POWER OPTIMIZATION OF WIND TURBINES SUBJECT TO NAVIER-STOKES EQUATIONS

YOUSEF AKHAVAN

A DISSERTATION SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN DEPARTMENT OF MATHEMATICS AND STATISTICS
YORK UNIVERSITY
TORONTO, ONTARIO

SEPTEMBER 2016

Abstract

In this thesis, we first develop a second-order corrected-explicit-implicit domain decomposition scheme (SCEIDD) for the parallel approximation of convection-diffusion equations over multi-block sub-domains. The stability and convergence properties of the SCEIDD scheme is analyzed, and it is proved that this scheme is unconditionally stable. Moreover, it is proved that the SCEIDD scheme is second-order accurate in time and space. Furthermore, three different numerical experiments are performed to verify the theoretical results. In all the experiments the SCEIDD scheme is compared with the EIPCMU2D scheme which is first-order in time.

Then, we focus on the application of numerical PDEs in wind farm power optimization. We develop a model for wind farm power optimization while considering the wake interaction among wind turbines. The proposed model is a PDE-constrained optimization model with the objective of maximizing the total power of the wind turbines where the three-dimensional Navier-Stokes equations are among the constraints. Moreover, we develop an efficient numerical algorithm to solve the model. This numerical algorithm is based on the pattern search method, the actuator line method and a numerical scheme which is used to solve the Navier-Stokes equations. Furthermore, the proposed numerical algorithm is used to investigate the wake structures. Numerical results are consistent with the field-tested results. Moreover, we find that by optimizing the turbines' operation while considering the wake effect, we can gain an additional 8% in the total power.

Finally, we relax the deterministic assumption for the incoming wind speed. The developed model is ultimately a PDE-constrained stochastic optimization model. Moreover, we develop an efficient numerical algorithm to solve this model. This numerical algorithm is based on the Monte Carlo simulation method, the pattern search method, the actuator line method and the corrected-explicit-implicit domain decomposition scheme which we develop for the parallel approximation of three-dimensional Navier-Stokes equations. The developed numerical algorithm, the parallel scheme, and the model are validated by a benchmark used in the literature and the experimental data. We find that by optimizing the turbines' operation and considering the randomness of incoming wind speed, we can gain an additional 9% in total power.

Acknowledgements

First and foremost, I would like to thank Professor Dong Liang and Professor Michael Chen for their invaluable guidance and constant support throughout my Ph.D. work. Without their academic guidance, continuous inquisition on the research topics and encouragements throughout my Ph.D. studies, I would not have completed this work. It was a privilege for me to work with extraordinary supervisors like them.

I would like to extend my sincere appreciations to my supervisory committee members, Prof. EJ Janse van Rensburg and Prof. Xin Gao for their valuable suggestions, feedback, and constructive advice.

This work, above all, owes much to my family for their unwavering support. Words can not express how grateful I am to my father, sisters, brothers and my mother who passed away from cancer. She sacrificed everything to give her children a better life. She was a critical part of our farm workforce and contributed toward the family income along with my father, not only carrying on housework and child care but also helping with the planting, harvesting, and improving the farm.

Finally, I would like to thank my supportive friends, Nariman Farsad, Mehdi Kargar, Masoud Ataei, Nikrooz Farsad, Morteza Zihayat, Mohammad Sajjadie and Alireza Moghadam.

Table of Contents

Abstract **ii**

Acknowledgements **iv**

Table of Contents **vi**

List of Tables **x**

List of Figures **xii**

Chapter One: Introduction **1**

1.1 Background 1

1.2 Work of the Thesis 9

Chapter Two: Second-Order Domain Decomposition Scheme **16**

2.1 Convection Diffusion Equations 16

2.2 Second-Order Corrected-Explicit-Implicit DD Scheme 17

2.3 Stability Analysis of the SCEIDD Scheme 22

2.4 Error Analysis 34

2.5 Numerical Experiment 38

2.6 Conclusion 50

Chapter Three: Power Optimization of Wind Turbines 54

3.1 Introduction 54

3.2 Power Production 55

3.3 Rotor Design 60

3.3.1 Airfoil Theory 60

3.3.2 Pitch Angle and Chord Length after Betz 62

3.3.3 Pitch Angle and Chord Length Considering Wake Rotation 65

3.4 Blade Element Theory 71

3.5 Modeling of the Flow for Power Optimization in Wind Farm 75

3.6 Blade Aerodynamics and Power Production 79

3.7 Numerical Solution of the Navier-Stokes Equations 85

3.7.1 Generalized Minimal Residual Algorithm 89

3.8 Joint Optimization Algorithm 89

3.9 Numerical Simulation 93

3.9.1 One Turbine 94

3.9.2 Two Turbines 102

3.10 Conclusion 104

Chapter Four: Stochastic Power Optimization Using DD 107

4.1 Introduction 107

4.2 Modeling of the Flow for Stochastic Power Optimization 108

4.3 Value of Stochastic Programming for Power Production 111

4.4 Scenario-Based Approximation Model 114

4.5 Parallel Scheme for Navier-Stokes Equations 116

4.5.1 Parallel Approximation of U 122

4.5.2 Parallel Approximation of V 124

4.5.3 Parallel Approximation of W 127

4.5.4 Parallel Approximation of ξ 130

4.5.5 Parallel Approximation of η 133

4.5.6 Parallel Approximation of ζ 135

4.6 Determination of Body Forces 138

4.7 Joint Optimization Using Stochastic Programming 139

4.8 Numerical Simulation 140

4.8.1 Validation of Parallel Simulation of Navier-Stokes Equations 140

4.8.2 Model Validation by Experimental Data 147

4.8.3 Numerical Results for NACA 23012 164

4.9 Conclusion 171

Bibliography **172**

List of Tables

Table 2.1: Error by SCEIDD at $T = 0.5$, with D=1e-2 and $h = \frac{1}{200}$. 41

Table 2.2: Error by SCEIDD at $T = 0.5$, with D=5e-3 and $h = \frac{1}{200}$. 41

Table 2.3: Ratio in h by SCEIDD at $T = 0.5$, with D=1e-2 and $\tau = \frac{h}{30}$. 42

Table 2.4: Ratio in h by SCEIDD at $T = 0.5$, with D=5e-3 and $\tau = \frac{h}{30}$. 42

Table 2.5: Ratio in τ by SCEