: Comparison of the deformed blade shapes predicted by isogeometric Kirchhoff–Love (IGA-KL) and LS-DYNA Reissner–Mindlin shell analysis.

4.3 Time Integration of the Structural Mechanics Equations

In the case of wind turbine rotors, the structural motions are dominated by the rotation of the blades around the hub axis. Here, we take advantage of this fact and propose a class of standard time integration techniques to exactly account for the rotational part of the structural motion.

For this, as a first step, it is useful to decompose the structural displacement $\mathbf{d}$ into its rotation and deflection components as

$$\mathbf{d} = \mathbf{d}_\theta + \mathbf{d}_d. \tag{4.44}$$

The rotational component of the displacement may be computed as

$$\mathbf{d}_\theta = (\mathbf{R}(\theta) - \mathbf{I})\,(\mathbf{X} - \mathbf{X}_0)\,, \tag{4.45}$$

where $\mathbf{X}$ are the coordinates of the structure reference configuration, $\mathbf{X}_0$ is a fixed point, θ is the time-dependent angle of rotation, $\mathbf{R}(\theta)$ is the rotation matrix, and $\mathbf{I}$ is the identity matrix. We specialize to the case of rotation about the z-axis, which gives

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.46}$$

The total structural velocity and acceleration may be computed as

$$\frac{\partial \mathbf{d}}{\partial t} = \dot{\mathbf{d}} = \dot{\mathbf{d}}_\theta + \dot{\mathbf{d}}_d = \dot{\mathbf{R}}(\theta)(\mathbf{X} - \mathbf{X}_0) + \dot{\mathbf{d}}_d, \tag{4.47}$$

$$\frac{\partial^2 \mathbf{d}}{\partial t^2} = \ddot{\mathbf{d}} = \ddot{\mathbf{d}}_\theta + \ddot{\mathbf{d}}_d = \ddot{\mathbf{R}}(\theta)(\mathbf{X} - \mathbf{X}_0) + \ddot{\mathbf{d}}_d, \tag{4.48}$$

where

$$\dot{\mathbf{R}}(\theta) = \begin{bmatrix} -\sin\theta & -\cos\theta & 0 \\ \cos\theta & -\sin\theta & 0 \\ 0 & 0 & 0 \end{bmatrix} \dot{\theta}, \tag{4.49}$$

$$\ddot{\mathbf{R}}(\theta) = \begin{bmatrix} -\cos\theta & \sin\theta & 0 \\ -\sin\theta & -\cos\theta & 0 \\ 0 & 0 & 0 \end{bmatrix} \dot{\theta}^2 + \begin{bmatrix} -\sin\theta & -\cos\theta & 0 \\ \cos\theta & -\sin\theta & 0 \\ 0 & 0 & 0 \end{bmatrix} \ddot{\theta}. \tag{4.50}$$

We repeat this decomposition at the discrete level, where we operate directly on the nodal or control-point displacement degrees of freedom. For this, we let $\mathbf{U}$, $\dot{\mathbf{U}}$, and $\ddot{\mathbf{U}}$ be the vectors of nodal or control-point displacements, velocities, and accelerations, respectively. We set

$$\mathbf{U} = \mathbf{U}_\theta + \mathbf{U}_d, \tag{4.51}$$

$$\dot{\mathbf{U}} = \dot{\mathbf{U}}_\theta + \dot{\mathbf{U}}_d, \tag{4.52}$$

$$\ddot{\mathbf{U}} = \ddot{\mathbf{U}}_\theta + \ddot{\mathbf{U}}_d, \tag{4.53}$$

where $\mathbf{U}_\theta$, $\dot{\mathbf{U}}_\theta$, and $\ddot{\mathbf{U}}_\theta$ are given by

$$\mathbf{U}_\theta = (\mathbf{R}(\theta) - \mathbf{I})\,(\mathbf{X} - \mathbf{X}_0), \tag{4.54}$$

$$\dot{\mathbf{U}}_\theta = \dot{\mathbf{R}}(\theta)(\mathbf{X} - \mathbf{X}_0), \tag{4.55}$$

$$\ddot{\mathbf{U}}_\theta = \ddot{\mathbf{R}}(\theta)(\mathbf{X} - \mathbf{X}_0). \tag{4.56}$$

The above Eqs. (4.54)–(4.56) present an exact relationship between the nodal or control-point displacements, velocities, and accelerations corresponding to the rotational motion. To relate the deflection degrees of freedom between time levels t_n and t_{n+1}, we make use of the standard Newmark formulas (see e.g., Hughes [133]):

$$\dot{\mathbf{U}}_d^{n+1} = \dot{\mathbf{U}}_d^n + \Delta t \left((1-\gamma)\ddot{\mathbf{U}}_d^n + \gamma \ddot{\mathbf{U}}_d^{n+1} \right), \tag{4.57}$$

$$\mathbf{U}_d^{n+1} = \mathbf{U}_d^n + \Delta t \dot{\mathbf{U}}_d^n + \frac{\Delta t^2}{2} \left((1-2\beta)\ddot{\mathbf{U}}_d^n + 2\beta \ddot{\mathbf{U}}_d^{n+1} \right), \tag{4.58}$$

where γ and β are the time integration parameters chosen to maintain second-order accuracy and unconditional stability of the method.

Combining exact rotations given by Eqs. (4.54)–(4.56) and time-discrete deflections given by Eqs. (4.57)–(4.58), we obtain the following modified Newmark formulas for the total discrete solution:

$$\begin{aligned}\dot{\mathbf{U}}^{n+1} &= \left\{\dot{\mathbf{R}}^{n+1} - \left[\dot{\mathbf{R}}^{n} + \Delta t\left((1-\gamma)\ddot{\mathbf{R}}^{n} + \gamma\ddot{\mathbf{R}}^{n+1}\right)\right]\right\}(\mathbf{X}-\mathbf{X}_0) \\ &\quad + \dot{\mathbf{U}}^{n} + \Delta t\left((1-\gamma)\ddot{\mathbf{U}}^{n} + \gamma\ddot{\mathbf{U}}^{n+1}\right), \qquad (4.59)\\ \mathbf{U}^{n+1} &= \left\{\mathbf{R}^{n+1} - \left[\mathbf{R}^{n} + \Delta t\dot{\mathbf{R}}^{n} + \frac{\Delta t^2}{2}\left((1-2\beta)\ddot{\mathbf{R}}^{n} + 2\beta\ddot{\mathbf{R}}^{n+1}\right)\right]\right\}(\mathbf{X}-\mathbf{X}_0) \\ &\quad + \mathbf{U}^{n} + \Delta t\dot{\mathbf{U}}^{n} + \frac{\Delta t^2}{2}\left((1-2\beta)\ddot{\mathbf{U}}^{n} + 2\beta\ddot{\mathbf{U}}^{n+1}\right). \qquad (4.60)\end{aligned}$$

We employ Eqs. (4.59)–(4.60), in conjunction with the generalized-α method of Chung and Hulbert [110], for the time discretization of the structure.

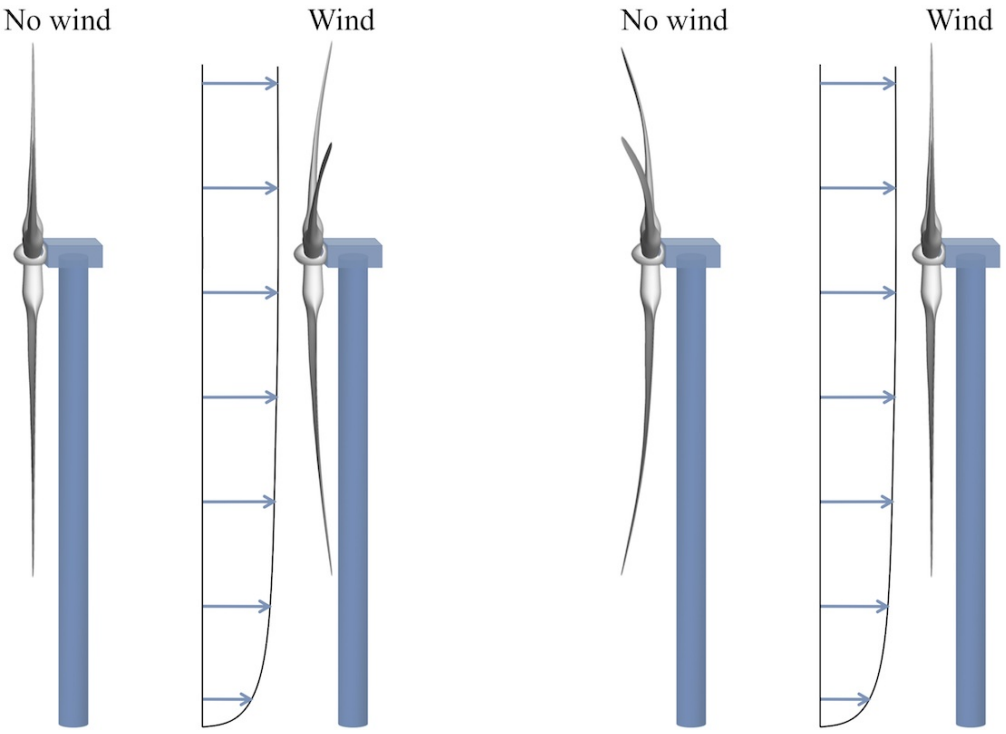
Remark 4.8. In the case of no rotation, for which $\mathbf{R}$ is an identity, Eqs. (4.59)–(4.60) reduce to the standard Newmark formulas. In the case of no deflection, pure rotation is likewise recovered.

4.4 Pre-Bending of the Wind Turbine Blades

The rotor blades of a wind turbine need to be designed such that they do not strike the tower as the rotor turns in strong winds. There are several ways of avoiding this. One is to tilt the rotor so that the blades are not at right angles to the nacelle. By the same token, it is possible to cone the rotor, thus ensuring tower clearance for the blades. However, both methods require adjustments to the nacelle design.

As an alternative, blade pre-bending may be employed. In this case, the blades are manufactured to flex toward the wind when they are mounted on the tower and are not subjected to the wind and inertial loading. Once the blades are exposed to the wind and the rotor starts turning, the blades are straightened to achieve their design shape. This situation is graphically illustrated in Figure 4.15.

Besides tower clearance, pre-bending the blades engenders additional benefits. For example, the blades are more flexible because the amount of allowable deflection is



(a) Without pre-bending (b) With pre-bending

Figure 4.15: Using pre-bent blades to ensure tower clearance and rotor operation in its design configuration.

greater. This makes it possible to use less material overall, resulting in lighter and more economical blades. During operation, the pre-bent blades straighten to their designed configuration, which is typically optimized for best possible aerodynamic performance.

Given the above advantages, it is important that one is able to determine the correct pre-bent shape given the blade structural and aerodynamic design, and the wind turbine operating conditions (i.e., wind and rotor speeds). In this section, we propose a method that makes use of standalone computational fluid and structural mechanics procedures to obtain a pre-bent shape of the wind turbine blades. The main idea consists of performing an aerodynamics simulation of a rigidly-rotating rotor to obtain the aerodynamic load acting on the blade. Given the aerodynamic and inertial loads acting in the design configuration, a stress-free pre-bent blade configuration is found using a simple iterative procedure that requires a sequence of structural mechanics simulations. Note that in the proposed approach the aerodynamic and structural computations are

decoupled, which avoids the challenges involved in solving the coupled FSI problem.

4.4.1 Problem Statement and the Pre-Bending Algorithm

We begin with the statement of virtual work for the structure from Eq. (4.36), where only the stress terms are left on the left-hand-side: find the displacement of the shell midsurface $\mathbf{d}^h \in \mathcal{S}_d^h$, such that $\forall \delta\mathbf{d}^h \in \mathcal{V}_d^h$,

$$\begin{aligned}&\int_{\Gamma_0^s} \delta\overline{\boldsymbol{\varepsilon}}^h \cdot \left(\mathbf{A}\,\overline{\boldsymbol{\varepsilon}}^h + \mathbf{B}\,\overline{\boldsymbol{\kappa}}^h\right) \mathrm{d}\Gamma + \int_{\Gamma_0^s} \delta\overline{\boldsymbol{\kappa}}^h \cdot \left(\mathbf{B}\,\overline{\boldsymbol{\varepsilon}}^h + \mathbf{D}\,\overline{\boldsymbol{\kappa}}^h\right) \mathrm{d}\Gamma + \int_{\Gamma_0^b} \delta\overline{\boldsymbol{\kappa}}^h \cdot \mathbf{D}_{\mathrm{b}}\,\overline{\boldsymbol{\kappa}}^h \,\mathrm{d}\Gamma \\ &= -\int_{\Gamma_t^s} \delta\mathbf{d}^h \cdot \rho h_{\mathrm{th}} \left(\left.\frac{\partial^2 \mathbf{d}^h}{\partial t^2}\right|_{\mathbf{X}} - \mathbf{f}^h \right) \mathrm{d}\Gamma + \int_{(\Gamma_t^s)_{\mathrm{h}}} \delta\mathbf{d}^h \cdot \mathbf{h}^h \,\mathrm{d}\Gamma. \end{aligned} \tag{4.61}$$

Although the above equation holds true, the problem setup is unusual in that the stress-free reference configuration Γ_0^s is unknown and the final configuration Γ_t^s is given. The formulation given by Eq. (4.61) is a form of the inverse deformation problem, whose general formulation and treatment was proposed by Shield [134], and further analyzed and studied computationally by Govindjee and Mihalic [135]. In these references, the focus was placed on developing the right kinematic and stress measures for the inverse deformation problem. Here we develop a simple algorithm for the solution of the inverse deformation equations with application to wind turbine blades.

We assume that the rotor spins around its axis at a constant angular speed and that the inflow wind conditions do not change. With this setup, the blade is subjected to a constant-in-time centripetal force density (per unit volume) given by

$$\rho \left.\frac{\partial^2 \mathbf{d}^h}{\partial t^2}\right|_{\mathbf{X}} = \rho\boldsymbol{\omega} \times (\boldsymbol{\omega} \times (\mathbf{x} - \mathbf{x}_0)), \tag{4.62}$$

where the coordinate system of the current configuration is assumed to rotate with the blade, $\boldsymbol{\omega}$ is the vector of angular velocities, and $\mathbf{x}_0$ is a fixed point. The centripetal force density per unit volume may be directly computed as

$$\rho\boldsymbol{\omega} \times (\boldsymbol{\omega} \times (\mathbf{x} - \mathbf{x}_0)) = \begin{bmatrix} -\rho x\dot{\theta}^2 \\ -\rho y\dot{\theta}^2 \\ 0 \end{bmatrix}, \tag{4.63}$$

where we assume that the coordinate system of the blade is chosen such that the y-axis is aligned with the blade axis, and the blade rotates around the z-axis with a constant angular speed $\dot{\theta}$.

The time-averaged aerodynamics traction vector $\mathbf{h}^h$ in Eq. (4.61) may be obtained from a separate aerodynamics computation of a rigidly-spinning rotor using the methods described in the previous chapter.

Here, we propose a two-stage iterative approach to solve the variational equations (4.61) for the shell midsurface displacement, which, in turn, gives the stress-free reference configuration:

Initialization: We initialize the unknown reference configuration to coincide with the current configuration, that is, $\Gamma_0^s = \Gamma_t^s$, which implies $\mathbf{d}^h = \mathbf{0}$.

Step 1: Given the reference configuration Γ_0^s, we solve the standard nonlinear structural problem: find the structural displacement $\mathbf{d}^h \in \mathcal{S}_d^h$ relative to Γ_0^s, such that $\forall \delta\mathbf{d}^h \in \mathcal{V}_d^h$,

$$\int_{\Gamma_0^s} \delta\overline{\boldsymbol{\varepsilon}}^h \cdot \left(\mathbf{A}\,\overline{\boldsymbol{\varepsilon}}^h + \mathbf{B}\,\overline{\boldsymbol{\kappa}}^h\right) \mathrm{d}\Gamma + \int_{\Gamma_0^s} \delta\overline{\boldsymbol{\kappa}}^h \cdot \left(\mathbf{B}\,\overline{\boldsymbol{\varepsilon}}^h + \mathbf{D}\,\overline{\boldsymbol{\kappa}}^h\right) \mathrm{d}\Gamma + \int_{\Gamma_0^b} \delta\overline{\boldsymbol{\kappa}}^h \cdot \mathbf{D}_\mathrm{b}\,\overline{\boldsymbol{\kappa}}^h \,\mathrm{d}\Gamma$$
$$= -\int_{\Gamma_t^s} \delta\mathbf{d}^h \cdot (\rho h_{\mathrm{th}} \boldsymbol{\omega} \times (\boldsymbol{\omega} \times (\mathbf{x} - \mathbf{x}_0))) \,\mathrm{d}\Gamma + \int_{(\Gamma_t^s)_\mathrm{h}} \delta\mathbf{d}^h \cdot \mathbf{h}^h \,\mathrm{d}\Gamma. \tag{4.64}$$

Standard Newton–Raphson iteration is employed in this work to compute the solution of the nonlinear structural problem given by Eq. (4.64).

Step 2: Given the displacement solution $\mathbf{d}^h$ from Step 1, we update the reference configuration as

$$\Gamma_0^s = \left\{\mathbf{X} \mid \mathbf{X} = \mathbf{x} - \mathbf{d}^h,\ \forall \mathbf{x} \in \Gamma_t^s\right\}, \tag{4.65}$$

and return to Step 1 using $\mathbf{d}^h$ as the initial data.

We repeat the above steps until convergence, that is, until $\mathbf{d}^h$ satisfies Eq. (4.64).

The above algorithm is based on the idea of computing negative increments of the displacement, or increments of the displacement away from the current configura-

tion, until the reference configuration is found. The mathematical justification for this approach may be found in the appendix of Bazilevs *et al.* [25]. In what follows, we will illustrate the good performance of the proposed algorithm on a full-scale wind turbine blade subjected to realistic wind and inertial loads.

4.4.2 Pre-Bending Results for the NREL 5 MW Wind Turbine Blade

The same blade design as in Section 4.2.3 and same wind conditions as in Section 3.3.2 Case II are taken for the pre-bending computations presented here. Figure 4.16 shows the tip displacement convergence of the iterative pre-bending algorithm. After a few (five to six) iterations of the two-step pre-bending algorithm the tip exhibits no further visible displacements, and the computation is stopped after a total of 15 iterations. Figure 4.17 shows the initial and the final stress-free blade shapes. As expected, the blade bends into the wind. The tip displacement is predicted to be 5.61 m.

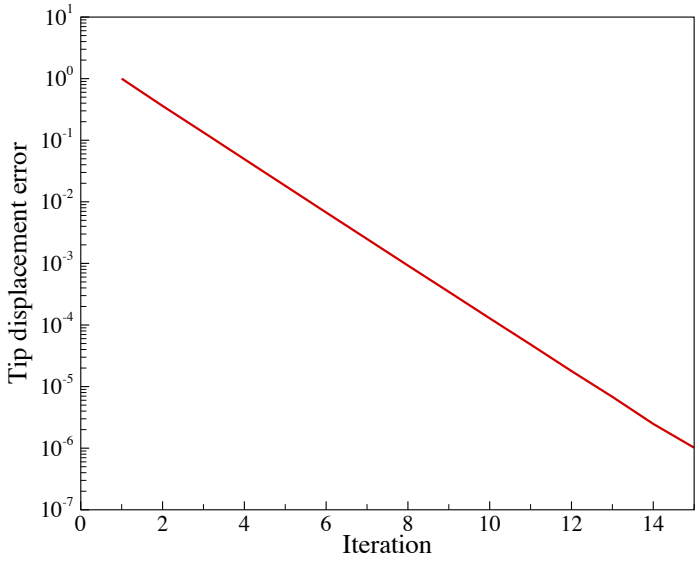


Figure 4.16: Blade tip displacement convergence as a function of the iteration number.

We then perform an elastostatic analysis of the pre-bent wind turbine blade subjected to the wind and inertial loads that were used for pre-bending computations. Figure 4.18(a) shows that the pre-bent blade is deformed and straightened into the design shape. The tip deflection is 5.61 m, which is consistent with the pre-bending tip dis-

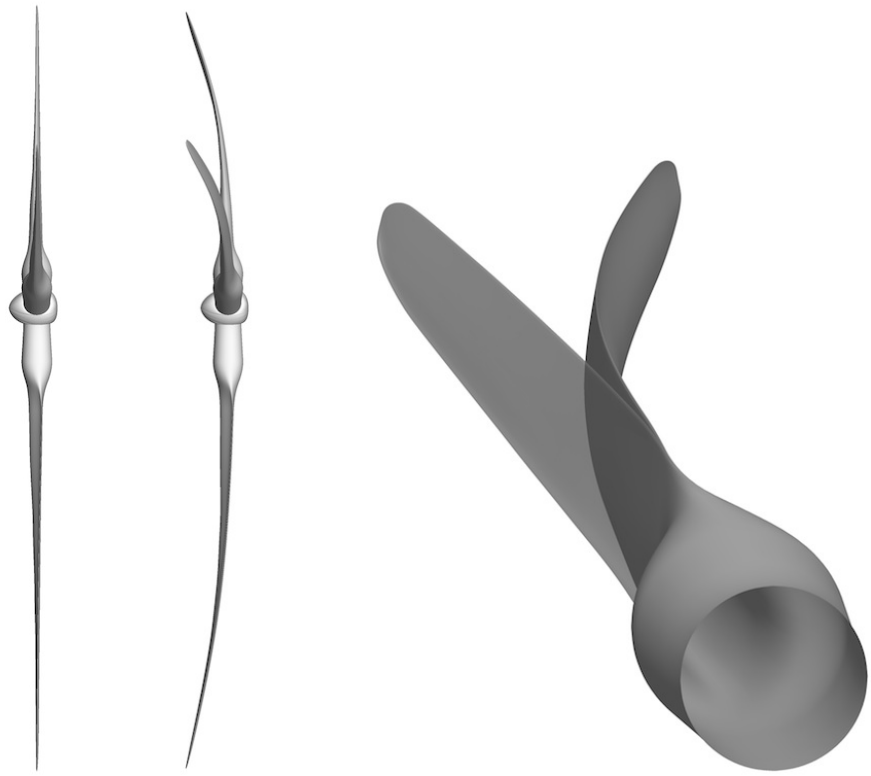
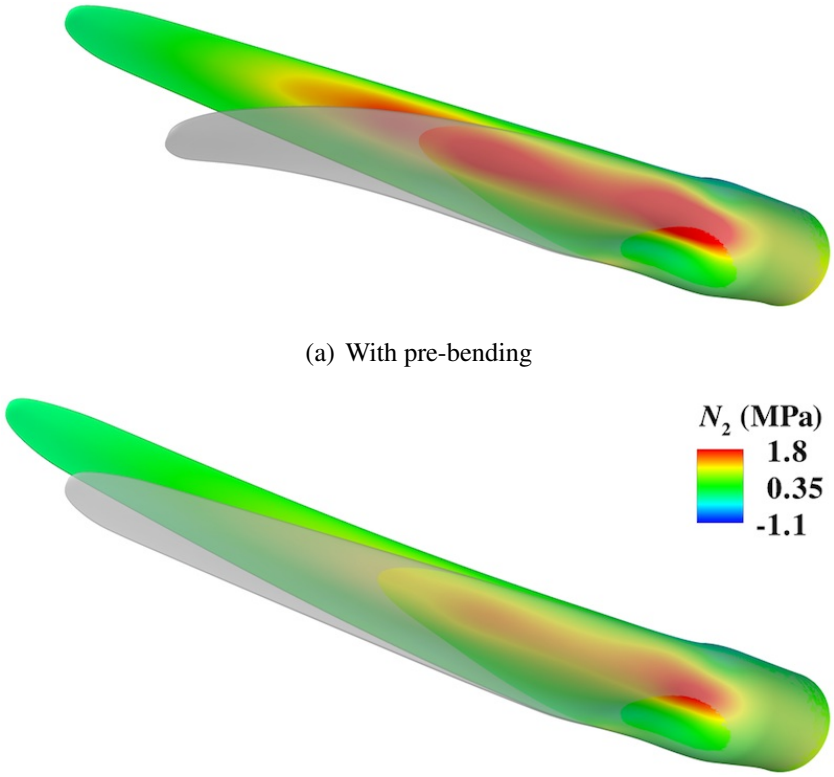


Figure 4.17: Pre-bending results for the NREL 5 MW wind turbine blade. Left: rotor design configuration. Middle: rotor pre-bent configuration. Right: rotor blade design and pre-bent configurations superposed.

placement. Another elastostatic analysis of the wind turbine blade without pre-bending (i.e., the design shape is taken as the reference configuration) is performed and the result is shown in Figure 4.18(b) for comparison. The design-shape blade is deformed under the wind and inertial loads. The tip deflection in this case is 3.62 m, which is much smaller than the previous case, in part shows the nonlinearity of the problem. The final deformed configuration in Figure 4.18 is colored by the force resultant component N_2, which is align in the flapwise direction of the blade.

We also examine the stress distribution in the composite plies of the blade skin. For each ply we compute the Cauchy stress tensor components with respect to the local Cartesian basis that is aligned with the material axes. The first basis vector points in the direction of the fiber and the second in the direction of the matrix, which is orthogonal to the fiber direction (see Eqs. (4.12) and (4.13)). In Table 4.5, we provide $\sigma_1^{t,u}$ and $\sigma_1^{c,u}$, the longitudinal tensile and compressive strength, respectively, $\sigma_2^{t,u}$ and $\sigma_2^{c,u}$, the transverse

tensile and compressive strength, respectively, and σ_{12}^{u}, the in-plane shear strength of the fiberglass/epoxy composite.



(a) With pre-bending

(b) Without pre-bending

Figure 4.18: The elastostatic analysis of the wind turbine blade subjected to the wind and inertial loads. The stress-free reference configuration is superposed with the deformed configuration, which is colored by the normal force in the flapwise direction.

Table 4.5: Strength of a unidirectional fiberglass/epoxy composite lamina taken from Daniel and Ishai [131].

$\sigma_1^{t,u}$ (MPa)	$\sigma_1^{c,u}$ (MPa)	$\sigma_2^{t,u}$ (MPa)	$\sigma_2^{c,u}$ (MPa)	σ_{12}^{u} (MPa)
1080	620	39	128	89

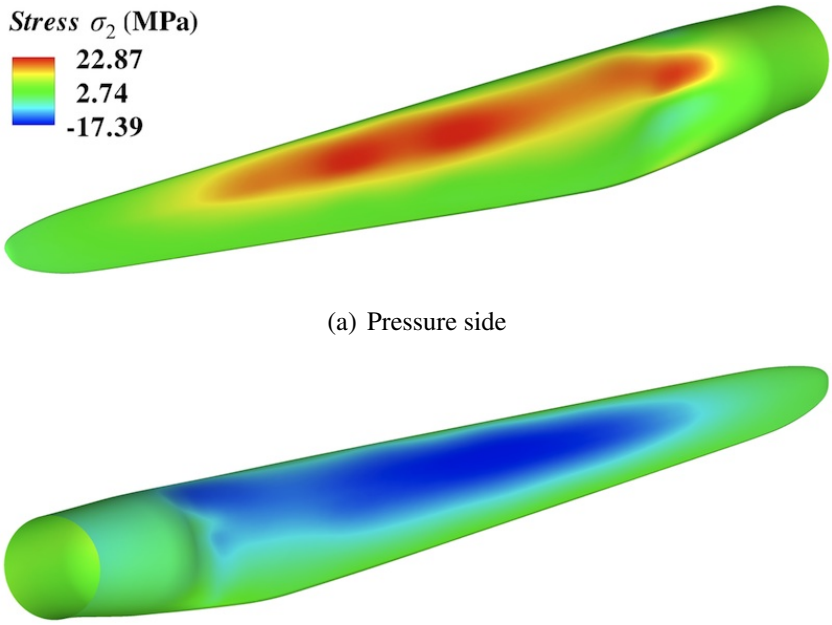
The maximum values of the tensile stresses (σ_1^t and σ_2^t), compressive stresses (σ_1^c and σ_2^c), and in-plane shear stress (σ_{12}) are computed for each ply and the results relative to the strength of the composite lamina are reported in Table 4.6. The highest ratio of the predicted Cauchy stress and the composite strength, i.e, $\sigma_2^t/\sigma_2^{t,u}$, occurs for the tensile stress in the direction of the matrix material. Although the ratio does not ex-

ceed 0.6, which means the predicted stress is below the composite failure strength, we feel this value is somewhat high. In the rest of the stress components the ratios are significantly lower. It is also evident from the table that the 0° plies are the most vulnerable in our design. The last column of Table 4.6 gives the Tsai-Wu failure criterion [136] for every ply. In this case, all the stress components are considered simultaneously, which gives a scalar-valued failure criterion that is widely used in practice.

Table 4.6: Maximum lamina stresses normalized by the failure strength of a unidirectional fiberglass/epoxy composite. The last column is the Tsai-Wu failure criterion [136].

Ply no.	$\sigma_1^t/\sigma_1^{t,u}$	$\sigma_1^c/\sigma_1^{c,u}$	$\sigma_2^t/\sigma_2^{t,u}$	$\sigma_2^c/\sigma_2^{c,u}$	$\lvert\sigma_{12}\rvert/\sigma_{12}^u$	Tsai–Wu
16 (45°)	0.060	0.089	0.449	0.136	0.166	0.355
15 (−45°)	0.064	0.104	0.393	0.107	0.156	0.295
14 (0°)	0.042	0.107	0.586	0.136	0.055	0.516
13 (90°)	0.098	0.131	0.248	0.095	0.051	0.174
12 (90°)	0.097	0.132	0.208	0.078	0.047	0.141
11 (0°)	0.022	0.056	0.573	0.138	0.043	0.504
10 (0°)	0.016	0.039	0.569	0.138	0.042	0.500
9 (0°)	0.009	0.022	0.564	0.139	0.042	0.495
8 (0°)	0.010	0.018	0.560	0.139	0.043	0.491
7 (0°)	0.020	0.020	0.556	0.140	0.046	0.487
6 (0°)	0.030	0.031	0.552	0.140	0.048	0.484
5 (90°)	0.093	0.138	0.297	0.044	0.051	0.201
4 (90°)	0.092	0.139	0.357	0.052	0.056	0.255
3 (0°)	0.059	0.065	0.540	0.142	0.060	0.472
2 (−45°)	0.072	0.071	0.458	0.076	0.127	0.349
1 (45°)	0.069	0.066	0.528	0.089	0.133	0.419

Figure 4.19 shows the distribution of σ_2 in the 0° ply number 14, which has the highest ratio of $\sigma_2^t/\sigma_2^{t,u}$. As expected, the pressure side of the blade is in tension and the suction side of the blade is in compression. However, the level of the tensile stress is not very far from the tensile failure strength, which suggests that stronger matrix material may be desirable for this blade design.



(a) Pressure side

(b) Suction side

Figure 4.19: The elastostatic analysis of the pre-bent wind turbine blade subjected to the wind and inertial loads. Normal stress distribution in the direction of the matrix material for the 0° ply number 14.

4.5 Acknowledgements

Chapter 4, in part, is a reprint of the material as it appears in: "ALE–VMS and ST–VMS methods for computer modeling of wind-turbine rotor aerodynamics and fluid–structure interaction," (with Y. Bazilevs, K. Takizawa and T.E. Tezduyar), *Mathematical Models and Methods in Applied Sciences*, 2012; "A computational procedure for prebending of wind turbine blades," (Y. Bazilevs, J. Kiendl and D.J. Benson), *International Journal for Numerical Methods in Engineering*, 2012; "3D simulation of wind turbine rotors at full scale. Part II: Fluid–structure interaction modeling with composite blades," (Y. Bazilevs, J. Kiendl, R. Wüchner and K.-U. Bletzinger), *International Journal for Numerical Methods in Fluids*, 2011; "The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches," (with J. Kiendl, Y. Bazilevs, K.-U. Bletzinger and R. Wüchner), *Computer Methods in Applied Mechanics and Engineering*, 2010. The dissertation author was the primary investigator and author of these papers.

Chapter 5

Fluid–Structure Interaction Modeling of Wind Turbines

In this chapter we focus on the coupling strategies, specific to IGA, for fluid–structure interaction (FSI) applications. We focus on and formulate computational approaches for IGA that make use of non-matching discretizations of the domain geometry at the interface between the fluid and structure subdomains. The proposed FSI methodology is based on the augmented Lagrangian concept. The Lagrange multiplier on the fluid–structure interface is formally eliminated to achieve a formulation purely in terms of the primal variables (i.e., velocity and pressure for the fluid and velocity for the structural problem). The fluid and structural mechanics trial and test functions are not assumed to be equal at the interface, and the FSI coupling is taken care of in the variational formulation. This presents a convenient point of departure for a discrete FSI formulation using non-matching fluid–structure interface discretizations. In the non-matching FSI formulation one needs a functional definition of the kinematic quantities (velocity, displacement, etc.) and tractions on the fluid and structure meshes, and the ability to "transfer" these quantities from one mesh to the other. The L^2-projection is chosen in this work for the transfer of kinematic and traction data between the subdomains.

The non-matching FSI coupling procedures allow greater flexibility in the computational analysis, and alleviate the difficulties of geometry modeling arising from the necessity to construct matching multi-physics interface discretizations. Furthermore, the

mesh resolution of the structural and fluid mechanics problems may be tailored to the analysis requirements of the individual subsystems, leading to improved computational efficiency. The increased complexity of structural geometry places heavy demands on the fluid volume mesh (or exterior mesh) generation around that structure. However, no "automatic mesh generation" software for IGA exists to this day. Volumetric geometry modeling and mesh generation for IGA is, at this point, still an active area of research (see Wang *et al.* [74, 75] for recent results of this challenging problem). As a result, in order to take advantage of the superior accuracy of IGA for structural mechanics applications (see, e.g., Cottrell *et al.* [31] and Auricchio *et al.* [38]), and to leverage the existing advanced volumetric mesh generation tools for standard finite elements, we propose to couple FEM for fluid mechanics and IGA for structural dynamics. Although IGA was shown to produce results that are of better per-degree-of-freedom quality than standard FEM for fluid mechanics and, especially, for turbulent flows [44], good quality aerodynamics may still be achieved with standard low-order FEM, with a manageable number of degrees of freedom (see Sections 3.4, 3.5 and 3.6).

5.1 FSI Formulation Based on the Augmented Lagrangian Approach

Let $(\Omega_t)_1 \in \mathbb{R}^d$, $d = 2, 3$, represent the time-dependent domain of the fluid mechanics (or aerodynamics) problem, let $(\Omega_t)_2 \in \mathbb{R}^d$ be the time-dependent domain of the structural mechanics problem, and let $(\Gamma_t)_\mathrm{I} \in \mathbb{R}^d$ denote the interface between the fluid and structural domains. All the domains are taken at current time t at this point in the developments. Let $\mathbf{u}_1$ and p denote the fluid velocity and pressure, respectively. We assume that the fluid mechanics is governed by the Navier–Stokes equations of incompressible, isothermal and Newtonian flow. Let $\mathbf{u}_2$ denote the velocity of the structure.

We define an augmented Lagrangian function $\mathbf{N}(\{\mathbf{u}_1, p\}, \mathbf{u}_2, \boldsymbol{\lambda})$ as

$$\begin{aligned}\mathbf{N}(\{\mathbf{u}_1, p\}, \mathbf{u}_2, \boldsymbol{\lambda}) = \mathbf{N}_1(\{\mathbf{u}_1, p\}) + \mathbf{N}_2(\mathbf{u}_2) \\ + \int_{(\Gamma_t)_\mathrm{I}} \boldsymbol{\lambda} \cdot (\mathbf{u}_1 - \mathbf{u}_2)\, \mathrm{d}\Gamma \\ + \frac{1}{2} \int_{(\Gamma_t)_\mathrm{I}} \beta(\mathbf{u}_1 - \mathbf{u}_2) \cdot (\mathbf{u}_1 - \mathbf{u}_2)\, \mathrm{d}\Gamma, \end{aligned} \tag{5.1}$$

where $\mathbf{N}_1$ and $\mathbf{N}_2$ are the Lagrangian functionals whose stationary points generate the variational equations of fluid and structural mechanics, $\boldsymbol{\lambda}$ is a Lagrange multiplier for the kinematic interface condition $\mathbf{u}_1 = \mathbf{u}_2$, and β is a penalty parameter, which we leave unspecified at the moment. The augmented Lagrangian approach may be interpreted as a combination of the Lagrange multiplier and penalty methods. It is a popular approach in optimization, as well as in applications of nonlinear structural mechanics that involve some form of constraints (e.g., contact). Here, we adopt it as a point of departure for generating a family of FSI formulations.

Taking the variational derivatives of $\mathbf{N}$ with respect to the fluid, structural and Lagrange multiplier unknowns, and setting the result to zero yields the following set of variational equations: find the fluid velocity and pressure, $\mathbf{u}_1 \in \mathcal{S}_u$ and $p \in \mathcal{S}_p$, the structural velocity, $\mathbf{u}_2 \in \mathcal{S}_d$, and the Lagrange multiplier $\boldsymbol{\lambda} \in \mathcal{S}_l$, such that for all weighting functions $\mathbf{w}_1 \in \mathcal{V}_u$, $q \in \mathcal{V}_p$, $\mathbf{w}_2 \in \mathcal{V}_d$, and $\delta\boldsymbol{\lambda} \in \mathcal{V}_l$,

$$B_1(\{\mathbf{w}_1, q\}, \{\mathbf{u}_1, p\}) - F_1(\{\mathbf{w}_1, q\}) + \int_{(\Gamma_t)_\mathrm{I}} \mathbf{w}_1 \cdot \boldsymbol{\lambda}\, \mathrm{d}\Gamma + \int_{(\Gamma_t)_\mathrm{I}} \mathbf{w}_1 \cdot \beta(\mathbf{u}_1 - \mathbf{u}_2)\, \mathrm{d}\Gamma = 0, \tag{5.2}$$

$$B_2(\mathbf{w}_2, \mathbf{u}_2) - F_2(\mathbf{w}_2) - \int_{(\Gamma_t)_\mathrm{I}} \mathbf{w}_2 \cdot \boldsymbol{\lambda}\, \mathrm{d}\Gamma - \int_{(\Gamma_t)_\mathrm{I}} \mathbf{w}_2 \cdot \beta(\mathbf{u}_1 - \mathbf{u}_2)\, \mathrm{d}\Gamma = 0, \tag{5.3}$$

$$\int_{(\Gamma_t)_\mathrm{I}} \delta\boldsymbol{\lambda} \cdot (\mathbf{u}_1 - \mathbf{u}_2)\, \mathrm{d}\Gamma = 0, \tag{5.4}$$

where $\mathcal{S}_u$, $\mathcal{S}_p$, $\mathcal{S}_d$, and $\mathcal{S}_l$ are the function spaces for the fluid velocity and pressure, structural velocity, and the Lagrange multiplier solutions, respectively, and $\mathcal{V}_u$, $\mathcal{V}_p$, $\mathcal{V}_d$, and $\mathcal{V}_l$ are the corresponding weighting function spaces. B_1, B_2, F_1, and F_2 are the semilinear forms and linear functional corresponding to the fluid and structural mechan-

ics problems, respectively, and are given by

$$B_1(\{\mathbf{w}, q\}, \{\mathbf{u}, p\}) = \int_{(\Omega_t)_1} \mathbf{w} \cdot \rho_1 \left(\left.\frac{\partial \mathbf{u}}{\partial t}\right|_x + \mathbf{u} \cdot \boldsymbol{\nabla}_x \mathbf{u} \right) \mathrm{d}\Omega + \int_{(\Omega_t)_1} \boldsymbol{\varepsilon}(\mathbf{w}) : \boldsymbol{\sigma}_1(\mathbf{u}, p) \ \mathrm{d}\Omega + \int_{(\Omega_t)_1} q \boldsymbol{\nabla}_x \cdot \mathbf{u} \ \mathrm{d}\Omega, \tag{5.5}$$

$$F_1(\{\mathbf{w}, q\}) = \int_{(\Omega_t)_1} \mathbf{w} \cdot \rho_1 \mathbf{f} \ \mathrm{d}\Omega + \int_{(\Gamma_t)_{1\mathrm{h}}} \mathbf{w} \cdot \mathbf{h} \ \mathrm{d}\Gamma, \tag{5.6}$$

$$B_2(\mathbf{w}, \mathbf{u}) = \int_{(\Omega_t)_2} \mathbf{w} \cdot \rho_2 \left.\frac{\partial \mathbf{u}}{\partial t}\right|_X \mathrm{d}\Omega + \int_{(\Omega_t)_2} \boldsymbol{\varepsilon}(\mathbf{w}) : \boldsymbol{\sigma}_2(\mathbf{u}) \ \mathrm{d}\Omega, \tag{5.7}$$

$$F_2(\mathbf{w}) = \int_{(\Omega_t)_2} \mathbf{w} \cdot \rho_2 \mathbf{f} \ \mathrm{d}\Omega + \int_{(\Gamma_t)_{2\mathrm{h}}} \mathbf{w} \cdot \mathbf{h} \ \mathrm{d}\Gamma, \tag{5.8}$$

where ρ_1 and ρ_2 are the fluid and structural densities, $\mathbf{f}$ and $\mathbf{h}$ are the applied body force and surface traction, $\left|_x\right.$ denotes the fact that the time derivative in the fluid mechanics equations is taken with respect to the fixed coordinates of the spatial configuration, while $\left|_X\right.$ in the structural problem denotes the fact that the time derivative is taken with respect to the fixed coordinates of the material configuration. The fluid Cauchy stress $\boldsymbol{\sigma}_1$ is given by

$$\boldsymbol{\sigma}_1(\mathbf{u}, p) = -p\mathbf{I} + 2\mu\boldsymbol{\varepsilon}(\mathbf{u}), \tag{5.9}$$

where $\mathbf{I}$ is the identity tensor, μ is the fluid dynamic viscosity, and $\boldsymbol{\varepsilon}(\mathbf{u})$ is the strain-rate tensor given by

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2}\left(\boldsymbol{\nabla}_x \mathbf{u} + \boldsymbol{\nabla}_x \mathbf{u}^T\right). \tag{5.10}$$

At this point we do not detail the structural Cauchy stress $\boldsymbol{\sigma}_2$ as we wish to keep the structural mechanics formulation as flexible as possible.

The variational formulations given by Eqs. (5.2) and (5.3) give the following Euler–Lagrange conditions on the fluid–structural interface $(\Gamma_t)_{\mathrm{I}}$:

$$\boldsymbol{\lambda} = -\boldsymbol{\sigma}_1 \mathbf{n}_1 - \beta(\mathbf{u}_1 - \mathbf{u}_2), \tag{5.11}$$

$$\boldsymbol{\lambda} = \boldsymbol{\sigma}_2 \mathbf{n}_2 - \beta(\mathbf{u}_1 - \mathbf{u}_2), \tag{5.12}$$

where $\mathbf{n}_1$ and $\mathbf{n}_2$ are the unit outward normal vectors to the fluid and structural domains, respectively. Note that, at the fluid–structure interface, $\mathbf{n}_1 = -\mathbf{n}_2$. Subtracting Eq. (5.11) from Eq. (5.12) results in

$$\boldsymbol{\sigma}_1\mathbf{n}_1 + \boldsymbol{\sigma}_2\mathbf{n}_2 = \mathbf{0}, \tag{5.13}$$

which the fluid and structure tractions are in equilibrium at their interface. The Lagrange multiplier Eq. (5.4) implies the kinematic compatibility condition at the fluid–structure interface, namely

$$\mathbf{u}_1 = \mathbf{u}_2. \tag{5.14}$$

Remark 5.1. Note that the traction compatibility condition given by Eq. (5.13) was derived without using the kinematic compatibility condition $\mathbf{u}_1 = \mathbf{u}_2$. This is a consequence of the augmented Lagrangian formulation.

Introducing Eq. (5.14) into Eqs. (5.11) and (5.12) yields

$$\boldsymbol{\lambda} = -\boldsymbol{\sigma}_1\mathbf{n}_1 = \boldsymbol{\sigma}_2\mathbf{n}_2, \tag{5.15}$$

which gives an interpretation of the Lagrange multiplier as the interface traction vector that may be computed from the fluid or structural subdomains. As a result, $\boldsymbol{\lambda}$ may be expressed as convex combination of the fluid and structure traction vectors as

$$\boldsymbol{\lambda} = -\alpha\boldsymbol{\sigma}_1\mathbf{n}_1 + (1-\alpha)\boldsymbol{\sigma}_2\mathbf{n}_2, \tag{5.16}$$

where α is a real number.

We now formally eliminate the Lagrange multiplier from the formulation of the FSI problem. The variation of $\boldsymbol{\lambda}$ with respect to the fluid and structural mechanics unknowns may be computed directly from Eq. (5.16), which gives

$$\delta\boldsymbol{\lambda} = -\alpha\delta_{\{\mathbf{u}_1,p\}}\boldsymbol{\sigma}_1\mathbf{n}_1(\{\mathbf{w}_1,q\}) + (1-\alpha)\delta_{\mathbf{u}_2}\boldsymbol{\sigma}_2\mathbf{n}_2(\mathbf{w}_2). \tag{5.17}$$

Introducing Eqs. (5.16) and (5.17) into Eqs. (5.2)–(5.4), scaling Eq. (5.4) by the parameter γ, and combining Eqs. (5.2)–(5.4) into a single variational form gives: find $\mathbf{u}_1 \in \mathcal{S}_u$,

$p \in \mathcal{S}_p$, and $\mathbf{u}_2 \in \mathcal{S}_d$, such that for all $\mathbf{w}_1 \in \mathcal{V}_u$, $q \in \mathcal{V}_p$, and $\mathbf{w}_2 \in \mathcal{V}_d$

$$
\begin{aligned}
&B_1(\{\mathbf{w}_1, q\}, \{\mathbf{u}_1, p\}) - F_1(\{\mathbf{w}_1, q\}) + B_2(\mathbf{w}_2, \mathbf{u}_2) - F_2(\mathbf{w}_2) \\
&+ \int_{(\Gamma_t)_\mathrm{I}} (\mathbf{w}_1 - \mathbf{w}_2) \cdot (-\alpha \boldsymbol{\sigma}_1 \mathbf{n}_1 + (1-\alpha)\boldsymbol{\sigma}_2 \mathbf{n}_2) \ \mathrm{d}\Gamma \\
&+ \gamma \int_{(\Gamma_t)_\mathrm{I}} \left(-\alpha \delta_{\{\mathbf{u}_1, p\}} \boldsymbol{\sigma}_1 \mathbf{n}_1(\{\mathbf{w}_1, q\}) + (1-\alpha)\delta_{\mathbf{u}_2} \boldsymbol{\sigma}_2 \mathbf{n}_2(\mathbf{w}_2) \right) \cdot (\mathbf{u}_1 - \mathbf{u}_2) \, \mathrm{d}\Gamma \\
&+ \int_{(\Gamma_t)_\mathrm{I}} (\mathbf{w}_1 - \mathbf{w}_2) \cdot \beta (\mathbf{u}_1 - \mathbf{u}_2) \, \mathrm{d}\Gamma = 0.
\end{aligned} \tag{5.18}
$$

The variational statement given by Eq. (5.18) defines a family of FSI formulations parameterized by α, β and γ. However, we note that the different choices of these parameters do not change the underlying FSI problem.

In this work we choose $\alpha = 1$ and $\gamma = 1$. The former choice is motivated by the fact that here we wish to avoid taking a variation of the structural Cauchy stress, while the latter choice leads to adjoint consistency of our discrete formulation (see Arnold *et al.* [68] for details). We postpone the choice of β until we state the discrete FSI problem.

With this definition of the parameters, we obtain the following coupled formulation: find $\mathbf{u}_1 \in \mathcal{S}_u$, $p \in \mathcal{S}_p$, and $\mathbf{u}_2 \in \mathcal{S}_d$, such that for all $\mathbf{w}_1 \in \mathcal{V}_u$, $q \in \mathcal{V}_p$, and $\mathbf{w}_2 \in \mathcal{V}_d$,

$$
\begin{aligned}
&B_1(\{\mathbf{w}_1, q\}, \{\mathbf{u}_1, p\}) - F_1(\{\mathbf{w}_1, q\}) + B_2(\mathbf{w}_2, \mathbf{u}_2) - F_2(\mathbf{w}_2) \\
&- \int_{(\Gamma_t)_\mathrm{I}} (\mathbf{w}_1 - \mathbf{w}_2) \cdot \boldsymbol{\sigma}_1 \mathbf{n}_1 \, \mathrm{d}\Gamma \\
&- \int_{(\Gamma_t)_\mathrm{I}} \left(\delta_{\{\mathbf{u}_1, p\}} \boldsymbol{\sigma}_1 \mathbf{n}_1(\{\mathbf{w}_1, q\}) \right) \cdot (\mathbf{u}_1 - \mathbf{u}_2) \, \mathrm{d}\Gamma \\
&+ \int_{(\Gamma_t)_\mathrm{I}} (\mathbf{w}_1 - \mathbf{w}_2) \cdot \beta (\mathbf{u}_1 - \mathbf{u}_2) \, \mathrm{d}\Gamma = 0.
\end{aligned} \tag{5.19}
$$

The coupled formulation given by Eq. (5.19) leads to the following interpretation of the individual fluid and structural subproblems. The fluid subproblem may be obtained by setting $\mathbf{w}_2 = \mathbf{0}$. This yields: find $\mathbf{u}_1 \in \mathcal{S}_u$ and $p \in \mathcal{S}_p$ such that for all $\mathbf{w}_1 \in \mathcal{V}_u$ and

$q \in \mathcal{V}_p$,

$$
\begin{aligned}
&B_1(\{\mathbf{w}_1, q\}, \{\mathbf{u}_1, p\}) - F_1(\{\mathbf{w}_1, q\}) \\
&- \int_{(\Gamma_t)_1} \mathbf{w}_1 \cdot \boldsymbol{\sigma}_1 \mathbf{n}_1 \, \mathrm{d}\Gamma \\
&- \int_{(\Gamma_t)_1} \left(\delta_{\{\mathbf{u}_1, p\}} \boldsymbol{\sigma}_1 \mathbf{n}_1(\{\mathbf{w}_1, q\}) \right) \cdot (\mathbf{u}_1 - \mathbf{u}_2) \, \mathrm{d}\Gamma \\
&+ \int_{(\Gamma_t)_1} \mathbf{w}_1 \cdot \beta(\mathbf{u}_1 - \mathbf{u}_2) \, \mathrm{d}\Gamma = 0,
\end{aligned} \tag{5.20}
$$

where one immediately recognizes a continuous counterpart of the fluid mechanics formulation with weakly enforced essential boundary conditions on the fluid velocity (see Section 3.1.3). These essential boundary conditions come from the structural velocity at the fluid–structure interface.

Setting $\{\mathbf{w}_1, q\} = \{\mathbf{0}, 0\}$ in Eq. (5.19) gives the following structural subproblem: find $\mathbf{u}_2 \in \mathcal{S}_d$, such that for all $\mathbf{w}_2 \in \mathcal{V}_d$,

$$
B_2(\mathbf{w}_2, \mathbf{u}_2) - F_2(\mathbf{w}_2) + \int_{(\Gamma_t)_1} \mathbf{w}_2 \cdot \left(\boldsymbol{\sigma}_1 \mathbf{n}_1 + \beta \left(\mathbf{u}_2 - \mathbf{u}_1 \right) \right) \, \mathrm{d}\Gamma = 0, \tag{5.21}
$$

which states that at the fluid–structure interface the structural problem is driven by the fluid traction vector $\mathbf{t}_1$ given by

$$
\mathbf{t}_1 = -\boldsymbol{\sigma}_1 \mathbf{n}_1 - \beta \left(\mathbf{u}_2 - \mathbf{u}_1 \right). \tag{5.22}
$$

The traction vector contains the usual term $-\boldsymbol{\sigma}_1 \mathbf{n}_1$, and it is also augmented by the term that is proportional to the difference between the fluid and structural velocities at their interface.

Remark 5.2. Choosing $\alpha = 1$ in Eq. (5.18) leads to subproblems defined by Eqs. (5.20) and (5.21). With this choice, the fluid mechanics problem is responsible for satisfying the kinematic compatibility condition, while the structural subproblem is responsible for satisfying the traction compatibility condition. This interpretation is consistent with how one typically thinks about FSI problems. Other choices of α lead to different interpretations of the FSI problem. For example, setting $\alpha = 0$ reverses the roles of the fluid and structural subproblems in that now the structure is responsible for satisfying the kinematic compatibility condition and the fluid enforces the compatibility of tractions. Intermediate values of α lead to other interpretations of the roles played by the individual subproblems.

Remark 5.3. The coupled formulation given by Eq. (5.18), derived using the augmented Lagrangian approach as a starting point, may also be interpreted as Nitsche's method for FSI (see, e.g., Nitsche [67] and Hansbo and Hermansson [137]) or as a continuous version of the Discontinuous Galerkin method for FSI applied at the fluid–structure interface.

Remark 5.4. The sliding interface formulation of Eq. (3.31), presented in Section 3.5.1 for handling objects in relative motion, can also be inspired from Eq. (5.18) by choosing B_1 and F_1 to be the bilinear form and linear functional, respectively, corresponding to the fluid mechanics problem on the stationary subdomain, B_2 and F_2 to be the bilinear form and linear functional, respectively, corresponding to the fluid mechanics problem on the rotating subdomain, and $\alpha = \frac{1}{2}$. Subscripts 1 and 2 denote the quantities pertaining to the fluid mechanics problem on the stationary and rotating subdomains, respectively.

Remark 5.5. In the above developments we assumed that the trial and test function spaces of the fluid and structural subproblems are independent of each other. This approach provides one with the framework that is capable of handling non-matching fluid and structural interface discretizations. If we explicitly assume that the fluid and structural velocities and the corresponding test functions are continuous at their interface, the FSI formulation given by Eq. (5.18) reduces to: find $\mathbf{u}_1 \in \mathcal{S}_u$, $p \in \mathcal{S}_p$, and $\mathbf{u}_2 \in \mathcal{S}_d$, such that for all $\mathbf{w}_1 \in \mathcal{V}_u$, $q \in \mathcal{V}_p$, and $\mathbf{w}_2 \in \mathcal{V}_d$,

$$B_1(\{\mathbf{w}_1, q\}, \{\mathbf{u}_1, p\}) - F_1(\{\mathbf{w}_1, q\}) + B_2(\mathbf{w}_2, \mathbf{u}_2) - F_2(\mathbf{w}_2) = 0. \tag{5.23}$$

This form of the FSI problem is suitable for matching fluid–structure interface meshes. Although somewhat limiting, matching interface discretizations were employed by many researchers [56, 57, 59, 138–145] to solve several problems of contemporary interest in computational mechanics and engineering.

5.2 FSI Formulation for Non-Matching Discretizations Suitable for IGA and FEM

In this section we develop the coupled FSI formulation at the discrete level. We assume non-matching fluid–structure interface discretization and take the continuous

fluid and structural formulations from the previous section as a point of departure. The details of the fluid and structural mechanics problems were presented in Chapters 3 and 4, respectively. Here, we briefly summarize the discrete variational formulations of the fluid and structural mechanics problems, describe the fluid–structural coupling and present our kinematic and traction data transfer methods between the non-matching fluid and structural interface discretizations.

5.2.1 The ALE–VMS Formulation of Fluid Mechanics with Weak Boundary Conditions

We begin with the fluid mechanics problem given by Eq. (5.20), and state the corresponding semi-discrete variational formulation: find $\mathbf{u}_1^h \in \mathcal{S}_u^h$ and $p^h \in \mathcal{S}_p^h$ such that for all $\mathbf{w}_1^h \in \mathcal{V}_u^h$ and $q^h \in \mathcal{V}_p^h$,

$$
\begin{aligned}
&B_1^{\mathrm{VMS}}(\{\mathbf{w}_1^h, q^h\}, \{\mathbf{u}_1^h, p^h\}; \hat{\mathbf{u}}^h) - F_1^{\mathrm{VMS}}(\{\mathbf{w}_1^h, q^h\}) \\
&- \sum_{b=1}^{N_{\mathrm{eb}}} \int_{\Gamma^b \bigcap (\Gamma_t)_{\mathrm{I}}} \mathbf{w}_1^h \cdot \boldsymbol{\sigma}_1 \left(\mathbf{u}_1^h, p^h\right) \mathbf{n}_1 \, \mathrm{d}\Gamma \\
&- \sum_{b=1}^{N_{\mathrm{eb}}} \int_{\Gamma^b \bigcap (\Gamma_t)_{\mathrm{I}}} \left(2\mu\boldsymbol{\varepsilon}\left(\mathbf{w}_1^h\right)\mathbf{n}_1 + q^h \mathbf{n}_1\right) \cdot \left(\mathbf{u}_1^h - \hat{\mathbf{u}}^h\right) \, \mathrm{d}\Gamma \\
&- \sum_{b=1}^{N_{\mathrm{eb}}} \int_{\Gamma^b \bigcap (\Gamma_t)_{\mathrm{I}}^-} \mathbf{w}_1^h \cdot \rho_1 \left(\left(\mathbf{u}_1^h - \hat{\mathbf{u}}^h\right) \cdot \mathbf{n}_1\right) \left(\mathbf{u}_1^h - \hat{\mathbf{u}}^h\right) \, \mathrm{d}\Gamma \\
&+ \sum_{b=1}^{N_{\mathrm{eb}}} \int_{\Gamma^b \bigcap (\Gamma_t)_{\mathrm{I}}} \tau_{\mathrm{B}}^{\mathrm{TAN}} \left(\mathbf{w}_1^h - \left(\mathbf{w}_1^h \cdot \mathbf{n}_1\right)\mathbf{n}_1\right) \cdot \left(\left(\mathbf{u}_1^h - \hat{\mathbf{u}}^h\right) - \left(\left(\mathbf{u}_1^h - \hat{\mathbf{u}}^h\right) \cdot \mathbf{n}_1\right)\mathbf{n}_1\right) \, \mathrm{d}\Gamma \\
&+ \sum_{b=1}^{N_{\mathrm{eb}}} \int_{\Gamma^b \bigcap (\Gamma_t)_{\mathrm{I}}} \tau_{\mathrm{B}}^{\mathrm{NOR}} \left(\mathbf{w}_1^h \cdot \mathbf{n}_1\right)\left(\left(\mathbf{u}_1^h - \hat{\mathbf{u}}^h\right) \cdot \mathbf{n}_1\right) \, \mathrm{d}\Gamma = 0,
\end{aligned}
\tag{5.24}
$$

where B_1^{VMS} and F_1^{VMS} are the discrete counterparts of B_1 and F_1, respectively, given by

$$
\begin{aligned}
B_1^{\mathrm{VMS}}(\{\mathbf{w}^h, q^h\}, \{\mathbf{u}^h, p^h\}; \hat{\mathbf{u}}^h) &= \int_{(\Omega_t)_1} \mathbf{w}^h \cdot \rho_1 \left(\left.\frac{\partial \mathbf{u}^h}{\partial t}\right|_{\hat{x}} + \left(\mathbf{u}^h - \hat{\mathbf{u}}^h\right) \cdot \boldsymbol{\nabla}_x \mathbf{u}^h \right) \mathrm{d}\Omega \\
&+ \int_{(\Omega_t)_1} \boldsymbol{\varepsilon}\left(\mathbf{w}^h\right) : \boldsymbol{\sigma}\left(\mathbf{u}^h, p^h\right) \mathrm{d}\Omega + \int_{(\Omega_t)_1} q^h \boldsymbol{\nabla}_x \cdot \mathbf{u}^h \, \mathrm{d}\Omega \\
&+ \sum_{e=1}^{N_{\mathrm{el}}} \int_{(\Omega_t^e)_1} \tau_{\mathrm{M}} \left(\left(\mathbf{u}^h - \hat{\mathbf{u}}^h\right) \cdot \boldsymbol{\nabla}_x \mathbf{w}^h + \frac{\boldsymbol{\nabla}_x q^h}{\rho_1} \right) \cdot \mathbf{r}_{\mathrm{M}}\left(\mathbf{u}^h, p^h\right) \mathrm{d}\Omega \\
&+ \sum_{e=1}^{N_{\mathrm{el}}} \int_{(\Omega_t^e)_1} \rho_1 \tau_{\mathrm{C}} \boldsymbol{\nabla}_x \cdot \mathbf{w}^h r_{\mathrm{C}}(\mathbf{u}^h, p^h) \, \mathrm{d}\Omega \\
&- \sum_{e=1}^{N_{\mathrm{el}}} \int_{(\Omega_t^e)_1} \tau_{\mathrm{M}} \mathbf{w}^h \cdot \left(\mathbf{r}_{\mathrm{M}}\left(\mathbf{u}^h, p^h\right) \cdot \boldsymbol{\nabla}_x \mathbf{u}^h \right) \mathrm{d}\Omega \\
&- \sum_{e=1}^{N_{\mathrm{el}}} \int_{(\Omega_t^e)_1} \frac{\boldsymbol{\nabla}_x \mathbf{w}^h}{\rho_1} \; : \; \left(\tau_{\mathrm{M}} \mathbf{r}_{\mathrm{M}}\left(\mathbf{u}^h, p^h\right)\right) \otimes \left(\tau_{\mathrm{M}} \mathbf{r}_{\mathrm{M}}\left(\mathbf{u}^h, p^h\right)\right) \mathrm{d}\Omega,
\end{aligned}
\tag{5.25}
$$

and

$$
F_1^{\mathrm{VMS}}(\{\mathbf{w}^h, q^h\}) = F_1(\{\mathbf{w}^h, q^h\}). \tag{5.26}
$$

The above formulation corresponds to an ALE–VMS method with weakly enforced essential boundary conditions. The discrete velocities and pressures and the corresponding test functions are now superscripted with h to denote their dependence on the mesh size. The ALE formulation is employed to handle the moving spatial domain of the fluid mechanics problem. Note that $\left.\right|_{\hat{x}}$ in Eq. (5.25) denotes the fact that the time derivative is taken with respect to a fixed referential domain spatial coordinates $\hat{\mathbf{x}}$, and $\hat{\mathbf{u}}^h$ is the mesh velocity. The time-dependent fluid domain $(\Omega_t)_1$ is divided into N_{el} individual element subdomains denoted by $(\Omega_t^e)_1$. The discrete trial function spaces $\mathcal{S}_u^h$ for the velocity and $\mathcal{S}_p^h$ for the pressure, as well as the corresponding test function spaces $\mathcal{V}_u^h$ and $\mathcal{V}_p^h$ are assumed to be of equal order, and, in this work, are comprised of NURBS or FEM functions. In Eq. (5.25), $\mathbf{r}_{\mathrm{M}}$ and r_{C} are the residuals of the momentum and continuity equations, respectively, and τ_{M} and τ_{C} are the stabilization parameters.

The fluid–structure interface $(\Gamma_t)_{\mathrm{I}}$ is decomposed into N_{eb} fluid domain surface elements, and $(\Gamma_t)_{\mathrm{I}}^-$ is defined as the "inflow" part of $(\Gamma_t)_{\mathrm{I}}$ as

$$
(\Gamma_t)_{\mathrm{I}}^- = \left\{ \mathbf{x} \;\middle|\; \left(\mathbf{u}_1^h - \hat{\mathbf{u}}^h\right) \cdot \mathbf{n}_1 < 0, \; \forall \mathbf{x} \subset (\Gamma_t)_{\mathrm{I}} \right\}. \tag{5.27}
$$

The penalty parameter τ_{B} is, in general, assumed to be different for the slip and penetration components of the flow velocity. However, for the computations presented in this work we use the same definition for both, namely,

$$\tau_{\mathrm{B}}^{\mathrm{TAN}} = \tau_{\mathrm{B}}^{\mathrm{NOR}} = \frac{C_I^B \mu_1}{h_n}, \tag{5.28}$$

where h_n is the wall-normal element size, and C_I^B is a sufficiently large positive constant computed from an appropriate element-level inverse estimate.

The mesh velocity $\hat{\mathbf{u}}^h$ in Eqs. (5.24) and (5.25) is obtained as follows. On the fluid–structure interface $(\Gamma_t)_{\mathrm{I}}$ the fluid domain mesh velocity $\hat{\mathbf{u}}^h$ is given by

$$\hat{\mathbf{u}}^h = \Pi_1^h \mathbf{u}_2^h, \tag{5.29}$$

where Π_1^h is a projection or interpolation operator onto the space spanned by the basis functions of the fluid mechanics problem restricted to the fluid–structure interface. In this work we use an L^2-projection. On the fluid mechanics domain interior the mesh velocity is obtained as a time derivative of an appropriate extension of the structural displacement on the fluid–structure boundary into the fluid domain interior. In the case of wind turbine simulations presented in this work, the differential equations of elastostatics (subject to time-dependent boundary conditions) are employed to handle the deflection part of the rotor structural motion, while the rotational part of the rotor motion is handled exactly (see Section 5.2.3 for details). Equation (5.29) couples the fluid mechanics problem to the structural mechanics problem, which is discussed in the next section.

To connect the discrete fluid mechanics formulation presented in this section to the continuous counterpart given by Eq. (5.20) in the previous section, we observe that: 1. The Galerkin terms are replaced with their ALE–VMS counterparts on the first line of Eq. (5.24); 2. The discrete counterpart of the variation of the fluid traction vector from the third line of Eq. (5.20) may be computed directly as

$$\delta_{\{\mathbf{u}_1^h, p^h\}} \boldsymbol{\sigma}_1 \mathbf{n}_1(\{\mathbf{w}_1^h, q^h\}) = 2\mu \boldsymbol{\varepsilon}(\mathbf{w}_1^h)\mathbf{n}_1 + q^h \mathbf{n}_1, \tag{5.30}$$

which motivates the third term in Eq. (5.24). However, note the sign change on the pressure test function, which is necessary for the pressure stability of the discrete formulation. The term on the fourth line of Eq. (5.24) is added to enhance the stability of

the formulation on $(\Gamma_t)_{\mathrm{I}}^-$. These enhancements do not violate consistency or adjoint consistency of the discrete formulation; 3. Finally, the penalty parameter β from the fourth line of Eq. (5.20) is set equal to τ_{B}. As a result, the penalty term of the augmented Lagrangian formulation directly translates to the penalty term of the weakly enforced boundary conditions.

5.2.2 Rotation-Free Kirchhoff–Love Shell Formulation of Structural Mechanics

In this work, we use a shell formulation for the structural mechanics problem. We assume that the shell midsurface is described using a piece-wise smooth (C^1-continuous or smoother) geometrical mapping. We also allow regions where the mapping reduces to the C^0 level. We are thinking of situations where the surface geometry of the shell is described using several NURBS patches, which are "glued" together with C^0 continuity, or as a single or multiple T-spline surfaces, also with a possible continuity reduction. For this reason we define

$$(\Gamma_2^s)_0 = \bigcup_{i=1}^{N_{\mathrm{sp}}} (\Gamma_2^s)_0^i, \tag{5.31}$$

$$(\Gamma_2^s)_t = \bigcup_{i=1}^{N_{\mathrm{sp}}} (\Gamma_2^s)_t^i, \tag{5.32}$$

$$(\Gamma_2^b)_0 = \bigcup_{i=1}^{N_{\mathrm{bp}}} (\Gamma_2^b)_0^i, \tag{5.33}$$

$$(\Gamma_2^b)_t = \bigcup_{i=1}^{N_{\mathrm{bp}}} (\Gamma_2^b)_t^i. \tag{5.34}$$

In the above equations, $(\Gamma_2^s)_0$ and $(\Gamma_2^s)_t$ denote the structure midsurface in the reference and deformed configuration, respectively, $(\Gamma_2^s)_0^i$ and $(\Gamma_2^s)_t^i$, $i = 1, 2, \ldots, N_{\mathrm{sp}}$, are the structural patches or subdomains in the reference and deformed configuration, respectively, and N_{sp} is their number. $(\Gamma_2^b)_0$ and $(\Gamma_2^b)_t$ denote the bending strip domain in the reference and deformed configuration, respectively, $(\Gamma_2^b)_0^i$ and $(\Gamma_2^b)_t^i$, $i = 1, 2, \ldots, N_{\mathrm{bp}}$, are the bending strip patch subdomains in the reference and deformed configuration, respectively, and N_{bp} is their number.

Let $\mathcal{S}_d^h$ and $\mathcal{V}_d^h$ denote the discrete trial and test function spaces for the structural problem. We state the discrete variational formulation of the rotation-free Kirchhoff–Love shell as: find the velocity of the shell midsurface $\mathbf{u}_2^h \in \mathcal{S}_d^h$, such that for all weighting functions $\mathbf{w}_2^h \in \mathcal{V}_d^h$,

$$\begin{aligned}
&\int_{(\Gamma_t^s)_2} \mathbf{w}_2^h \cdot \rho_2 h_{\text{th}} \left(\left.\frac{\partial \mathbf{u}_2^h}{\partial t}\right|_{\mathbf{X}} - \mathbf{f} \right) \mathrm{d}\Gamma \\
&+ \int_{(\Gamma_0^s)_2} \delta\overline{\boldsymbol{\epsilon}}^h \cdot \left(\mathbf{A}\, \overline{\boldsymbol{\epsilon}}^h + \mathbf{B}\, \overline{\boldsymbol{\kappa}}^h \right) \mathrm{d}\Gamma + \int_{(\Gamma_0^s)_2} \delta\overline{\boldsymbol{\kappa}}^h \cdot \left(\mathbf{B}\, \overline{\boldsymbol{\epsilon}}^h + \mathbf{D}\, \overline{\boldsymbol{\kappa}}^h \right) \mathrm{d}\Gamma \\
&+ \int_{(\Gamma_0^b)_2} \delta\overline{\boldsymbol{\kappa}}^h \cdot \mathbf{D}_{\text{b}}\, \overline{\boldsymbol{\kappa}}^h \, \mathrm{d}\Gamma - \int_{(\Gamma_t)_1} \mathbf{w}_2^h \cdot (\Pi_2^h \mathbf{t}_1^h) \, \mathrm{d}\Gamma = 0.
\end{aligned} \tag{5.35}$$

In the above formulation, h_{th} is the shell thickness, $\overline{\boldsymbol{\epsilon}}^h$ and $\overline{\boldsymbol{\kappa}}^h$ are the tensors of membrane strains and curvature changes, written in Voigt notation and with respect to the local Cartesian basis oriented on the first covariant basis vector of the midsurface, $\delta\overline{\boldsymbol{\epsilon}}^h$ and $\delta\overline{\boldsymbol{\kappa}}^h$ are their variations, $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{D}$ are the membrane, coupling and bending stiffnesses, respectively, pre-integrated through the thickness, and $\mathbf{D}_{\text{b}}$ is the bending stiffness of the bending strips.

In the last term on the left-hand-side of Eq. (5.35), $\mathbf{t}_1^h$ is the discrete counterpart of the fluid traction vector from Eq. (5.22) and is given by

$$\mathbf{t}_1^h = -\boldsymbol{\sigma}_1^h \mathbf{n}_1 - \tau_{\text{B}} \left(\hat{\mathbf{u}}^h - \mathbf{u}_1^h \right), \tag{5.36}$$

where the mesh velocity $\hat{\mathbf{u}}^h$ is obtained from the structural velocity $\mathbf{u}_2^h$ using Eq. (5.29), and Π_2^h is a projection or interpolation operator onto the space spanned by the basis functions of the structural mechanics problem restricted to the fluid–structure interface. Here, Π_2^h corresponds to the L^2-projection operator.

While Eq. (5.35) is able to accommodate large structural deformations, it is assumed that the stress–strain relationship is linear (i.e., the St. Venant–Kirchhoff material model is used). Composite materials, which are typically employed in the manufacturing of wind turbine blades, are modeled within this framework by homogenizing the structural material properties (density and stiffness) in the through-thickness direction. See Section 4.1 for more details.

5.2.3 Motion of the Fluid Domain (Fluid Mesh Update)

In the ALE framework the fluid domain moves to accommodate the structural motion. To ensure a smooth evolution of the fluid domain, one can employ the equations of linear elastostatics subject to dynamic boundary conditions coming from the structural displacement to update the kinematics (position and velocity) of the fluid mesh [56, 57, 146–148]. However, in the case of wind turbine rotors, where the motions are dominated by rotation, this may not be a preferred procedure due to the fact that the linear elastostatics operator does not vanish on large rotational motions (i.e., large rotations generate stresses in the linear elastic framework). This, in turn, may lead to the loss of the fluid mesh quality if one plans to simulate the FSI problem for many revolutions of the wind turbine rotor.

To circumvent this potential difficulty, we modify our fluid mesh motion strategy as follows. We decompose the structural displacement vector into the rotation and deflection parts (see Section 4.3). As the increment of the structural displacement is computed, we extract the deflection part, apply the elasticity-based mesh-moving method [146–148] to compute only the deflection part of the mesh displacement, rotate the (deformed) mesh from the previous time level to the current time, and add the mesh deflection increment to obtain its current position. A precise mathematical formulation of this procedure is given as follows: we introduce the total fluid domain displacement $\mathbf{d}_3$ as

$$\mathbf{d}_3 = \mathbf{d}_3^{\theta} + \mathbf{d}_3^{d}, \tag{5.37}$$

where

$$\mathbf{d}_3^{\theta} = (\mathbf{R}(\theta) - \mathbf{I})\,(\mathbf{X} - \mathbf{X}_0) \tag{5.38}$$

is the fluid domain displacement associated with rotation about a fixed point $\mathbf{X}_0$, $\mathbf{X}$ are the coordinates of the structure reference configuration, θ is the time-dependent angle of rotation, $\mathbf{R}(\theta)$ is the rotation matrix given by Eq. (4.46), and $\mathbf{I}$ is the identity matrix. The fluid domain deflection may be computed from the following variational problem: find $\mathbf{d}_3^{d}(t) \in \mathcal{S}_m$, such that $\forall \mathbf{w}_3 \in \mathcal{V}_m$,

$$B_3\left(\mathbf{w}_3, \mathbf{d}_3^{d}(t)\right) - F_3\left(\mathbf{w}_3\right) = 0, \tag{5.39}$$

where the argument t is used to denote deflection at the current time. In Eq. (5.39),

$$B_3\left(\mathbf{w},\mathbf{d}\right) = \int_{(\Omega_{\tilde{t}})_1} \boldsymbol{\nabla}^s_{\tilde{x}}\mathbf{w} : \left(2\mu_3\boldsymbol{\nabla}^s_{\tilde{x}}\mathbf{d} + \lambda_3(\boldsymbol{\nabla}_{\tilde{x}}\cdot\mathbf{d})\,\mathbf{I}\right)\mathrm{d}\Omega, \tag{5.40}$$

where

$$\boldsymbol{\nabla}^s_{\tilde{x}}\mathbf{w} = \frac{1}{2}\left(\boldsymbol{\nabla}_{\tilde{x}}\mathbf{w} + \boldsymbol{\nabla}_{\tilde{x}}\mathbf{w}^T\right), \tag{5.41}$$

and μ_3 and λ_3 are the Lamé parameters of the fictitious linear elastic model characterizing the motion of the fluid domain. In the discrete setting μ_3 and λ_3 should be selected such that the fluid mesh quality is preserved for as long as possible (see, e.g., Johnson and Tezduyar [148]). Equation (5.40) is the linear elasticity operator defined on the "nearby configuration" $(\Omega_{\tilde{t}})_1$ taken at time $\tilde{t} < t$. In Eq. (5.39),

$$F_3\left(\mathbf{w}_3\right) = B_3\left(\mathbf{w}_3, \mathbf{d}^d_3(\tilde{t})\right), \tag{5.42}$$

where $\mathbf{d}^d_3(\tilde{t})$ is considered known. The configuration $(\Omega_{\tilde{t}})_1$ is obtained by rotating the fluid domain to current time t and deflecting to $\tilde{t}$, namely,

$$(\Omega_{\tilde{t}})_1 = \left\{\tilde{\mathbf{x}} \mid \tilde{\mathbf{x}} = \mathbf{X} + \mathbf{d}^\theta_3(t) + \mathbf{d}^d_3(\tilde{t})\right\}. \tag{5.43}$$

In practice, $\tilde{t} = t^n$, that is, in the definition of $(\Omega_{\tilde{t}})_1$, the deflection is taken at the previous time step. The total fluid domain displacement at time t is now computed as

$$\mathbf{d}_3(t) = \mathbf{d}^\theta_3(t) + \mathbf{d}^d_3(t), \tag{5.44}$$

and the fluid domain configuration at t becomes

$$(\Omega_t)_1 = \left\{\mathbf{x} \mid \mathbf{x} = \mathbf{X} + \mathbf{d}_3(t)\right\}. \tag{5.45}$$

Remark 5.6. The pure rotation case is recovered by setting $\mathbf{d}^d_3 = \mathbf{0}$. Likewise, the pure deflection case is recovered by setting $\mathbf{d}^\theta_3 = \mathbf{0}$.

Remark 5.7. As an alternative to the proposed approach, nonlinear elasticity with an objective measure of strain may be used to compute the fluid domain motion. This, however, introduces an additional level of nonlinearity into the problem.

5.2.4 Time Integration of the FSI Equations and Coupling

The ALE formulation for fluid mechanics and the Lagrangian formulation for the structural mechanics give a natural setting for a finite-difference-in-time time integration of this coupled FSI system. We employ the Generalized-α technique [57, 110, 111] in this work, which is a fully-implicit second-order accurate method with control over the dissipation of high-frequency modes. At each time step the combined fluid, structure and mesh motion discrete residuals are converged to zero by means of a block-iterative FSI coupling approach (see Tezduyar *et al.* [112, 149, 150] for the terminology). The block-iterative approach consists of repeating the following sequence of linear solves: 1. We obtain the fluid mechanics solution increment holding the structural and mesh solutions fixed. 2. We update the fluid mechanics solution, compute the fluid traction force on the structure, and compute the structural solution increment; 3. We update the structural solution and use elastic mesh motion to update the fluid domain velocity and position. We note that only the deflection part of the mesh motion is computed using linear elastostatics, while the rotation part is computed exactly (see Section 5.2.3). This three-step iteration sequence is repeated until convergence to an appropriately coupled discrete solution is achieved. The block-iterative approach to FSI coupling is stable for our application, in part because wind turbine blades are heavy relative to the surrounding air. In addition, the fluid mechanics problem is not posed on an enclosed domain, which presents difficulties for incompressible flow [151]. The block-iterative FSI methodology for the simulation of wind turbine rotors is illustrated in Figure 5.1.

Remark 5.8. For this coupling strategy the fluid and structural meshes may or may not be conforming. In the case of conforming meshes, the conservative nodal or control-point traction vector from the fluid side is applied directly at the nodes or control points of the structure, while the structural nodal or control-point kinematic data is applied directly to the nodes or control points of the fluid. When the fluid and structural meshes are non-conforming, additional projection of the traction and kinematic data is necessary before they are transferred to the neighboring subdomain.

Remark 5.9. So far, we discussed weakly enforced boundary conditions for the fluid mechanics problem. In the case where strong boundary conditions are employed, we

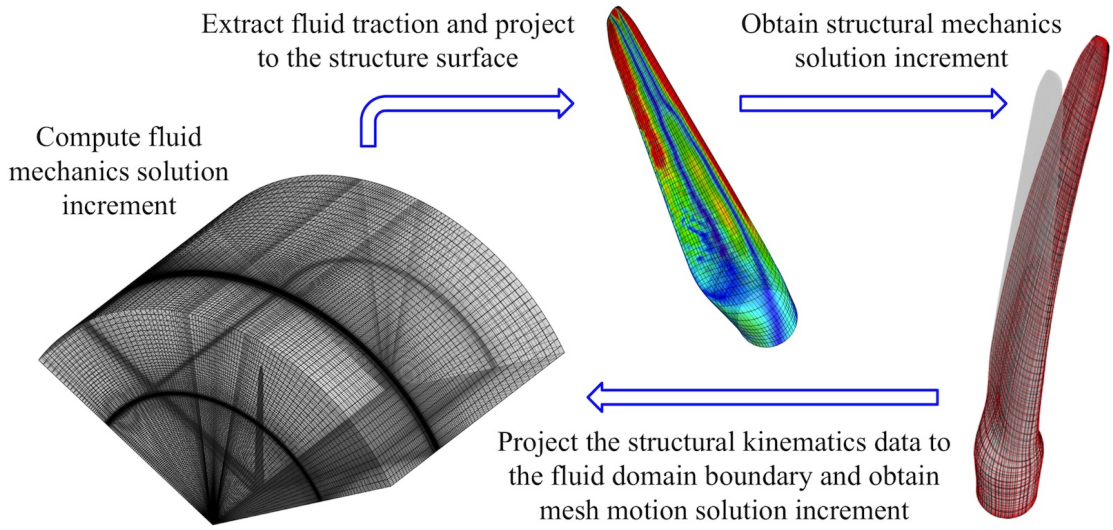


Figure 5.1: Illustration of the block-iterative FSI methodology for the simulation of wind turbine rotors. Due to the relatively heavy weight of the wind turbine blades with respect to the surrounding air, this type of coupling is sufficient for rapid convergence of the coupled nonlinear FSI equation system.

propose to modify the fluid mechanics formulation and the FSI coupling procedures as follows:

- Set $\mathbf{u}_1^h = \hat{\mathbf{u}}^h = \Pi_1^h \mathbf{u}_2^h$ strongly on $(\Gamma_t)_\mathrm{I}$. This reduces the discrete formulation of the fluid mechanics problem given by Eq. (5.24) to: find $\mathbf{u}_1^h$ and p^h such that for all $(\mathbf{w}_1^h)_0$ and q^h,

$$B_1^{\mathrm{VMS}}\left(\{(\mathbf{w}_1^h)_0, q^h\}, \{\mathbf{u}_1^h, p^h\}\right) - F_1^{\mathrm{VMS}}\left(\{(\mathbf{w}_1^h)_0, q^h\}\right) = 0, \tag{5.46}$$

  where the subscript 0 on $\mathbf{w}_1^h$ indicates that a subset of the fluid mechanics test functions, which are not supported on the fluid–structure interface, is employed.

- Given the fluid mechanics solution from the previous step, compute the fluid traction vector on the fluid–structure interface as: find $\mathbf{t}_1^h$, such that for all $(\mathbf{w}_1^h)_g$,

$$\int_{(\Gamma_t)_\mathrm{I}} (\mathbf{w}_1^h)_g \cdot \mathbf{t}_1^h \, \mathrm{d}\Gamma = B_1^{\mathrm{VMS}}\left(\{(\mathbf{w}_1^h)_g, q^h\}, \{\mathbf{u}_1^h, p^h\}\right) - F_1^{\mathrm{VMS}}\left(\{(\mathbf{w}_1^h)_g, q^h\}\right), \tag{5.47}$$

  where the subscript g on $\mathbf{w}_1^h$ indicates that a subset of the fluid mechanics test functions, which are supported on the fluid–structure interface, is employed. This is known as the conservative definition of the traction vector on the boundary with

strongly enforced essential boundary conditions (see, e.g., Melbø and Kvamsdal [152] and van Brummelen *et al.* [153] for the importance of using conservative flux definitions on essential boundaries and in coupled problems).

- Project the computed fluid mechanics traction vector $\mathbf{t}_1^h$ to the structural mechanics domain discretization of the fluid–structure interface and solve the structural mechanics problem. In the case of matching fluid–structure interface discretization, no projection is necessary.

- Update the fluid mechanics mesh position and velocity, check for convergence of the fluid, structure and mesh residuals, and repeat the iteration if necessary. If not, go to the next time step.

5.2.5 Data Transfer for IGA and FEM Surface Discretizations

Non-matching interface discretizations in the FSI problem necessitate the use of interpolation or projection of kinematic and traction data between the fluid and structural surface meshes (see, e.g., Farhat *et al.* [154] and Takizawa and Tezduyar [101], or the developments in the previous sections). Here we describe the computational procedures, which can simultaneously handle IGA and FEM discretizations.

Let f_1^h and f_2^h be the arbitrary functions defined on two surfaces Γ_1^h and Γ_2^h, respectively. The functions are represented in the discrete IGA or FEM spaces defined on the surfaces, which may be the standalone surfaces or boundaries of the corresponding volumetric domains. We also define Π_1^h and Π_2^h to be the L^2-projection operators corresponding to the discrete function spaces defined on Γ_1^h and Γ_2^h, respectively. We write

$$f_1^h = \Pi_1^h f_2^h \tag{5.48}$$

to mean that f_1^h is an L^2-projection of f_2^h onto the space of functions defined on Γ_1^h. In the case Γ_1^h and Γ_2^h coincide geometrically, Eq. (5.48) is unambiguous. However, when the two domains are mismatched, which is typically the case for FSI involving complex structural surfaces, the L^2-projection given by Eq. (5.48) requires a more careful consideration.

The matrix form of Eq. (5.48) may be written as

$$\sum_{B\in\boldsymbol{\eta}_1}\int_{\Gamma_1^h} N_A N_B \,\mathrm{d}\Gamma f_B = \int_{\Gamma_1^h} N_A f_2^h \,\mathrm{d}\Gamma, \tag{5.49}$$

where N_A's are the basis functions defined on Γ_1^h, $\boldsymbol{\eta}_1$ is their set, f_B's are the discrete solution coefficients of the L^2-projection problem giving $f_1^h = \sum_{B\in\boldsymbol{\eta}_1} N_B f_B$, and $A, B \in \boldsymbol{\eta}_1$. The Gauss quadrature (or other numerical integration) approximation of the right-hand-side of Eq. (5.49) gives

$$\int_{\Gamma_1^h} N_A f_2^h \,\mathrm{d}\Gamma \approx \sum_{e=1}^{N_{el}}\sum_{i=1}^{N_{int}^e} N_A(\boldsymbol{\zeta}_i^e) f_2^h(\boldsymbol{\zeta}_i^e) W_i^e J(\boldsymbol{\zeta}_i^e), \tag{5.50}$$

where N_{el} is the number of elements on Γ_1^h, N_{int}^e is a number of Gauss points used on element e, $\boldsymbol{\zeta}_i^e$'s are the locations of the Gauss points in the parametric domain of element e, W_i^e's are the corresponding quadrature weights, and J is the surface Jacobian determinant.

In the case of surface discretizations matching geometrically, the quadrature formula given by Eq. (5.50) may be evaluated directly. In the case of non-matching discretizations, however, "$f_2^h(\boldsymbol{\zeta}_i^e)$" needs to be suitably defined. We do this as follows: for every quadrature point on Γ_1^h we find its *closest point* on Γ_2^h and evaluate f_2^h there. The closest point, illustrated in Figure 5.2, is defined as follows: if $\mathbf{x}_G$ is a generic point on Γ_1^h, then its closest point $\mathbf{x}_2^e(\boldsymbol{\xi}^e)$ on Γ_2^h is such that the distance $\|\mathbf{x}_2^e(\boldsymbol{\xi}^e) - \mathbf{x}_G\|$ is minimized. Here $\mathbf{x}_2^e(\boldsymbol{\xi}^e)$ is the geometrical mapping defined on a parametric element $\hat{\Gamma}_2^e$ of the surface Γ_2^h, and $\boldsymbol{\xi}^e$ are the parametric coordinates of this element. The closest point is characterized by the solution of the following problem: find $\hat{\Gamma}_2^e$ and $\boldsymbol{\xi}^e = \{\xi_1^e, \xi_2^e\}^T \in \hat{\Gamma}_2^e$ such that

$$\frac{\partial \mathbf{x}_2^e(\boldsymbol{\xi}^e)}{\partial \xi_1^e} \cdot (\mathbf{x}_2^e(\boldsymbol{\xi}^e) - \mathbf{x}_G) = 0, \tag{5.51}$$

$$\frac{\partial \mathbf{x}_2^e(\boldsymbol{\xi}^e)}{\partial \xi_2^e} \cdot (\mathbf{x}_2^e(\boldsymbol{\xi}^e) - \mathbf{x}_G) = 0, \tag{5.52}$$

that is, the distance vector $(\mathbf{x}_2^e(\boldsymbol{\xi}^e) - \mathbf{x}_G)$ is orthogonal to the two tangent vectors $\frac{\partial \mathbf{x}_2^e(\boldsymbol{\xi}^e)}{\partial \xi_1^e}$ and $\frac{\partial \mathbf{x}_2^e(\boldsymbol{\xi}^e)}{\partial \xi_2^e}$ of the surface Γ_2^h. The 2×2 system of Eqs. (5.51) and (5.52) is solved using the Newton–Raphson iteration procedure. We find that a complete linearization of the equations is important for rapid convergence of the iterations.

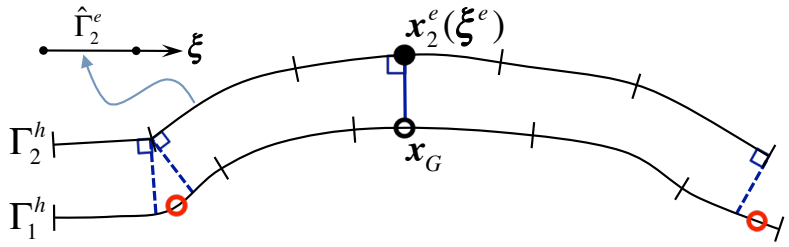


Figure 5.2: Illustration of the closest point projection, and the associated difficulties, using 2D curves. Vertical notches on the curves denote element boundaries in the physical domain.

Remark 5.10. Note that the problem of finding the closest point and using it to define a projection between two possibly non-matching surfaces is form-identical in IGA and FEM. All that is needed is a parameterized surface, the notion of a parametric element, and the ability to define a set of linearly independent basis functions on the surface. These features are common to both IGA and FEM.

Existence (and uniqueness) of closest points depends on several factors. Situations that may lead to non-existence of solutions in the sense of Eqs. (5.51) and (5.52) typically arise in the case of non-smooth surfaces with discontinuous tangent vectors, or in the case of surfaces that are not properly aligned (see Figure 5.2 for an illustration). As a result, the procedure described above needs to be augmented to account for these situations. In what follows, we describe an approach that appears to be robust in identifying closest points for arbitrary shaped surfaces.

We assume that for a given $\mathbf{x}_G$ we solved the nonlinear system given by Eqs. (5.51) and (5.52) for $\{\xi_1^e, \xi_2^e\}^T$ on every element $\hat{\Gamma}_2^e$. We also assume that $\hat{\Gamma}_2^e$ is either a bi-unit quad or an isosceles right triangle (see Figure 5.3). In the case of rectangular topology, for an appropriate subset of elements in Γ_2^h, we go through the following cases:

1. If $|\xi_1^e| \leq 1$ and $|\xi_2^e| \leq 1$, store ξ_1^e and ξ_2^e.

2. If $|\xi_1^e| > 1$ and $|\xi_2^e| \leq 1$,

$$\text{set } \xi_1^e = \text{sgn}(\xi_1^e), \text{ then solve } \frac{\partial \mathbf{x}_2^e(\boldsymbol{\xi}^e)}{\partial \xi_2^e} \cdot (\mathbf{x}_2^e(\boldsymbol{\xi}^e) - \mathbf{x}_G) = 0.$$

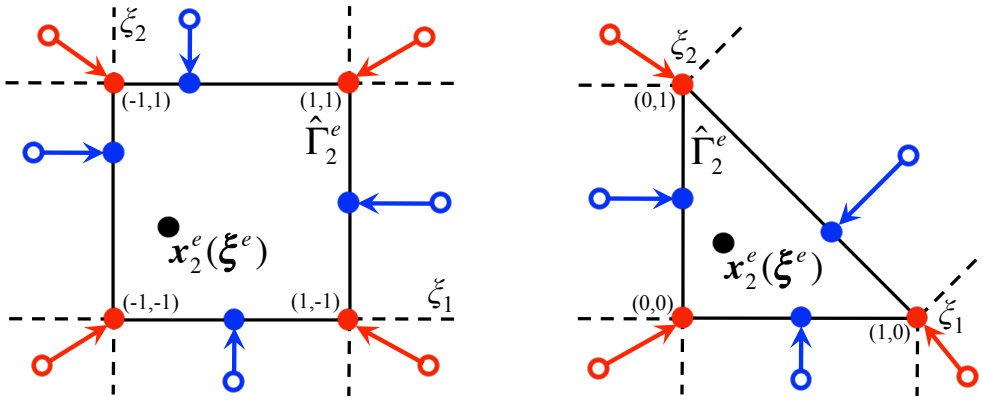
3. If $|\xi_1^e| \leq 1$ and $|\xi_2^e| > 1$,

$$\text{set } \xi_2^e = \text{sgn}(\xi_2^e), \text{ then solve } \frac{\partial \mathbf{x}_2^e(\boldsymbol{\xi}^e)}{\partial \xi_1^e} \cdot (\mathbf{x}_2^e(\boldsymbol{\xi}^e) - \mathbf{x}_G) = 0.$$

4. If $|\xi_1^e| > 1$ and $|\xi_2^e| > 1$, or if the above case 2 or 3 generates a parameter that is out of bounds,

$$\text{set } \xi_1^e = \text{sgn}(\xi_1^e), \text{ and set } \xi_2^e = \text{sgn}(\xi_2^e).$$

At this point each element in the subset generates one candidate for the closest point, and we select the candidate that minimizes $\|(\mathbf{x}_2^e(\boldsymbol{\xi}^e) - \mathbf{x}_G)\|$. The four cases described above are illustrated in Figure 5.3(a). The triangular topology case, shown in Figure 5.3(b) is handled in an analogous fashion. The rectangular topology approach is suitable for NURBS, T-spline, Catmull–Clark subdivision, and quadrilateral FEM surfaces, while the triangular topology approach may be used for Loop subdivison and triangular FEM surfaces.



(a) Rectangular topology (b) Triangular topology

Figure 5.3: Illustration of the procedure of finding the closest point in the case the Newton–Raphson iteration converges to a parametric value outside an element. In this case, additional candidates for the closest point are searched on element boundaries and corners. The procedure is equally applicable to the case of triangular and rectangular topology of the underlying surface.

5.3 Computational Testing of the Data Transfer Algorithm

Before presenting the FSI results, we test the performance of the data transfer algorithm. For this, the NREL 5 MW wind turbine rotor, assumed rigid, is simulated at prescribed steady inlet wind velocity of 11.4 m/s and rotor angular velocity of 12.1 rpm. The dimensions of the problem domain, the NURBS mesh employed, and the overall problem setup are the same as in Section 3.3.2. The aerodynamic traction vector is collected on the NURBS surface of the rotor, and used in the structural computations of a single wind turbine blade whose surface meshes are shown in Figure 5.4. Here we employ two discretizations of the blade. One is based on quadratic NURBS and the other on cubic T-splines. The T-spline surface was obtained by reparameterizing the NURBS surface. Singularity in the NURBS geometrical mapping due to coalesced control points at the blade tip is removed during the reparameterization process (see Figure 5.5). Instead, several extraordinary points [78, 155] are used to model the geometry in the vicinity of the tip. The T-spline reparameterization introduced only very minor changes in the blade geometry (e.g., the blade surface area was reduced by about 0.5%). The geometry error may be minimized further by locally adjusting the positions of the T-spline mesh control points.

The NURBS model of the blade, which makes use of the Kirchhoff–Love rotation-free composite shell formulation, in conjunction with the bending strip method, is described in detail in Section 4.2.3. There are 4,897 control points in the NURBS mesh. The T-spline model is based on the same structural mechanics formulation, however, no bending strips are employed. The underlying T-spline basis functions have sufficient continuity almost everywhere on the blade surface to obviate the use of bending strips. The T-spline mesh has 3,572 control points.

We compute the blade static response under the action of the fluid traction vector. We also include the effect of the centripetal force due to rotation, which is significant. In the case of the NURBS blade, due to the fact that the fluid and structural meshes are matching at the interface, no data transfer is necessary. The predicted blade tip deflection is 3.6098 m. We repeat the calculation for the T-spline blade. The T-spline

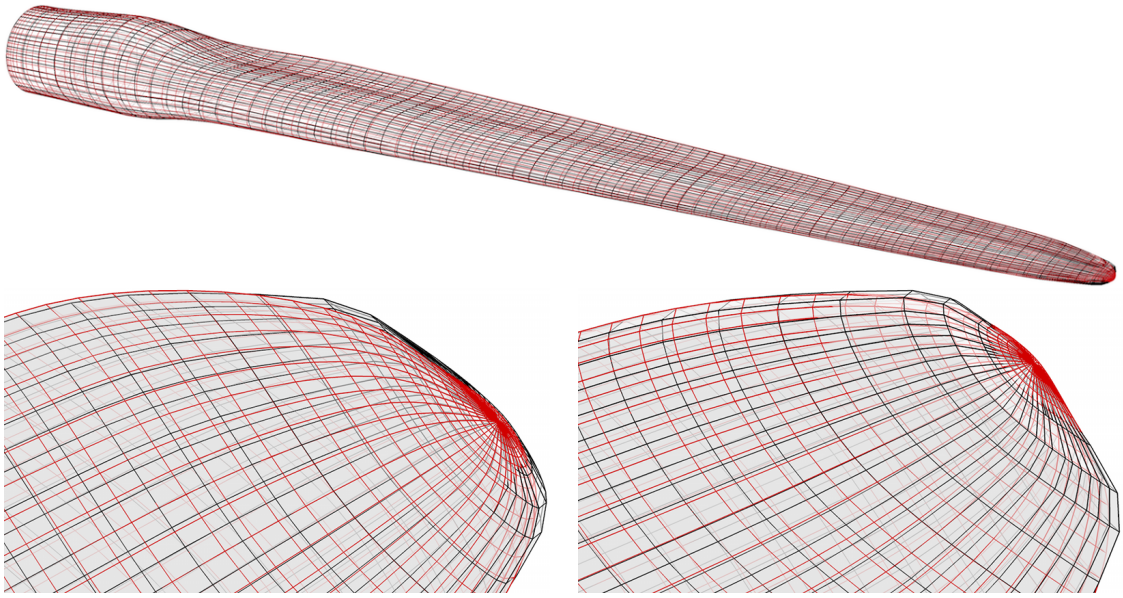
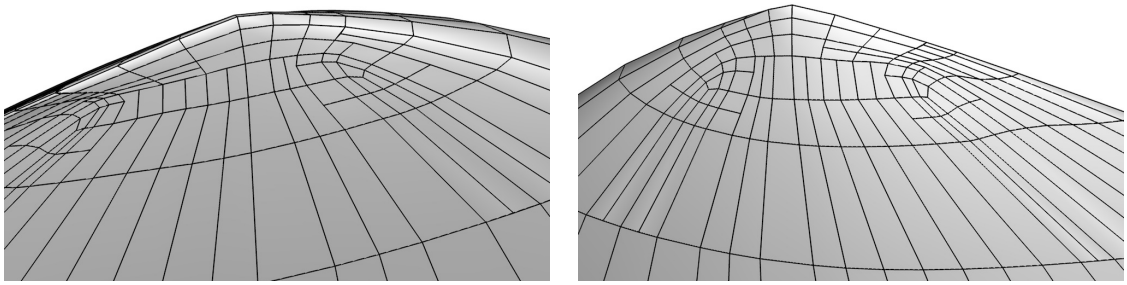


Figure 5.4: Surface meshes of the wind turbine blade used in the computations in this paper. The black mesh lines correspond to a bi-cubic T-spline discretization and the red mesh lines correspond to a bi-quadratic NURBS discretization. Top: complete blade surface. Bottom: zoom on the blade tip, where the differences between the two surfaces are clearly visible.



(a) Pressure side (b) Suction side

Figure 5.5: Zoom on the tip of the blade surface mesh reparameterized by T-splines. Note that the tip singularity is removed. Several extraordinary points [78, 155] are used to model the geometry in the vicinity of the tip.

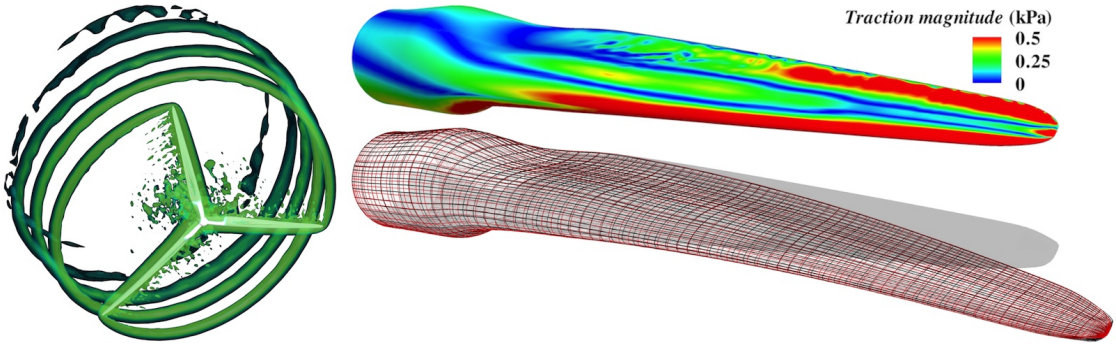


Figure 5.6: Left: Aerodynamics of the rigidly rotating wind turbine rotor. Isosurfaces of wind speed at a time instant. Top right: Magnitude of the aerodynamic traction vector on the blade surface. Bottom right: The blade in the reference and deformed configurations. NURBS and T-spline meshes are superposed on the deformed configuration, showing no visible differences between the two structural mechanics solutions.

implementation in the isogeometric Kirchhoff–Love shell code makes use of the Bézier extraction decomposition of the analysis geometry (see Borden *et al.* [82] and Scott *et al.* [83]). In this case, we project the aerodynamic traction vector from the boundary of the fluid mechanics NURBS mesh to the T-spline structural mesh, compute the structural displacement taking the centripetal force into account, and project the displacement back to the NURBS mesh. In this case, the tip deflection is predicted to be 3.6193 m. This gives the relative error of 0.26% between the matching and non-matching discretization simulations, which is felt to be almost negligible. We also assessed the accuracy of the load transfer by comparing the global aerodynamic force and moment on the NURBS and T-spline meshes. After the load transfer, the error in the global force is only 0.14%, and the error in the global moment is only 0.10%. The results of the aerodynamics and two structural computations are shown in Figure 5.6.

5.4 FSI Simulations of the NREL 5 MW Offshore Baseline Wind Turbine Rotor

In this section we present FSI simulations of the NREL 5 MW offshore baseline wind turbine rotor. We first show the all-NURBS (fluid and structure) matching-interface computation, which we take as a benchmark calculation. We then show the

simulation where we couple the NURBS fluid mechanics and T-spline structural mechanics using a non-matching discretization of the fluid–structure interface. Finally we show the results of the coupling between low-order FEM for fluid mechanics and T-splines for structural mechanics.

The FSI coupling procedure employed in all cases is described in Section 5.2.4 and is also illustrated in Figure 5.1. The strong boundary condition version of the FSI formulation is employed in the case of NURBS fluid mechanics coupled with NURBS or T-spline structural mechanics. This is due to the fact that the NURBS fluid mechanics mesh has sufficient boundary layer refinement to produce accurate results for the quantities of interest in the simulation. In the case of FEM and T-spline coupling, because of the lack of adequate boundary layer resolution, a weakly enforced boundary condition version of the FSI coupling method is employed.

In all cases we use steady inlet wind velocity of 11.4 m/s and rotor angular velocity of 12.1 rpm. The dimensions of the computational domain, the NURBS mesh employed, and the overall problem setup are the same as in Section 3.3.2. The structural definition of the wind turbine blade is described in detail in Section 4.2.3.

5.4.1 Coupling of NURBS for Fluid and NURBS for Structural Mechanics

In this section we show the all-NURBS (fluid and structure) matching-interface computation, which we take as a benchmark calculation. Rotationally periodic boundary conditions for the fluid are imposed in order to reduce computational cost. However, because the rotor blades are subject to gravity forces, a fully rotationally periodic structural solution is not expected in this case. Nevertheless, we feel that the use of rotationally periodic boundary conditions for the fluid domain is justified due to the fact that the fluid periodic boundaries are located sufficiently far away from the structure and are not expected to affect the structural response.

The computations are advanced in time until a statistically-stationary value of the aerodynamic torque is obtained. The time step size is chosen to be $\Delta t = 0.0003$ s. The rigid rotor under the same wind and rotor speed conditions is simulated for comparison. Contours of the pressure on the flexible blade in the current configuration are shown in

Figure 5.7. The large negative pressure on the suction side creates a lift force vector with a component in the direction of the blade rotation, which generates a favorable aerodynamic torque.

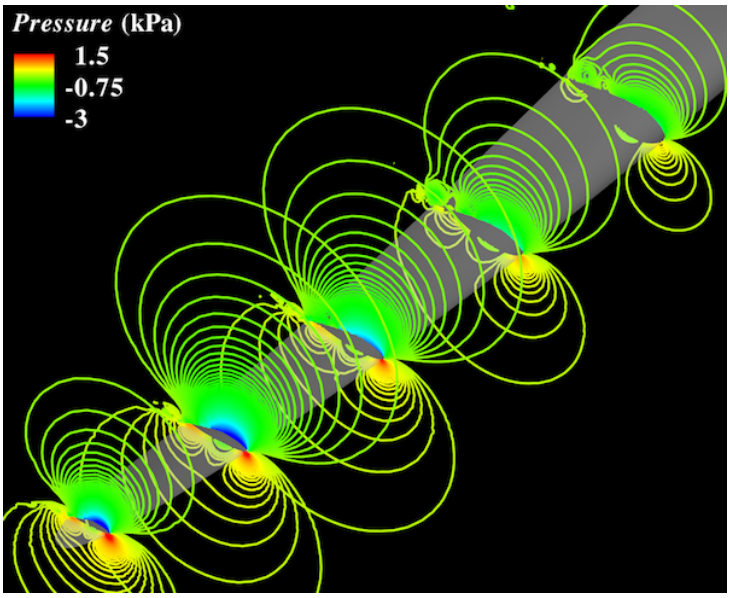


Figure 5.7: Pressure contours at several blade cross-sections at t = 0.7 s viewed from the back of the blade plotted on the deformed configuration. The large negative pressure at the suction side of the airfoil creates a favorable aerodynamic torque.

Rotor blade deflected shape at the point of maximum tip displacement is shown in Figure 5.8. As expected, the blade mostly displaces in the flapwise direction, although some edgewise deflection is also present. Time histories of the flapwise and edgewise displacements are shown in Figure 5.9. The maximum flapwise tip deflection reaches nearly 6 m, which is significant, and is consistent with the data reported in Jonkman *et al.* [4]. There is a sudden decrease in the edgewise deflection around $t = 1.2$ s. At that time the blade tip passes its lowest vertical position (see Figure 5.10 for the blade location at different time instances) and the direction of the gravity force vector reverses with respect to the direction of the lift force vector.

Isosurfaces of the air speed at different time instances are shown in Figure 5.11. Note that, for visualization purposes the rotationally periodic 120° domain was merged into a full 360° domain. Fine-grained turbulent structures are generated at the trailing edge of the blade along its entire length. The vortex forming at the tip of the blades is

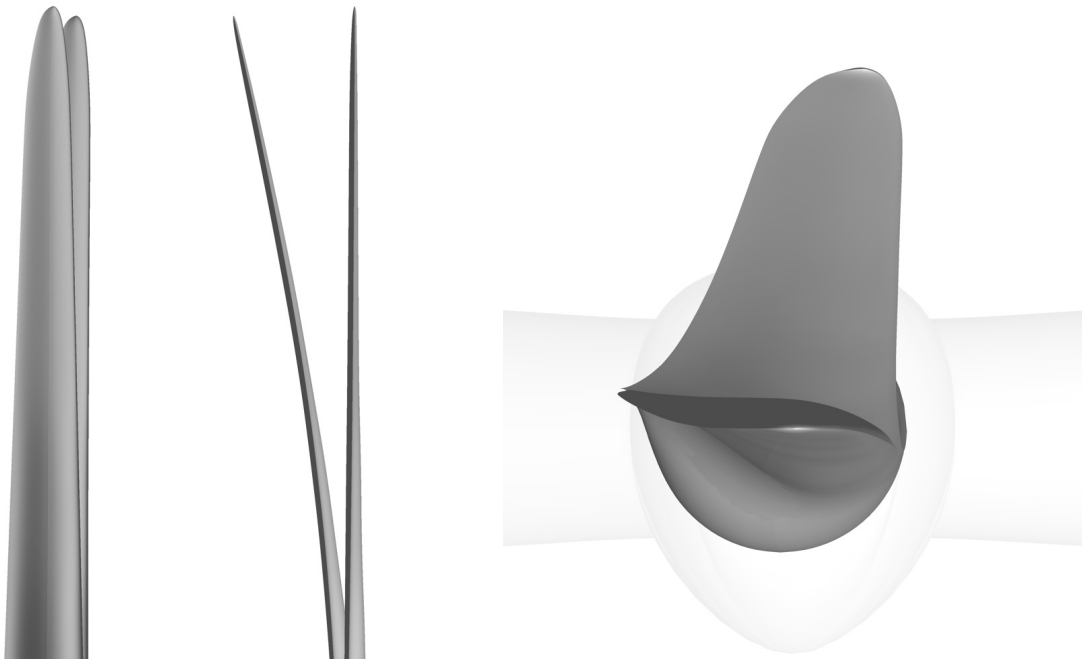


Figure 5.8: Rotor blade deflected shape at the point of maximum tip displacement. Front, side and top views are shown to better illustrate blade deflection characteristics.

convected downstream of the rotor with little decay.

Figure 5.12 shows the isocontours of air speed at a planar cut superposed on the spinning rotor. Note the high-intensity turbulence in the blade aerodynamic zone, which is a segment of the blade where the cylindrical root rapidly transitions to a thin airfoil shape. This suggests that the blade trailing edge in this location is subjected to high-frequency loads that are fatiguing the blade. The blade displacement under the action of wind forces is also clearly visible.

Figure 5.13 shows the isocontours of relative wind speed at a 30 m radial cut at different time instances. For every snapshot the blade is rotated to the reference configuration to better illustrate the deflection part of the motion. On the pressure side, the air flow boundary layer is attached to the blade for the entire cord length. On the suction side, the flow detaches near the trailing edge and transitions to turbulence.

The aerodynamic torque is plotted in Figure 5.14 for both rigid and flexible blade simulations. Both cases compare favorably to the data reported by Jonkman *et al.* [4]. Note that the aerodynamic torque for the flexible blade exhibits low-magnitude,

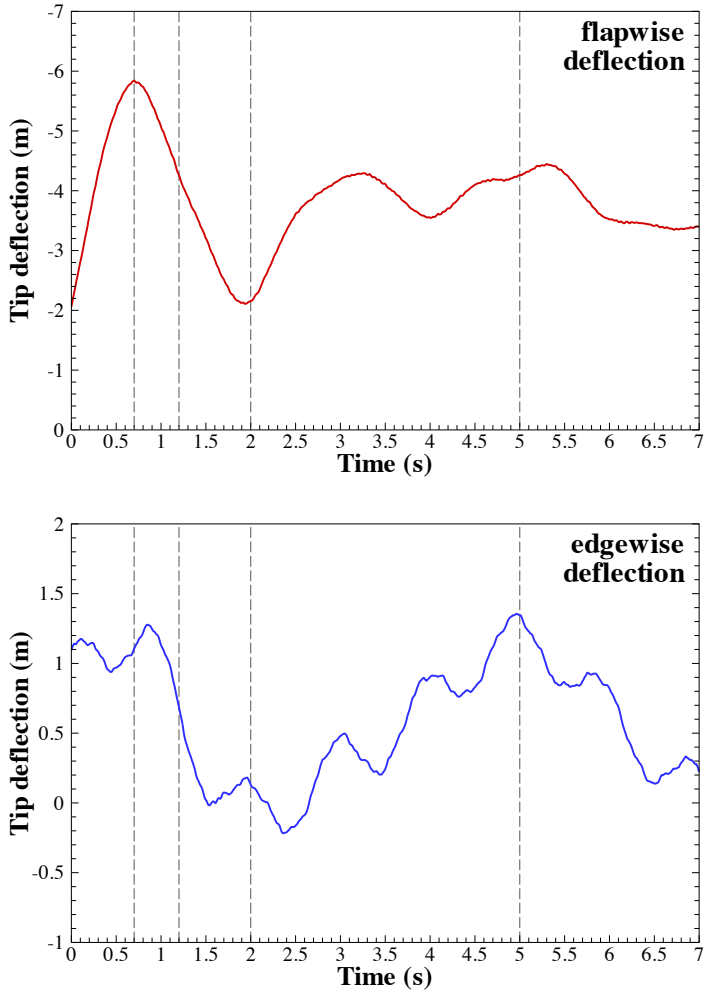
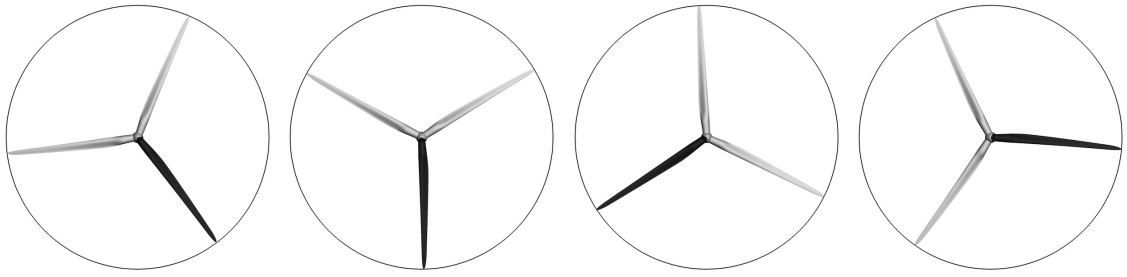
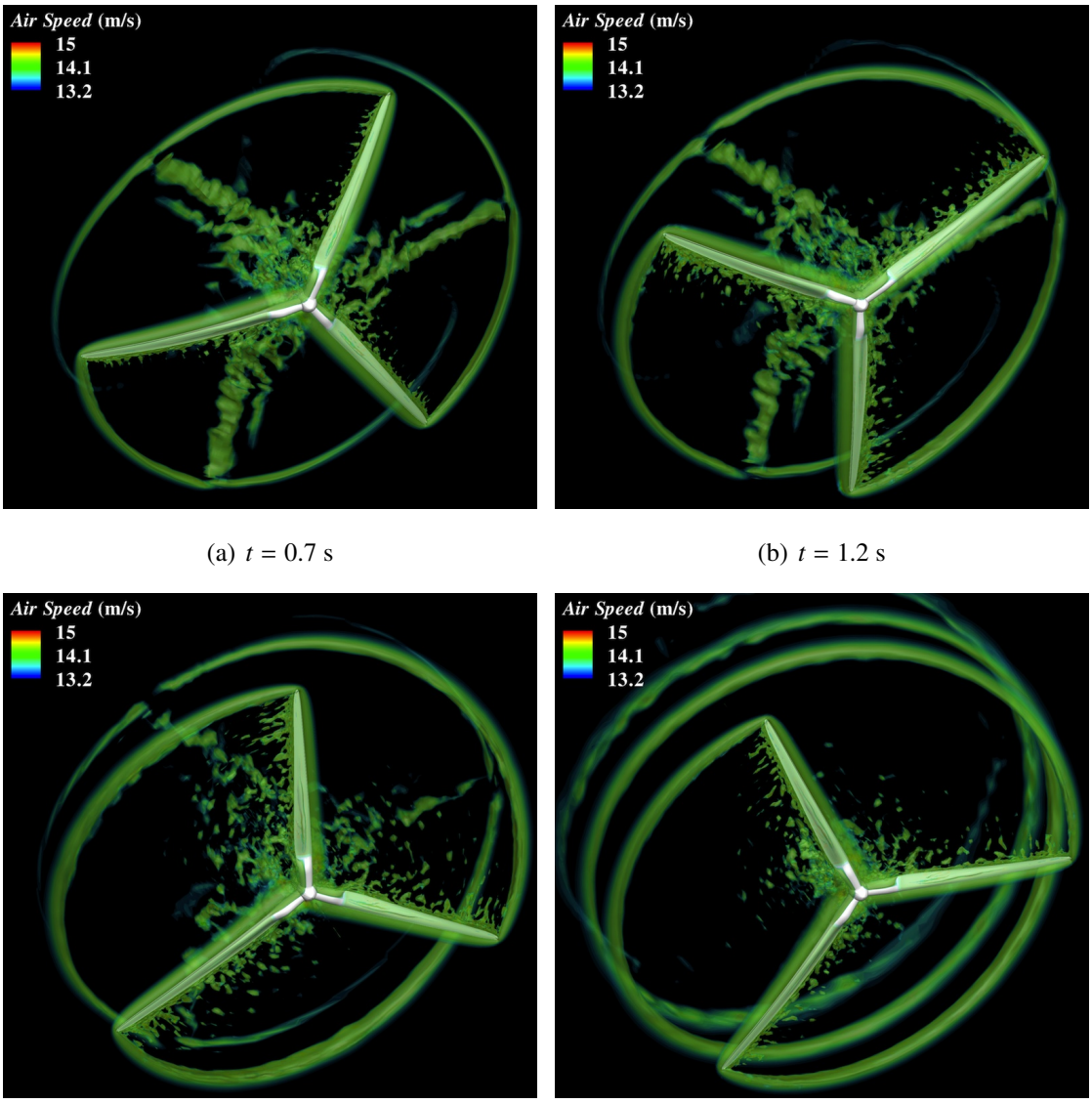


Figure 5.9: Time histories of the blade tip flapwise (front-to-back) and edgewise (side-to-side) deflection.



(a) t = 0.7 s (b) t = 1.2 s (c) t = 2.0 s (d) t = 5.0 s

Figure 5.10: Merged domain of the rotor configuration as several time instants during the simulation. The actual computational domain of the wind turbine rotor is highlighted using a darker shade.



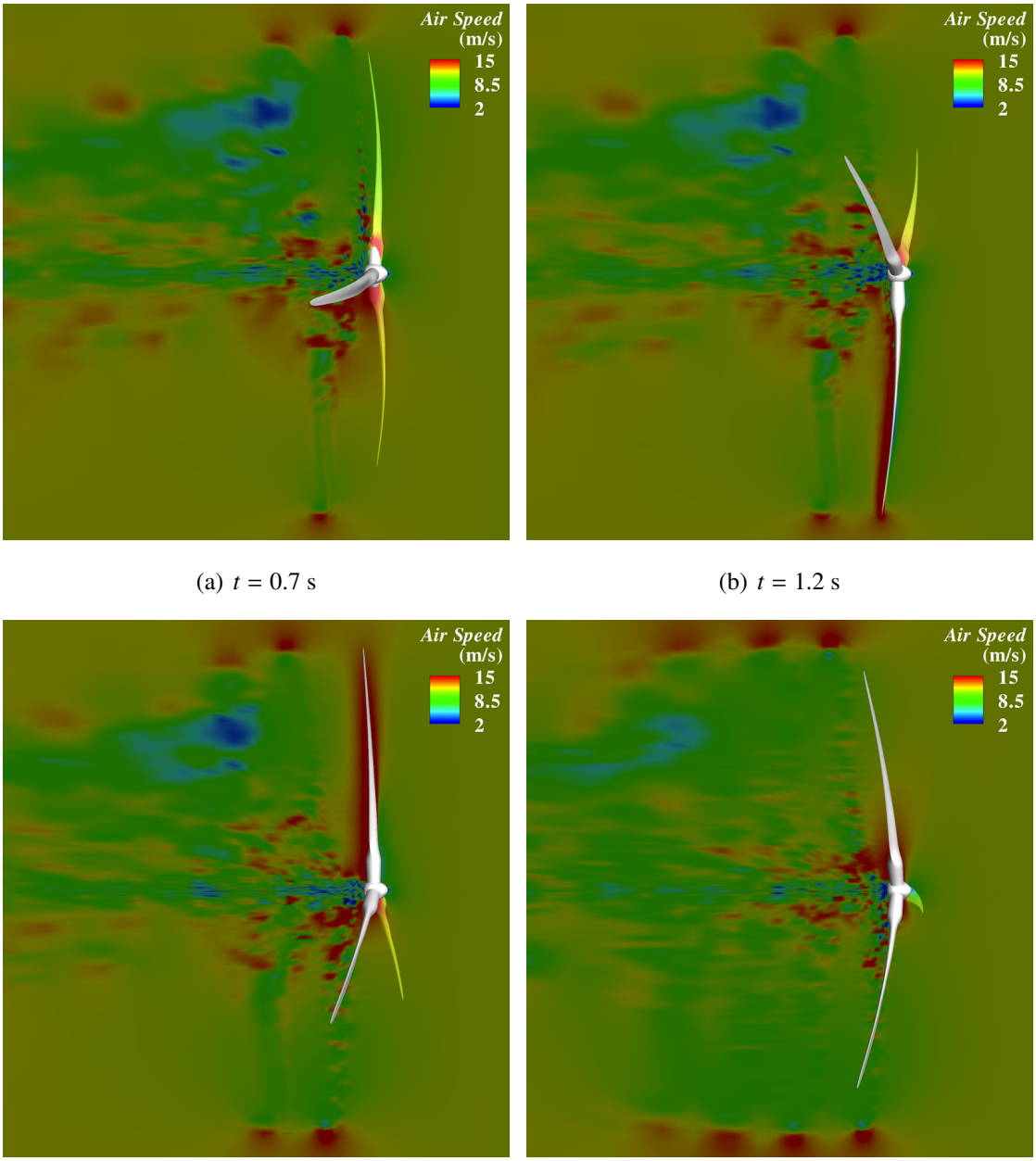
(a) $t = 0.7$ s

(b) $t = 1.2$ s

(c) $t = 2.0$ s

(d) $t = 5.0$ s

Figure 5.11: Isosurfaces of air speed at several instants in the simulation. The flow exhibits complex behavior. The vortical feature generated at the blade tip is convected downstream of the rotor with very little decay.



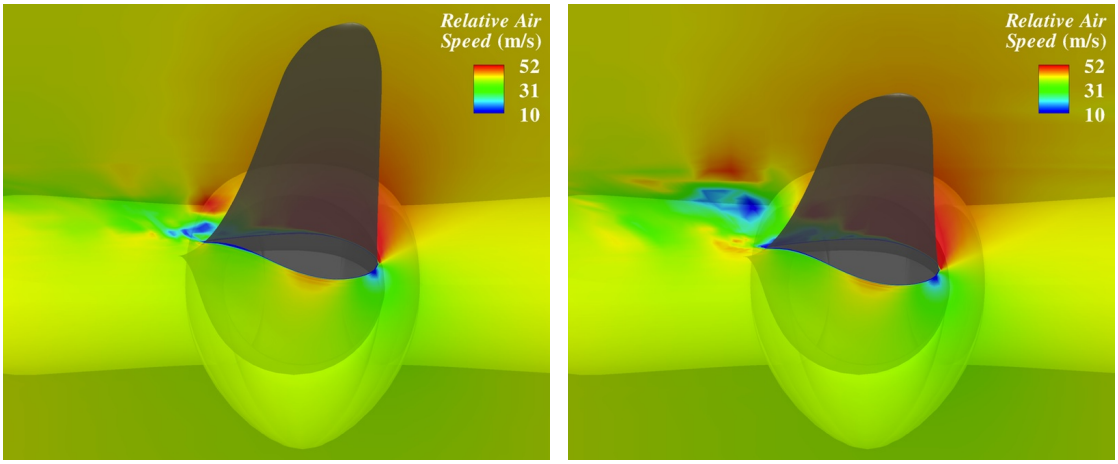
(a) t = 0.7 s

(b) t = 1.2 s

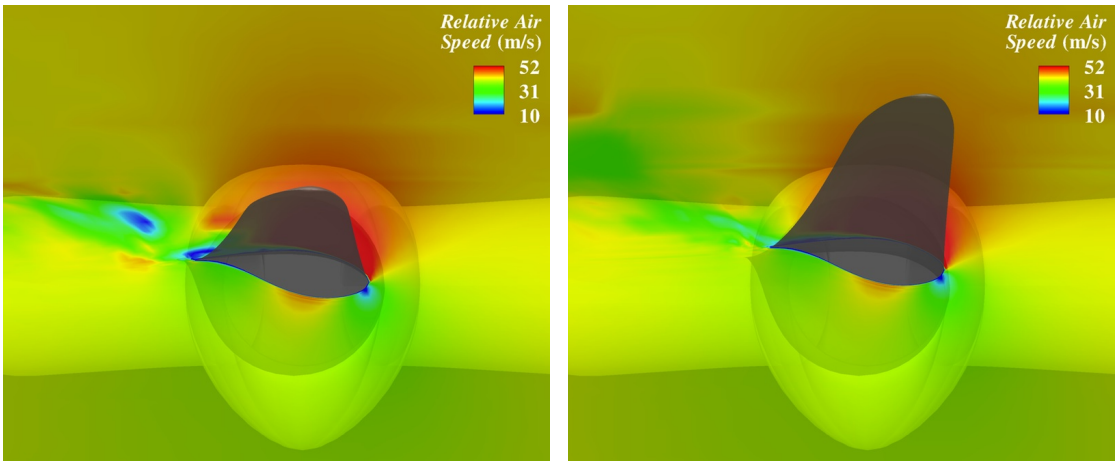
(c) t = 2.0 s

(d) t = 5.0 s

Figure 5.12: Isocontours of air speed at a planar cut superposed with the wind turbine rotor in the deformed configuration. Rotor blade deflection is clearly visible.



(a) t = 0.7 s (b) t = 1.2 s



(c) t = 2.0 s (d) t = 5.0 s

Figure 5.13: Isocontours of relative wind speed at a 30 m radial cut at different time instances superposed on a moving blade. The air flow is fully attached on the pressure side of the blade and separates on the suction side. The flow separation point varies as the blade moves under the action of wind, inertial, and gravitational forces.

high-frequency oscillations, while the rigid blade torque is smooth (see Figure 5.14). To better understand this behavior, we examine the twisting motion of the wind turbine blade about its axis. Figure 5.15 provides a definition of the twist angle for a given blade cross-section. Time histories of the twist angle at four different cross-sections are shown in Figure 5.16. The twist angle increases with distance from the root and reaches almost 2° near the tip in the early stages of the simulation. However, starting at t = 1.2 s, when the blade tip reaches its lowest vertical position, the magnitude of the twist angle is reduced significantly. The reversal of the gravity vector with respect to the lift direction clearly affects the edgewise bending and twisting behavior of the blade. The blade twist angle undergoes high frequency oscillations, which are driven by the trailing-edge vortex shedding and turbulence. Local oscillations of the twist angle lead to the temporal fluctuations in the aerodynamic torque.

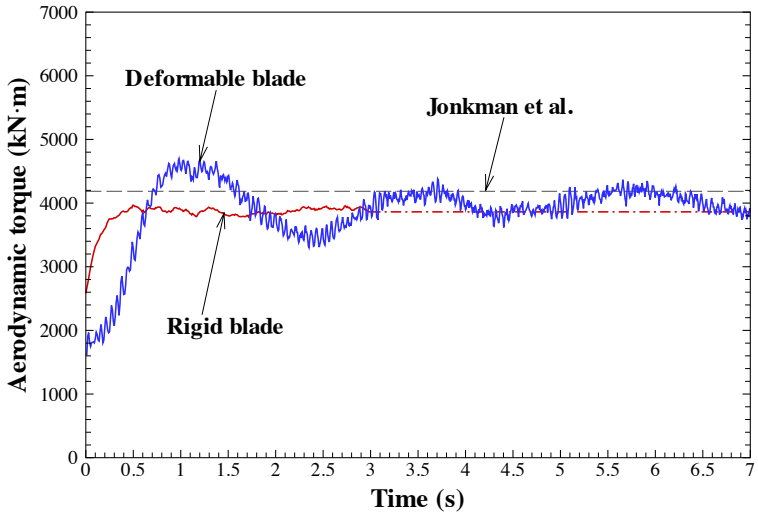


Figure 5.14: Time history of the aerodynamic torque. Both rigid and flexible rotor results are plotted. The reference steady-state result from Jonkman *et al.* [4] is also shown for comparison.

Figure 5.17 shows the blade cross-section twist angle as a function of cross-section distance from the root at different time instances. After the blade passes its lowest point, the distribution of the twist angle changes drastically.

At t = 0.7 s the composite blade experienced the maximum flapwise tip deflection. We found the magnitudes of the stress components (in the basis corresponding to the material axes) for every ply are below the composite strength. The most critical

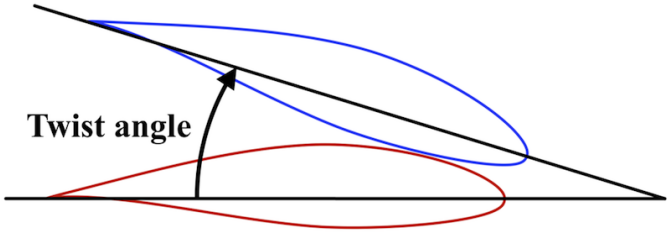


Figure 5.15: Definition of the blade cross-section twist angle.

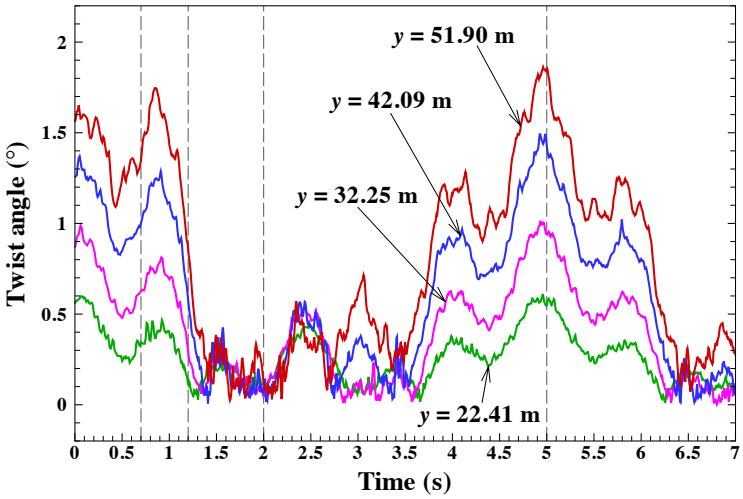


Figure 5.16: Time histories of the twist angle at four cross-sections along the blade axis.

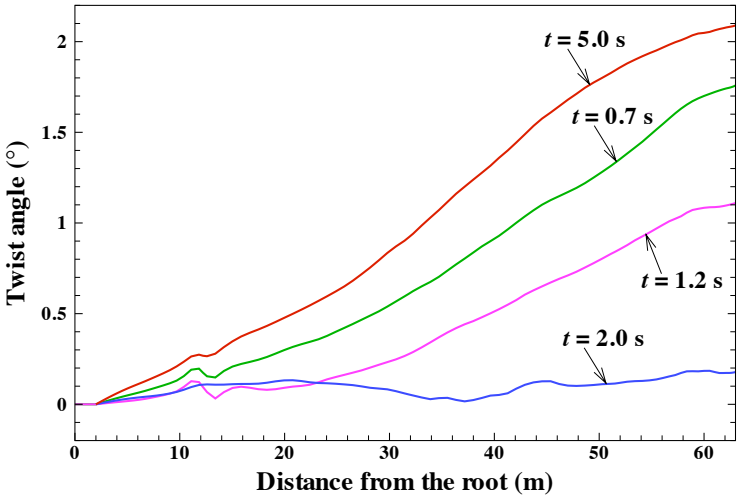
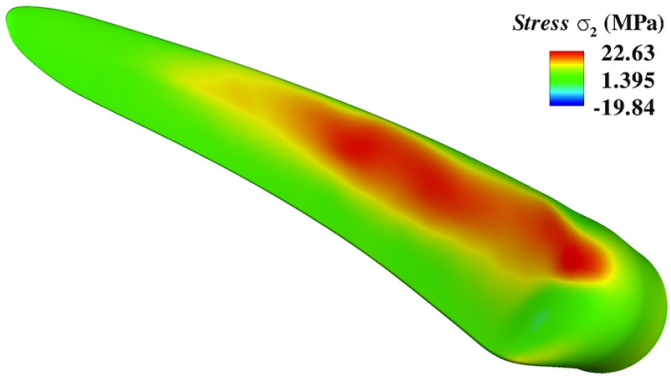
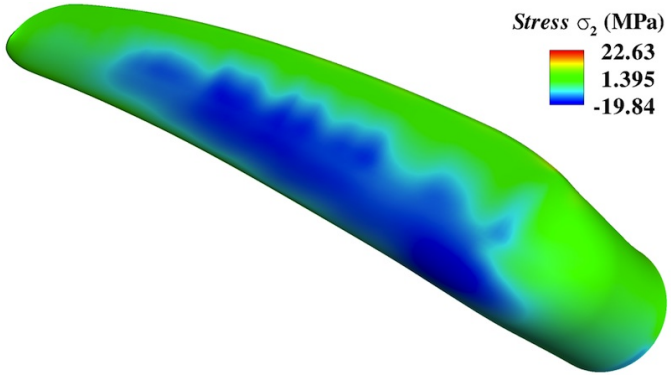


Figure 5.17: Distribution of the cross-section twist angle along the length of the blade at different time instances.

stress component of the entire blade is σ_{22} in ply number 14 (0° fiber orientation). The maximum value of σ_{22} reaches 22.63 MPa, while the corresponding failure strength is 39 MPa [131]. This indicates the proposed blade design can withstand the simulated operating conditions. The isocontours of σ_{22} are plotted in Figure 5.18 and show strong tension on the pressure side and compression on the suction side of the blade.



(a) Pressure side



(b) Suction side

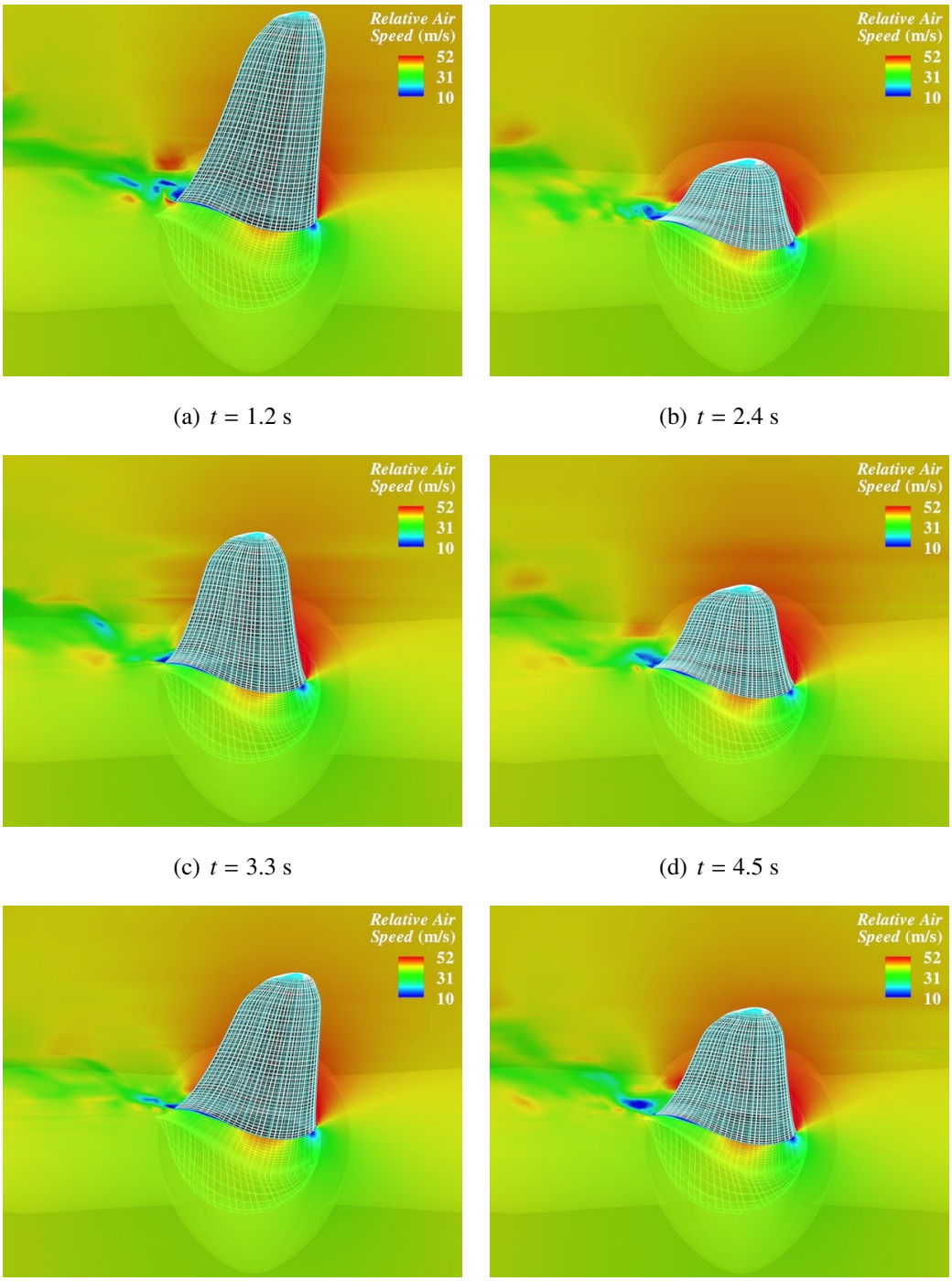
Figure 5.18: Isocontours of stress component σ_{22} (in the direction transverse to the fiber) in the 14^{th} ply (0° fiber orientation) of the composite blade at $t = 0.7$ s view from the pressure and suction side of the blade. Strong tension on the pressure side and compression on the suction side of the blade are found.

Remark 5.11. We note that in the computations presented here the structure is modeled as a shell with a smooth thickness variation. Structural members, such as spar caps and shear webs, which provide additional bending and torsional stiffness for improved blade response, are not considered here and will be added to the blade structural model in the future.

5.4.2 Coupling of NURBS for Fluid and T-Splines for Structural Mechanics

In this section we show the simulation where we couple the NURBS fluid mechanics and T-spline structural mechanics using a non-matching discretization of the fluid–structure interface. We perform an FSI computation for 8 s of real time, by which time the initial transient nearly settles. For the three-bladed 5 MW rotor we perform a single-blade simulation with only one third of the cylindrical computational domain and rotationally periodic boundary conditions. Snapshots of the relative air speed on the blade axis cross-section at different times during the simulation are shown in Figures 5.19 and 5.20. The results are shown on the deformed configuration, which is rotated back to the initial position in order to show the blade deflection. Both the T-spline mesh of the blade and the NURBS mesh of the fluid mechanics domain boundary corresponding to the fluid–structure interface are superposed on the figures. It is clear, especially from Figure 5.20, which zooms on the blade tip and shows how different the NURBS and T-spline meshes are, that the meshes stay "glued" to one another for the entire simulation. In Figure 5.21 we plot the blade tip displacement error between the fluid and structural surface meshes as a function of time. The sole source of the error is the L^2-projection of the kinematic data between the two non-matching meshes. We note that the error magnitude does not exceed 0.03%.

The remainder of this section compares the non-matching mesh simulation with the T-spline structural model of the blade, and the all-NURBS (fluid and structure) simulation using a *matching discretization* of the fluid–structure interface. Figure 5.22 shows the time history of flapwise and edgewise tip displacement. For this quantity, the results of the matching and non-matching mesh simulations are nearly indistinguishable for the entire duration of the simulation. Time history of the aerodynamic torque is shown in Figure 5.23. Both simulations produce the same large-scale response, with some differences in the details of the low-amplitude high-frequency torque oscillations. The time history of the twist angle at two different blade axial locations is shown in Figure 5.24. The blade twists and untwists as it undergoes rotational motion. This is due to the reversal of the gravity vector with respect to the aerodynamic traction vector as the blade tip passes its low and high points. High-frequency twisting oscillations are present, show-



(a) t = 1.2 s

(b) t = 2.4 s

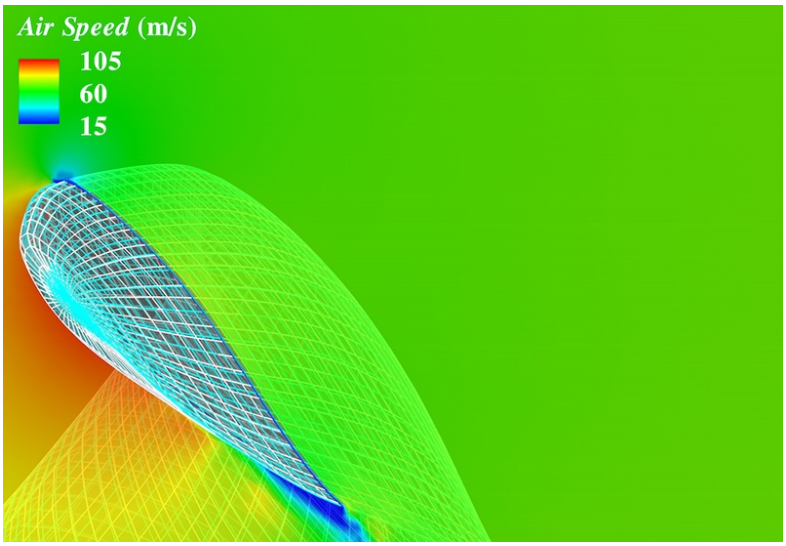
(c) t = 3.3 s

(d) t = 4.5 s

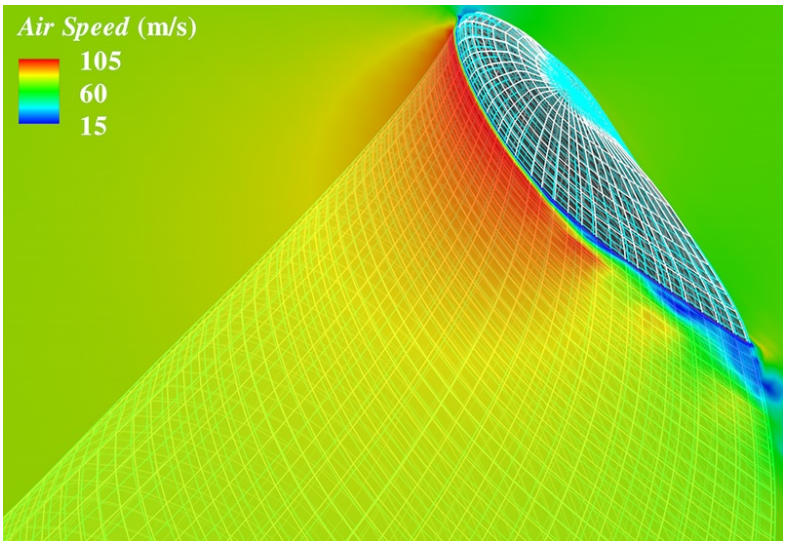
(e) t = 5.7 s

(f) t = 7.8 s

Figure 5.19: Relative air speed in a blade cross-section, rotated to the reference configuration, corresponding to a 30 m radial cut. Note that the blade deflection is quite significant. Both the NURBS mesh of the fluid domain boundary and the T-spline mesh of the blade are shown on the blade surface.



(a)



(b)

Figure 5.20: Relative air speed in a blade cross-section, rotated to the reference configuration. Zoom on the blade tip, showing the superposition of the NURBS mesh of the fluid domain boundary and the T-spline mesh of the blade. The snapshots correspond to the time close to (a) the beginning and (b) the end of the simulation, illustrating that the T-spline and NURBS meshes stay "glued" to one another for the entire duration of the simulation.

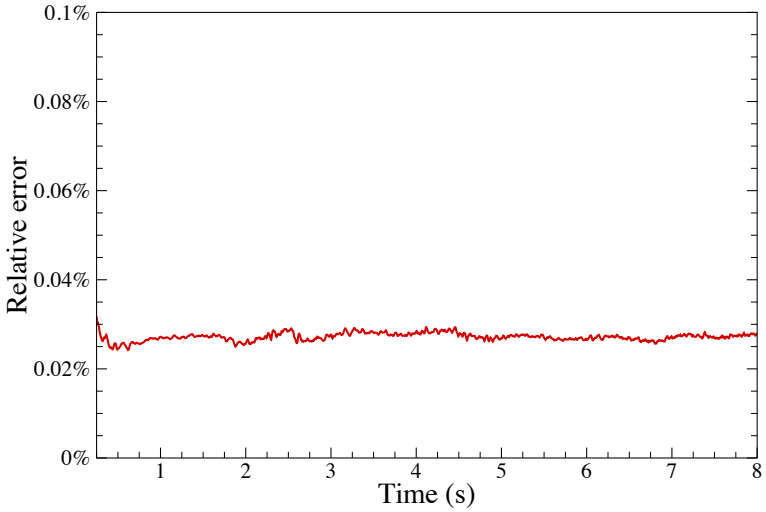


Figure 5.21: Relative error between the T-spline mesh tip displacement and the L^2-projection thereof on the surface mesh of the fluid domain. The relative error is very small and stays nearly constant for the entire simulation. This shows the that structural kinematics is correctly and accurately transferred to the fluid mesh.

ing minor differences between the two simulations, however, the agreement between the matching and non-matching FSI cases is very good.

This example shows that the proposed coupling methodology is such that there is very little or virtually no accuracy loss due to the presence of the non-matching FSI interface discretization.

5.4.3 Coupling of FEM for Fluid and T-Splines for Structural Mechanics

In this section, we show computational results for the FSI coupling of FEM and IGA. Instead of using rotational periodicity, we compute the full domain with three rotor blades. We use an automatic mesh generator to create a linear tetrahedral mesh of the aerodynamics domain, consisting of only 1,193,404 nodes. This is a significantly coarser fluid mechanics mesh than the NURBS mesh used in the previous section. About 1.5 million control points were employed for the discretization of 1/3 of the fluid domain in the case of the quadratics NURBS mesh. The same structural T-spline mesh is employed here as in the previous section.

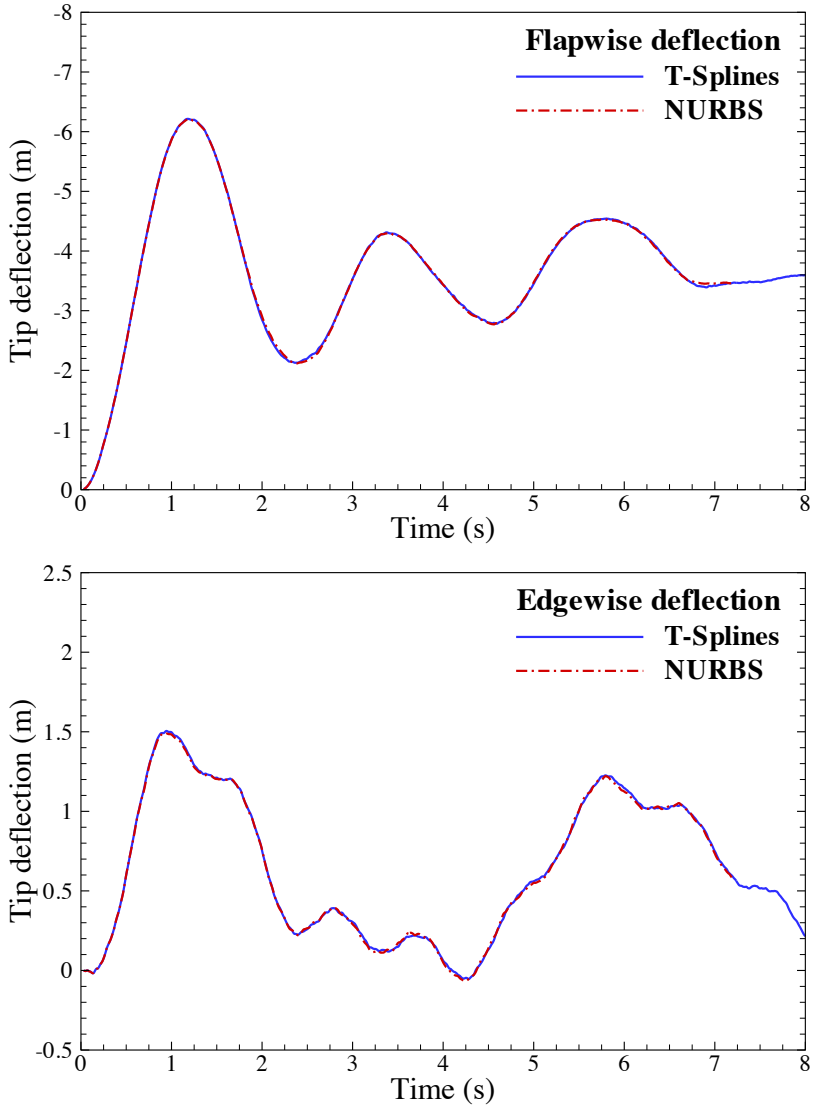


Figure 5.22: Time history of the blade tip displacement. Comparison between the matching and non-matching interface discretization FSI simulations reveals virtually no difference in tip displacement.

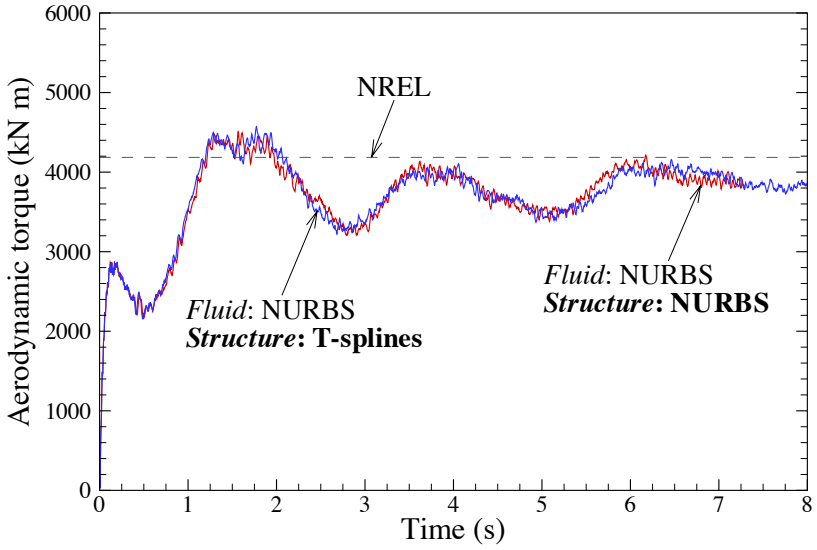


Figure 5.23: Time history of the aerodynamic torque. Comparison between the matching and non-matching interface discretization FSI simulations reveals no difference in large-scale response, while minor differences are present in the small-scale oscillations. Both simulations compare favorably to the NREL data [4].

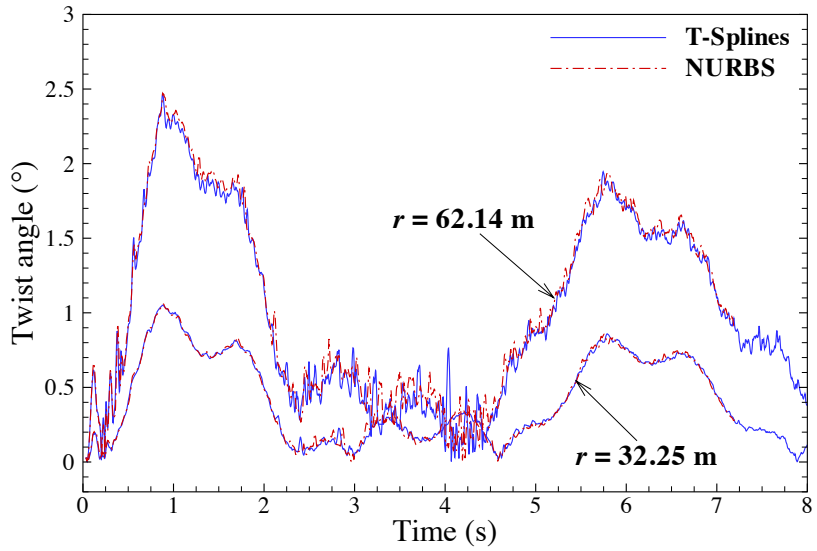


Figure 5.24: Time history of the twist angle at the two cross-sections on the blade axis. Comparison between the matching and non-matching interface discretization FSI simulations reveals no difference in large-scale response, while minor differences are present in the small-scale oscillations.

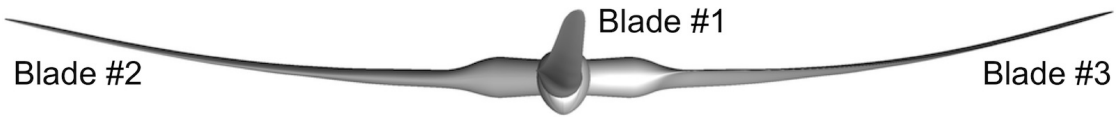


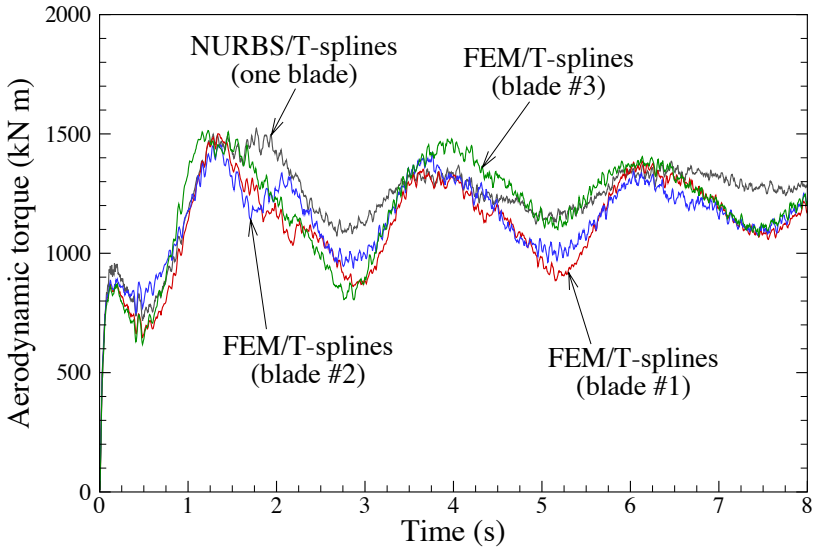
Figure 5.25: Three-bladed rotor in a deformed configuration at a time instant during the FSI simulation.

Figure 5.25 shows the three-bladed rotor at one time instant during the FSI simulation. Note that some differences in the flapwise deflection are present from one blade to another. Figure 5.26 shows the time history of the aerodynamic torque and the flapwise deflection of the rotor blades. These quantities are reported for each blade individually, and the results are compared with a NURBS/T-spline simulation from the previous section. While there is a very reasonable overall agreement, the FEM/T-spline simulation predicts larger peak-to-peak variations in both quantities during the initial transient response of the coupled system.

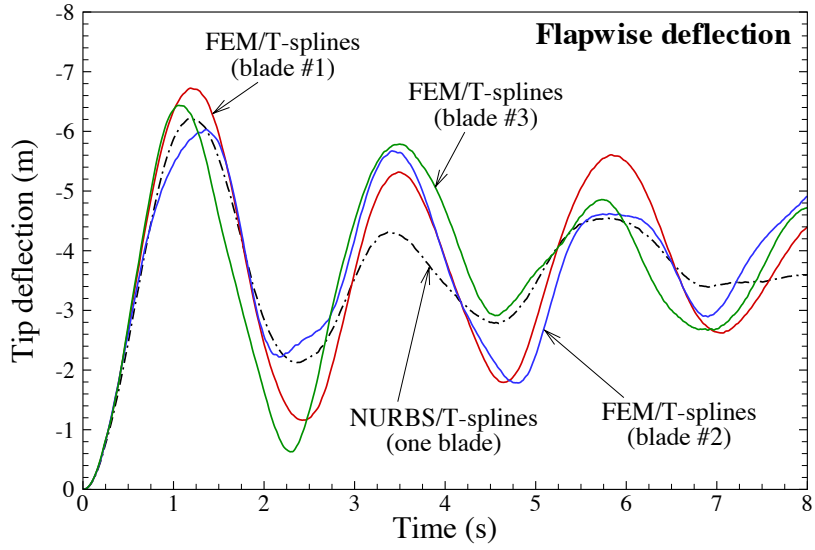
Some of the discrepancies in the results come from the differences in the starting configuration for each blade, which directly affects the structural dynamics of the rotor. Other differences come from the prediction of the aerodynamics phenomena. Figure 5.27 compares relative air speed and pressure distribution at a fixed time on a cross-section corresponding to 80% of the blade radius. We notice that the significantly finer NURBS mesh is capable of resolving some of the trailing edge turbulence, while the coarse FEM mesh produces a "smoother" solution. Nevertheless, despite the large differences in mesh resolution and the lack of real boundary-layer meshing in the case of linear FEM, the large-scale features of the flow are qualitatively and quantitatively very similar for both discretizations.

5.5 Acknowledgements

Chapter 5, in part, is a reprint of the material as it appears in: "ALE–VMS and ST–VMS methods for computer modeling of wind-turbine rotor aerodynamics and fluid–structure interaction," (Y. Bazilevs, K. Takizawa and T.E. Tezduyar), *Mathematical Models and Methods in Applied Sciences*, 2012; "Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with application to

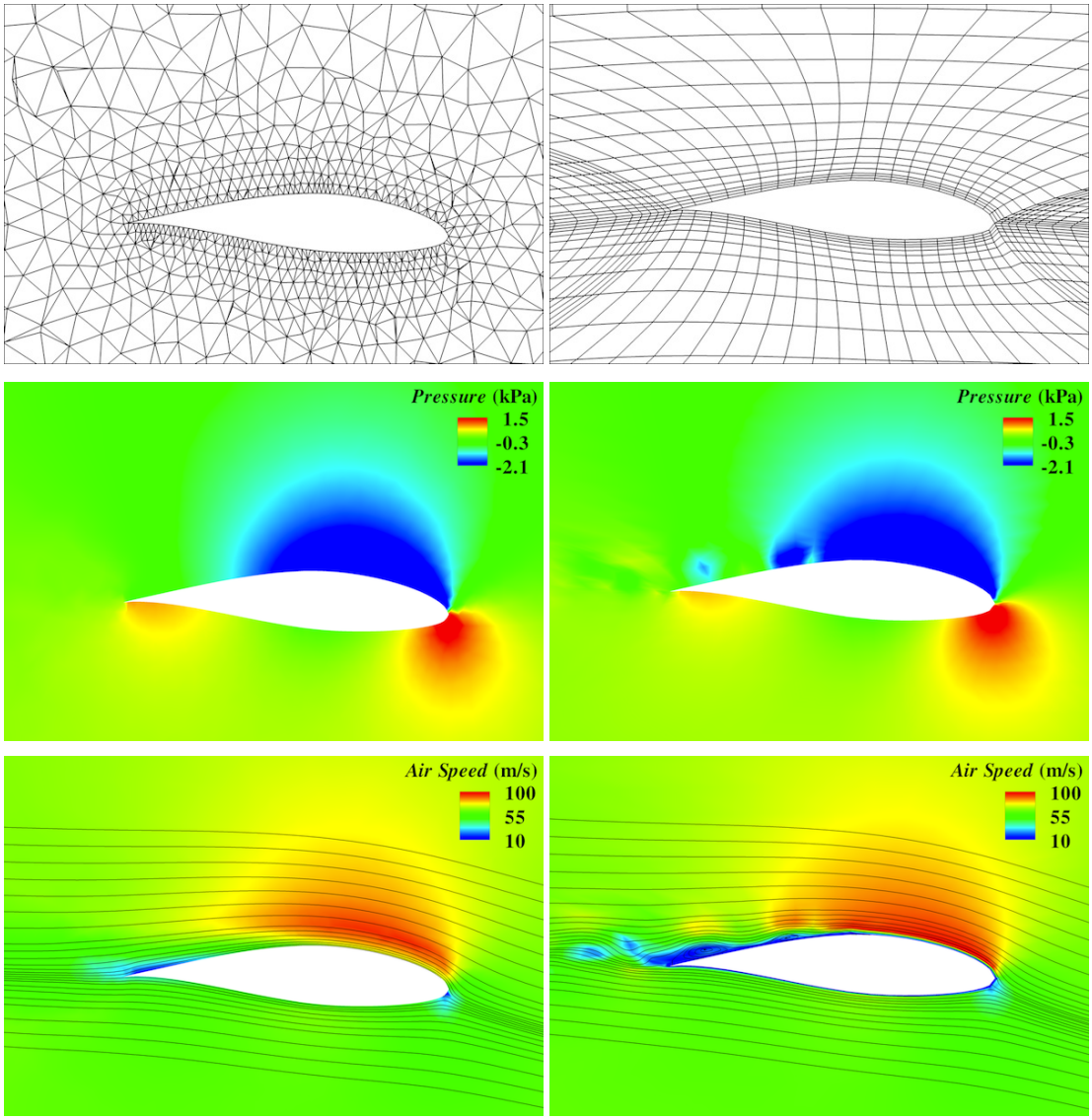


(a) Aerodynamic torque



(b) Flapwise deflection

Figure 5.26: Time history of the aerodynamic torque and flapwise deflection. Comparison between the FEM/T-spline and NURBS/T-spline FSI simulations. The data is presented for each blade individually in the case of FEM/T-spline simulation. The numbering of the blades is consistent with that in Figure 5.25.



(a) FEM/T-spline (b) NURBS/T-spline

Figure 5.27: Comparison of aerodynamic results between FEM/T-spline and NURBS/T-spline FSI simulations. Mesh (top), pressure (middle), and relative air speed (bottom) at a fixed time $t = 7.5$ s are plotted on a cross-section corresponding to 80% of the blade radius.

wind turbines," (Y. Bazilevs and M.A. Scott), *Computer Methods in Applied Mechanics and Engineering*, 2012; "3D simulation of wind turbine rotors at full scale. Part II: Fluid–structure interaction modeling with composite blades," (Y. Bazilevs, J. Kiendl, R. Wüchner and K.-U. Bletzinger), *International Journal for Numerical Methods in Fluids*, 2011. The dissertation author was the primary investigator and author of these papers.

Chapter 6

Conclusions

In this dissertation, we developed the computational procedures, which include both aerodynamics and FSI techniques, for the simulation of wind turbines at full scale. Isogeometric analysis was adopted as the primary geometry modeling and simulation framework. A template-based approach was developed in which the analysis-suitable geometry for both fluid and structural wind turbine rotor domains is constructed. We applied the developed framework to the simulation of the NREL Phase VI wind turbine and the NREL 5 MW offshore baseline wind turbine rotor. The methodology was validated using the NREL Phase VI wind turbine for which there is an extensive set of experimental results. We found that the combination of ALE–VMS formulation of the Navier–Stokes equations of incompressible flows and the weakly enforced essential boundary conditions was able to accurately predict the key quantities of engineering interest such as the aerodynamic torque and pressure distribution at the blade cross-sections. Furthermore, no "tuning" of the turbulence model was necessary as the same multiscale formulation was able to correctly represent both the fully attached and detached turbulent boundary layers.

The wind turbine blade structure is governed by the isogeometric Kirchhoff–Love shell formulation complemented with the bending strip method. This rotation-free formulation is appropriate for thin-shell structures comprised of multiple C^1- or higher-order continuous surface patches that are joined or merged with continuity no greater than C^0. The structural formulation includes numerical treatment and modeling of composite materials, which are used in the manufacturing of modern wind turbine blades.

In the case of FSI simulations, The fluid and structure are allowed to be non-matching at the interface. The augmented Lagrangian framework with a formal elimination of the Lagrange multiplier generates a family of FSI formulations at the continuum level, which serves as a point of departure for the discrete FSI formulation proposed in this work. The augmented Lagrangian approach naturally motivates weak enforcement of essential boundary conditions for the fluid mechanics problem, explains the structure of the penalty terms that stabilize the formulation, and leads to the definition of the fluid traction vector acting on the structural surface that consistently accounts for slip velocity. We also proposed a robust procedure for transferring kinematic and traction data between the non-matching IGA and FEM discretizations.

The coupled system was solved in a block-iterative fashion, which is robust enough for the present application due to the relatively high structural mass of the wind turbine blades. For wind turbine rotors the structural motion is dominated by rotation about the horizontal axis. For this we found it advantageous for overall accuracy of the computations to separate the structural displacement into rotation and deflection parts. With this decomposition, we modified the time integration formulas to treat the rotation part of the structural motion exactly. The decomposition of the structural displacement also yields a more effective mesh-moving computation, where only the deflection part of the fluid-mesh motion is computed with the elasticity-based mesh-moving method. The rotational part of the mesh motion is calculated exactly.

We successfully applied the proposed wind turbine FSI framework to the simulation of the NREL 5 MW offshore baseline wind turbine rotor at full scale. The rotor blades were modeled as symmetric composite laminates homogenized in the through-thickness direction. The computation yielded good prediction of the aerodynamic torque and blade-tip deflection. The computational results also suggest that virtually no accuracy is lost due to the use of non-matching discretizations. Simulations coupling low-order FEM for the fluid and IGA for the structure are likewise fairly successful.

The blade–tower interaction was handled using a sliding interface technique. In the case of NREL Phase VI wind turbine, the influence of the tower was correctly captured in the aerodynamic loads, where appreciable differences between the full-wind-turbine and rotor-only simulations were observed. The simulation results for the blade–

tower interaction cases are also in very good agreement with the experimental data. We feel that the rotor-only computations are reasonably accurate for predicting the mean aerodynamic performance of the wind turbine rotors. However, the presence of the tower has an appreciable effect for the dynamic blade loading, which may be important when considering structural analysis of wind turbine blades, or coupled FSI. The tower effects may be even more pronounced for large rotors, which are used in offshore environments, as well as in the cases of low blade–tower clearance and downwind rotor configurations. We plan to look at some of these cases in the future.

Bibliography

[1] C. A. Walford. Wind turbine reliability: Understanding and minimizing wind turbine O & M costs. Technical Report SAND2006-1100, Sandia National Laboratories, 2006.

[2] E. Echavarria, B. Hahn, G. J. W. van Bussel, and T. Tomiyama. Reliability of wind turbine technology through time. *Journal of Solar Energy Engineering*, 130:031005–1–8, 2008.

[3] J. M. Jonkman and M. L. Buhl Jr. FAST user's guide. Technical Report NREL/EL-500-38230, National Renewable Energy Laboratory, Golden, CO, 2005.

[4] J. Jonkman, S. Butterfield, W. Musial, and G. Scott. Definition of a 5-MW reference wind turbine for offshore system development. Technical Report NREL/TP-500-38060, National Renewable Energy Laboratory, Golden, CO, 2009.

[5] N. N. Sørensen, J. A. Michelsen, and S. Schreck. Navier–Stokes predictions of the NREL Phase VI rotor in the NASA Ames 80 ft × 120 ft wind tunnel. *Wind Energy*, 5:151–169, 2002.

[6] E. P. N. Duque, M. D. Burklund, and W. Johnson. Navier–Stokes and comprehensive analysis performance predictions of the NREL Phase VI experiment. *Journal of Solar Energy Engineering*, 125:457–467, 2003.

[7] A. Le Pape and J. Lecanu. 3D Navier–Stokes computations of a stall-regulated wind turbine. *Wind Energy*, 7:309–324, 2004.

[8] D. D. Chao and C. P. van Dam. Computational aerodynamic analysis of a blunt trailing-edge airfoil modification to the NREL Phase VI rotor. *Wind Energy*, 10:529–550, 2007.

[9] S. Gómez-Iradi, R. Steijl, and G. N. Barakos. Development and validation of a CFD technique for the aerodynamic analysis of HAWT. *Journal of Solar Energy Engineering*, 131:031009–1–13, 2009.

[10] F. Zahle, N. N. Sørensen, and J. Johansen. Wind turbine rotor-tower interaction using an incompressible overset grid method. *Wind Energy*, 12:594–619, 2009.

[11] A. Bechmann, N. N. Sørensen, and F. Zahle. CFD simulations of the MEXICO rotor. *Wind Energy*, 14:677–689, 2011.

[12] Y. Bazilevs, M.-C. Hsu, I. Akkerman, S. Wright, K. Takizawa, B. Henicke, T. Spielman, and T. E. Tezduyar. 3D simulation of wind turbine rotors at full scale. Part I: Geometry modeling and aerodynamics. *International Journal for Numerical Methods in Fluids*, 65:207–235, 2011.

[13] K. Takizawa, B. Henicke, T. E. Tezduyar, M.-C. Hsu, and Y. Bazilevs. Stabilized space–time computation of wind-turbine rotor aerodynamics. *Computational Mechanics*, 48:333–344, 2011.

[14] K. Takizawa, B. Henicke, D. Montes, T. E. Tezduyar, M.-C. Hsu, and Y. Bazilevs. Numerical-performance studies for the stabilized space–time computation of wind-turbine rotor aerodynamics. *Computational Mechanics*, 48:647–657, 2011.

[15] Y. Li and P. M. Carrica K. J. Paik, T. Xing. Dynamic overset CFD simulations of wind turbine aerodynamics. *Renewable Energy*, 37:285–298, 2012.

[16] N. N. Sørensen and S. Schreck. Computation of the National Renewable Energy Laboratory Phase-VI rotor in pitch motion during standstill. *Wind Energy*, 2012. doi:10.1002/we.480.

[17] F. Scheurich and R. E. Brown. Modelling the aerodynamics of vertical-axis wind turbines in unsteady wind conditions. *Wind Energy*, 2012. doi:10.1002/we.532.

[18] R. Chow and C. P. van Dam. Verification of computational simulations of the NREL 5 MW rotor with a focus on inboard flow separation. *Wind Energy*, 2012. doi:10.1002/we.529.

[19] P. Stein and K. Beucke M.-C. Hsu, Y. Bazilevs. Operator- and template-based modeling of solid geometry for isogeometric analysis with application to vertical axis wind turbine simulation. *Computer Methods in Applied Mechanics and Engineering*, 213–216:71–83, 2012.

[20] M.-C. Hsu, I. Akkerman, and Y. Bazilevs. Wind turbine aerodynamics using ALE–VMS: Validation and the role of weakly enforced boundary conditions. *Computational Mechanics*, 2012. doi:10.1007/s00466-012-0686-x.

[21] C. Kong, J. Bang, and Y. Sugiyama. Structural investigation of composite wind turbine blade considering various load cases and fatigue life. *Energy*, 30:2101–2114, 2005.

[22] E. Lund and J. Stegmann. On structural optimization of composite shell structures using a discrete constitutive parametrization. *Wind Energy*, 8:109–124, 2005.

[23] M. O. L. Hansen, J. N. Sørensen, S. Voutsinas andN. Sørensen, and H. Aa. Madsen. State of the art in wind turbine aerodynamics and aeroelasticity. *Progress in Aerospace Sciences*, 42:285–330, 2006.

[24] F. M. Jensen, B. G. Falzon, J. Ankersen, and H. Stang. Structural testing and numerical simulation of a 34 m composite wind turbine blade. *Composite Structures*, 76:52–61, 2006.

[25] Y. Bazilevs, M.-C. Hsu, J. Kiendl, and D. J. Benson. A computational procedure for prebending of wind turbine blades. *International Journal for Numerical Methods in Engineering*, 89:323–336, 2012.

[26] D. Cárdenas, A. A. Escárpita, H. Elizalde, J. J. Aguirre, P. Marzocca H. Ahuett, and O. Probst. Numerical validation of a finite element thin-walled beam model of a composite wind turbine blade. *Wind Energy*, 15:203–223, 2012.

[27] T. J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, Mineola, NY, 2000.

[28] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.

[29] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley, Chichester, 2009.

[30] Y. Bazilevs, V. M. Calo, J. A. Cottrel, T. J. R. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 197:173–201, 2007.

[31] J. A. Cottrell, A. Reali, Y. Bazilevs, and T. J. R. Hughes. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 195:5257–5297, 2006.

[32] Y. Bazilevs, L. Beirao da Veiga, J. A. Cottrell, T. J. R. Hughes, and G. Sangalli. Isogeometric analysis: Approximation, stability and error estimates for *h*-refined meshes. *Mathematical Models and Methods in Applied Sciences*, 16:1031–1090, 2006.

[33] J. A. Cottrell, T. J. R. Hughes, and A. Reali. Studies of refinement and continuity in isogeometric structural analysis. *Computer Methods in Applied Mechanics and Engineering*, 196:4160–4183, 2007.

[34] W. A. Wall, M. A. Frenzel, and C. Cyron. Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 197:2976–2988, 2008.

[35] J. A. Evans, Y. Bazilevs, I. Babuška, and T. J. R. Hughes. n-Widths, sup-infs, and optimality ratios for the k-version of the isogeometric finite element method. *Computer Methods in Applied Mechanics and Engineering*, 198:1726–1741, 2009.

[36] M. R. Dörfel, B. Jüttler, and B. Simeon. Adaptive isogeometric analysis by local h-refinement with T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199:264–275, 2010.

[37] Y. Bazilevs, V. M. Calo, J. A. Cottrell, J. A. Evans, T. J. R. Hughes, S. Lipton, M. A. Scott, and T. W. Sederberg. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199:229–263, 2010.

[38] F. Auricchio, L. Beirão da Veiga, C. Lovadina, and A. Reali. The importance of the exact satisfaction of the incompressibility constraint in nonlinear elasticity: Mixed FEMs versus NURBS-based approximations. *Computer Methods in Applied Mechanics and Engineering*, 199:314–323, 2010.

[39] W. Wang and Y. Zhang. Wavelets-based NURBS simplification and fairing. *Computer Methods in Applied Mechanics and Engineering*, 199:290–300, 2010.

[40] E. Cohen, T. Martin, R. M. Kirby, T. Lyche, and R. F. Riesenfeld. Analysis-aware modeling: Understanding quality considerations in modeling for isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 199:334–356, 2010.

[41] V. Srinivasan, S. Radhakrishnan, and G. Subbarayan. Coordinated synthesis of hierarchical engineering systems. *Computer Methods in Applied Mechanics and Engineering*, 199:392–404, 2010.

[42] Y. Bazilevs, C. Michler, V. M. Calo, and T. J. R. Hughes. Weak Dirichlet boundary conditions for wall-bounded turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 196:4853–4862, 2007.

[43] Y. Bazilevs, C. Michler, V. M. Calo, and T. J. R. Hughes. Isogeometric variational multiscale modeling of wall-bounded turbulent flows with weakly enforced boundary conditions on unstretched meshes. *Computer Methods in Applied Mechanics and Engineering*, 199:780–790, 2010.

[44] I. Akkerman, Y. Bazilevs, V. M. Calo, T. J. R. Hughes, and S. Hulshoff. The role of continuity in residual-based variational multiscale modeling of turbulence. *Computational Mechanics*, 41:371–378, 2008.

[45] M.-C. Hsu, Y. Bazilevs, V. M. Calo, T. E. Tezduyar, and T. J. R. Hughes. Improving stability of stabilized and multiscale formulations in flow simulations at small time steps. *Computer Methods in Applied Mechanics and Engineering*, 199:828–840, 2010.

[46] Y. Bazilevs and I. Akkerman. Large eddy simulation of turbulent Taylor–Couette flow using isogeometric analysis and the residual–based variational multiscale method. *Journal of Computational Physics*, 229:3402–3414, 2010.

[47] T. Elguedj, Y. Bazilevs, V. M. Calo, and T. J. R. Hughes. $\overline{\mathrm{B}}$ and $\overline{\mathrm{F}}$ projection methods for nearly incompressible linear and non-linear elasticity and plasticity using higher-order NURBS elements. *Computer Methods in Applied Mechanics and Engineering*, 197:2732–2762, 2008.

[48] S. Lipton, J. A. Evans, Y. Bazilevs, T. Elguedj, and T. J. R. Hughes. Robustness of isogeometric structural discretizations under severe mesh distortion. *Computer Methods in Applied Mechanics and Engineering*, 199:357–373, 2010.

[49] D. J. Benson, Y. Bazilevs, E. De Luycker, M.-C. Hsu, M. Scott, T. J. R. Hughes, and T. Belytschko. A generalized finite element formulation for arbitrary basis functions: from isogeometric analysis to XFEM. *International Journal for Numerical Methods in Engineering*, 83:765–785, 2010.

[50] D. J. Benson, Y. Bazilevs, M.-C. Hsu, and T. J. R. Hughes. Isogeometric shell analysis: The Reissner–Mindlin shell. *Computer Methods in Applied Mechanics and Engineering*, 199:276–289, 2010.

[51] D. J. Benson, Y. Bazilevs, M.-C. Hsu, and T. J. R. Hughes. A large deformation, rotation-free, isogeometric shell. *Computer Methods in Applied Mechanics and Engineering*, 200:1367–1378, 2011.

[52] J. Kiendl, K.-U. Bletzinger, J. Linhard, and R. Wüchner. Isogeometric shell analysis with Kirchhoff–Love elements. *Computer Methods in Applied Mechanics and Engineering*, 198:3902–3914, 2009.

[53] J. Kiendl, Y. Bazilevs, M.-C. Hsu, R. Wüchner, and K.-U. Bletzinger. The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches. *Computer Methods in Applied Mechanics and Engineering*, 199:2403–2416, 2010.

[54] E. De Luycker, D. J. Benson, T. Belytschko, Y. Bazilevs, and M.-C. Hsu. X-FEM in isogeometric analysis for linear fracture mechanics. *International Journal for Numerical Methods in Engineering*, 87:541–565, 2011.

[55] Y. Zhang, Y. Bazilevs, S. Goswami, C. Bajaj, and T. J. R. Hughes. Patient-specific vascular nurbs modeling for isogeometric analysis of blood flow. *Computer Methods in Applied Mechanics and Engineering*, 196:2943–2959, 2007.

[56] Y. Bazilevs, V. M. Calo, Y. Zhang, and T. J. R. Hughes. Isogeometric fluid–structure interaction analysis with applications to arterial blood flow. *Computational Mechanics*, 38:310–322, 2006.

[57] Y. Bazilevs, V. M. Calo, T. J. R. Hughes, and Y. Zhang. Isogeometric fluid–structure interaction: theory, algorithms, and computations. *Computational Mechanics*, 43:3–37, 2008.

[58] J. G. Isaksen, Y. Bazilevs, T. Kvamsdal, Y. Zhang, J. H. Kaspersen, K. Waterloo, B. Romner, and T. Ingebrigtsen. Determination of wall tension in cerebral artery aneurysms by numerical simulation. *Stroke*, 39:3172–3178, 2008.

[59] Y. Bazilevs, M.-C. Hsu, J. Kiendl, R. Wüchner, and K.-U. Bletzinger. 3D simulation of wind turbine rotors at full scale. Part II: Fluid–structure interaction modeling with composite blades. *International Journal for Numerical Methods in Fluids*, 65:236–253, 2011.

[60] Y. Bazilevs, M.-C. Hsu, and M. A. Scott. Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines. *Computer Methods in Applied Mechanics and Engineering*, 2012. doi:10.1016/j.cma.2012.03.028.

[61] Y. Bazilevs and T. J. R. Hughes. NURBS-based isogeometric analysis for the computation of flows about rotating components. *Computational Mechanics*, 43:143–150, 2008.

[62] T. J. R. Hughes, W. K. Liu, and T. K. Zimmermann. Lagrangian–Eulerian finite element formulation for incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 29:329–349, 1981.

[63] T. J. R. Hughes. Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles, and the origins of stabilized methods. *Computer Methods in Applied Mechanics and Engineering*, 127:387–401, 1995.

[64] T. J. R. Hughes, A. A. Oberai, and L. Mazzei. Large eddy simulation of turbulent channel flows by the variational multiscale method. *Physics of Fluids*, 13:1784–1799, 2001.

[65] T. J. R. Hughes and G. Sangalli. Variational multiscale analysis: the fine-scale Green's function, projection, optimization, localization, and stabilized methods. *SIAM Journal of Numerical Analysis*, 45:539–557, 2007.

[66] Y. Bazilevs and T. J. R. Hughes. Weak imposition of Dirichlet boundary conditions in fluid mechanics. *Computers and Fluids*, 36:12–26, 2007.

[67] J. Nitsche. Uber ein variationsprinzip zur losung von Dirichlet-problemen bei verwendung von teilraumen, die keinen randbedingungen unterworfen sind. *Abh. Math. Univ. Hamburg*, 36:9–15, 1971.

[68] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of Discontinuous Galerkin methods for elliptic problems. *SIAM Journal of Numerical Analysis*, 39:1749–1779, 2002.

[69] M. Bischoff, W. A. Wall, K. -U. Bletzinger, and E. Ramm. Models and finite elements for thin-walled structures. In E. Stein, R. de Borst, and T. J. R. Hughes, editors, *Encyclopedia of Computational Mechanics, Vol. 2, Solids, Structures and Coupled Problems*, chapter 3. Wiley, 2004.

[70] L. Piegl and W. Tiller. *The NURBS Book (Monographs in Visual Communication), 2nd ed.* Springer-Verlag, New York, 1997.

[71] F. Cirak, M. Ortiz, and P. Schröder. Subdivision surfaces: a new paradigm for thin shell analysis. *International Journal for Numerical Methods in Engineering*, 47:2039–2072, 2000.

[72] F. Cirak and M. Ortiz. Fully C^1-conforming subdivision elements for finite deformation thin shell analysis. *International Journal for Numerical Methods in Engineering*, 51:813–833, 2001.

[73] F. Cirak, M. J. Scott, E. K. Antonsson, M. Ortiz, and P. Schröder. Integrated modeling, finite-element analysis, and engineering design for thin-shell structures using subdivision. *Computer-Aided Design*, 34:137–148, 2002.

[74] W. Wang, Y. Zhang, M. A. Scott, and T. J. R. Hughes. Converting an unstructured quadrilateral mesh to a standard T-spline surface. *Computational Mechanics*, 48:477–498, 2011.

[75] W. Wang, Y. Zhang, G. Xu, and T.J.R. Hughes. Converting an unstructured quadrilateral/hexahedral mesh to a rational T-spline. *Computational Mechanics*, 50:65–84, 2012.

[76] G. E. Farin. *NURBS Curves and Surfaces: From Projective Geometry to Practical Use*. A. K. Peters, Ltd., Natick, MA, 1995.

[77] D. F. Rogers. *An Introduction to NURBS With Historical Perspective*. Academic Press, San Diego, CA, 2001.

[78] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and T-NURCCS. *ACM Transactions on Graphics*, 22(3):477–484, 2003.

[79] T. W. Sederberg, D.L. Cardon, G.T. Finnigan, N.S. North, J. Zheng, and T. Lyche. T-spline simplification and local refinement. *ACM Transactions on Graphics*, 23(3):276–283, 2004.

[80] X. Li, J. Zheng, T. W. Sederberg, T. J. R. Hughes, and M. A. Scott. On linear independence of t-spline blending functions. *Computer-Aided Geometric Design*, 29:63–76, 2012.

[81] M. A. Scott, X. Li, T. W. Sederberg, and T. J. R. Hughes. Local refinement of analysis-suitable T-splines. *Computer Methods in Applied Mechanics and Engineering*, 213–216:206–222, 2012.

[82] M. J. Borden, M. A. Scott, J. A. Evans, and T. J. R. Hughes. Isogeometric finite element data structures based on Bézier extraction of NURBS. *International Journal for Numerical Methods in Engineering*, 87:15–47, 2011.

[83] M. A. Scott, M. J. Borden, C. V. Verhoosel, T. W. Sederberg, and T. J. R. Hughes. Isogeometric finite element data structures based on Bézier extraction of T-splines. *International Journal for Numerical Methods in Engineering*, 88:126–156, 2011.

[84] H. Gomez, V. M. Calo, Y. Bazilevs, and T. J. R. Hughes. Isogeometric analysis of the Cahn–Hilliard phase-field model. *Computer Methods in Applied Mechanics and Engineering*, 197:4333–4352, 2008.

[85] H. Gomez, T. J. R. Hughes, X. Nogueira, and V. M. Calo. Isogeometric analysis of the isothermal Navier–Stokes–Korteweg equations. *Computer Methods in Applied Mechanics and Engineering*, 199:1828–1840, 2010.

[86] H. J. T. Kooijman, C. Lindenburg, and D. Winkelaarand E. L. van der Hooft. DOWEC 6 MW pre-design: Aero-elastic modelling ofthe DOWEC 6 MW pre-design in PHATAS. Technical Report DOWEC-F1W2-HJK-01-046/9, 2003.

[87] C. Johnson. *Numerical solution of partial differential equations by the finite element method*. Cambridge University Press, Sweden, 1987.

[88] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods, 2nd ed.* Springer, 2002.

[89] A. Ern and J. L. Guermond. *Theory and Practice of Finite Elements*. Springer, 2004.

[90] A. N. Brooks and T. J. R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32:199–259, 1982.

[91] T. J. R. Hughes and T. E. Tezduyar. Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations. *Computer Methods in Applied Mechanics and Engineering*, 45:217–284, 1984.

[92] T. E. Tezduyar and Y. J. Park. Discontinuity capturing finite element formulations for nonlinear convection-diffusion-reaction equations. *Computer Methods in Applied Mechanics and Engineering*, 59:307–325, 1986.

[93] T. J. R. Hughes, L. P. Franca, and M. Balestra. A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuška–Brezzi condition: A stable Petrov–Galerkin formulation of the Stokes problem accommodating equal-order interpolations. *Computer Methods in Applied Mechanics and Engineering*, 59:85–99, 1986.

[94] T. E. Tezduyar and Y. Osawa. Finite element stabilization parameters computed from element matrices and vectors. *Computer Methods in Applied Mechanics and Engineering*, 190:411–430, 2000.

[95] T. E. Tezduyar. Computation of moving boundaries and interfaces and stabilization parameters. *International Journal for Numerical Methods in Fluids*, 43:555–575, 2003.

[96] T. J. R. Hughes, G. Scovazzi, and L. P. Franca. Multiscale and stabilized methods. In E. Stein, R. de Borst, and T. J. R. Hughes, editors, *Encyclopedia of Computational Mechanics, Vol. 3, Fluids*, chapter 2. Wiley, 2004.

[97] T. W. H. Sheu and M.-C. Hsu. Finite-element simulation of incompressible viscous flows in moving meshes. *Numerical Heat Transfer, Part B: Fundamentals*, 56:38–57, 2009.

[98] T. J. R. Hughes, G. R. Feijóo, L. Mazzei, and J. B. Quincy. The variational multiscale method–A paradigm for computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 166:3–24, 1998.

[99] R. Calderer and A. Masud. A multiscale stabilized ALE formulation for incompressible flows with moving boundaries. *Computational Mechanics*, 46:185–197, 2010.

[100] F. Brezzi, L. P. Franca, T. J. R. Hughes, and A. Russo. $b = \int g$. *Computer Methods in Applied Mechanics and Engineering*, 145:329–339, 1997.

[101] K. Takizawa and T. E. Tezduyar. Multiscale space–time fluid–structure interaction techniques. *Computational Mechanics*, 48:247–267, 2011.

[102] Y. Saad and M. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal of Scientific and Statistical Computing*, 7:856–869, 1986.

[103] M.-C. Hsu, I. Akkerman, and Y. Bazilevs. High-performance computing of wind turbine aerodynamics using isogeometric analysis. *Computers & Fluids*, 49:93–100, 2011.

[104] A. K. Karanam, K. E. Jansen, and C. H. Whiting. Geometry based pre-processor for parallel fluid dynamic simulations using a hierarchical basis. *Engineering with Computers*, 24:17–26, 2008.

[105] O. Sahni, M. Zhou, M. S. Shephard, and K. E. Jansen. Scalable implicit finite element solver for massively parallel processingwith demonstration to 160K cores. In *Proceedings of the ACM/IEEE Conference on High Performance Computing*, Portland, OR, USA, 2009.

[106] D. A. Spera. Introduction to modern wind turbines. In D. A. Spera, editor, *Wind Turbine Technology: Fundamental Concepts of Wind Turbine Engineering*, pages 47–72. ASME Press, 1994.

[107] Texas Advanced Computing Center (TACC). Available at: http://www.tacc.utexas.edu. Accessed 29 March 2012.

[108] E. Hau. *Wind Turbines: Fundamentals, Technologies, Application, Economics. 2nd Edition*. Springer, Berlin, 2006.

[109] Ranger User Guide. http://services.tacc.utexas.edu/index.php/ranger-user-guide. Accessed 29 March 2012.

[110] J. Chung and G. M. Hulbert. A time integration algorithm for structural dynamics withimproved numerical dissipation: The generalized-α method. *Journal of Applied Mechanics*, 60:371–75, 1993.

[111] K. E. Jansen, C. H. Whiting, and G. M. Hulbert. A generalized-α method for integrating the filtered Navier-Stokes equations with a stabilized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 190:305–319, 2000.

[112] T. E. Tezduyar, S. Sathe, R. Keedy, and K. Stein. Space–time finite element techniques for computation of fluid–structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 195:2002–2027, 2006.

[113] M. M. Hand, D. A. Simms, L. J. Fingersh, D. W. Jager, J. R. Cotrell, S. Schreck, and S. M. Larwood. Unsteady aerodynamics experiment phase VI: Wind tunnel test configurations and available data campaigns. Technical Report NREL/TP-500-29955, National Renewable Energy Laboratory, Golden, CO, 2001.

[114] J. Johansen, N. N. Sørensen, J. A. Michelsen, and S. Schreck. Detached-eddy simulation of flow around the NREL Phase VI blade. *Wind Energy*, 5:185–197, 2002.

[115] D. J. Laino, A. C. Hansen, and J. E. Minnema. Validation of the AeroDyn subroutines using NREL Unsteady Aerodynamics Experiment data. *Wind Energy*, 5:227–244, 2002.

[116] J. M. Jonkman. Modeling of the UAE wind turbine for refinement of FAST_AD. Technical Report NREL/TP-500-34755, National Renewable Energy Laboratory, Golden, CO, 2003.

[117] C. Tongchitpakdee, S. Benjanirat, and L. N. Sankar. Numerical simulation of the aerodynamics of horizontal axis wind turbines under yawed flow conditions. *Journal of Solar Energy Engineering*, 127:464–474, 2005.

[118] N. Sezer-Uzol and L. N. Long. 3-D time-accurate CFD simulations of wind turbine rotor flow fields. AIAA Paper 2006-0394, 2006.

[119] S. Schmitz and J. J. Chattot. Characterization of three-dimensional effects for the rotating and parked NREL Phase VI wind turbine. *Journal of Solar Energy Engineering*, 128:445–454, 2006.

[120] J. C. Huang, H. Lin, T. J. Hsieh, and T. Y. Hsieh. Parallel preconditioned WENO scheme for three-dimensional flow simulation of NREL Phase VI rotor. *Computers & Fluids*, 45:276–282, 2011.

[121] M.-C. Hsu, I. Akkerman, and Y. Bazilevs. Finite element simulation of wind turbine aerodynamics: Validation study using NREL Phase VI experiment. *Wind Energy*, 2012. In review.

[122] D. M. Somers. Design and experimental results for the S809 airfoil. Technical Report NREL/SR-440-6918, National Renewable Energy Laboratory, Golden, CO, 1997.

[123] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20:359–392, 1999.

[124] F. Shakib, T. J. R. Hughes, and Z. Johan. A multi-element group preconditioned GMRES algorithm for nonsymmetric systems arising in finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 75:415–456, 1989.

[125] Longhorn User Guide. Available at: http://www.tacc.utexas.edu/user-services/user-guides/longhorn-user-guide. Accessed 29 March 2012.

[126] D. D. Chao and C. P. van Dam. CFD analysis of rotating two-bladed flatback wind turbine rotor. Sandia Report SAND2008-1688, Sandia National Laboratories, Albuquerque, NM, 2008.

[127] T. Belytschko, W. K. Liu, and B. Moran. *Nonlinear Finite Elements for Continua and Structures*. Wiley, 2000.

[128] J. Kiendl. *Isogeometric Analysis and Shape Optimal Design of Shell Structures*. PhD thesis, Lehrstuhl für Statik, Technische Universität München, 2011.

[129] J. N. Reddy. *Mechanics of Laminated Composite Plates and Shells: Theory and Analysis, 2nd ed.* CRC Press, Boca Raton, FL, 2004.

[130] T. Belytschko, H. Stolarski, W. K. Liu, N. Carpenter, and J. S.-J. Ong. Stress projection for membrane and shear locking in shell finite elements. *Computer Methods in Applied Mechanics and Engineering*, 51:221–258, 1985.

[131] I. M. Daniel and O. Ishai. *Engineering Mechanics of Composite Materials*. Oxford University Press, New York, NY, 1994.

[132] LS-DYNA Finite Element Software: Livermore Software Technology Corp. http://www.lstc.com/products/ls-dyna. Accessed 29 March 2012.

[133] T. J. R. Hughes. *The finite element method: Linear static and dynamic finite element analysis*. Prentice Hall, Englewood Cliffs, NJ, 1987.

[134] R. T. Shield. Inverse deformation results in finite elasticity. *ZAMP*, 18:381–389, 1967.

[135] S. Govindjee and P. A. Mihalic. Computational methods for inverse finite elastostatics. *Computer Methods in Applied Mechanics and Engineering*, 136:47–57, 1996.

[136] S. W. Tsai and E. M. Wu. A general theory of strength for anisotropic materials. *Journal of Composite Materials*, 5:58–80, 1971.

[137] P. Hansbo and J. Hermansson. Nitsche's method for coupling non-matching meshes in fluid-structure vibrational problems. *Computational Mechanics*, 32:134–139, 2004.

[138] Y. Bazilevs, J. R. Gohean, T. J. R. Hughes, R. D. Moser, and Y. Zhang. Patient-specific isogeometric fluid–structure interaction analysis of thoracic aortic blood flow due to implantation of the Jarvik 2000 left ventricular assist device. *Computer Methods in Applied Mechanics and Engineering*, 198:3534–3550, 2009.

[139] Y. Bazilevs, M.-C. Hsu, D. J. Benson, S. Sankaran, and A. L. Marsden. Computational fluid–structure interaction: Methods and application to a total cavopulmonary connection. *Computational Mechanics*, 45:77–89, 2009.

[140] Y. Bazilevs, M.-C. Hsu, Y. Zhang, W. Wang, X. Liang, T. Kvamsdal, R. Brekken, and J. Isaksen. A fully-coupled fluid–structure interaction simulation of cerebral aneurysms. *Computational Mechanics*, 46:3–16, 2010.

[141] Y. Bazilevs, M.-C. Hsu, Y. Zhang, W. Wang, T. Kvamsdal, S. Hentschel, and J. Isaksen. Computational fluid–structure interaction: Methods and application to cerebral aneurysms. *Biomechanics and Modeling in Mechanobiology*, 9:481–498, 2010.

[142] Y. Zhang, W. Wang, X. Liang, Y. Bazilevs, M.-C. Hsu, T. Kvamsdal, R. Brekken, and J.G. Isaksen. High-fidelity tetrahedral mesh generation from medical imaging data for fluid–structure interaction analysis of cerebral aneurysms. *Computer Modeling in Engineering & Sciences*, 42:131–150, 2009.

[143] M.-C. Hsu and Y. Bazilevs. Blood vessel tissue prestress modeling for vascular fluid–structure interaction simulations. *Finite Elements in Analysis and Design*, 47:593–599, 2011.

[144] C. C. Long, M.-C. Hsu, Y. Bazilevs, J. A. Feinstein, and A. L. Marsden. Fluid–structure interaction simulations of the Fontan procedure using variable wall properties. *International Journal for Numerical Methods in Biomedical Engineering*, 28:512–527, 2012.

[145] Y. Bazilevs, M.-C. Hsu, K. Takizawa, and T. E. Tezduyar. ALE–VMS and ST–VMS methods for computer modeling of wind-turbine rotor aerodynamics and fluid–structure interaction. *Mathematical Models and Methods in Applied Sciences*, 2012. doi:10.1142/S0218202512300025.

[146] T. E. Tezduyar, M. Behr, S. Mittal, and A. A. Johnson. Computation of unsteady incompressible flows with the finite element methods – space–time formulations, iterative strategies and massively parallel implementations. In *New Methods in Transient Analysis*, PVP-Vol.246/AMD-Vol.143, pages 7–24, New York, 1992. ASME.

[147] T. Tezduyar, S. Aliabadi, M. Behr, A. Johnson, and S. Mittal. Parallel finite-element computation of 3D flows. *Computer*, 26(10):27–36, 1993.

[148] A. A. Johnson and T. E. Tezduyar. Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *Computer Methods in Applied Mechanics and Engineering*, 119:73–94, 1994.

[149] T. E. Tezduyar, S. Sathe, and K. Stein. Solution techniques for the fully-discretized equations in computation of fluid–structure interactions with the space–time formulations. *Computer Methods in Applied Mechanics and Engineering*, 195:5743–5753, 2006.

[150] T. E. Tezduyar and S. Sathe. Modeling of fluid–structure interactions with the space–time finite elements: Solution techniques. *International Journal for Numerical Methods in Fluids*, 54:855–900, 2007.

[151] U. Kuttler, C. Forster, and W. A. Wall. A solution for the incompressibility dilemma in partitioned fluid–structure interaction with pure Dirichlet fluid domains. *Computational Mechanics*, 38:417–429, 2006.

[152] H. Melbø and T. Kvamsdal. Goal oriented error estimators for Stokes equations based on variationally consistent postprocessing. *Computer Methods in Applied Mechanics and Engineering*, 192:613–633, 2003.

[153] E. H. van Brummelen, K. G. van der Zee, V. V. Garg, and S. Prudhomme. Flux evaluation in primal and dual boundary-coupled problems. *Journal of Applied Mechanics*, 79:010904, 2012.

[154] C. Farhat, M. Lesoinne, and P. Le Tallec. Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 157:95–114, 1998.

[155] M. A. Scott. *T-splines as a Design-Through-Analysis Technology*. PhD thesis, The University of Texas at Austin, August 2011.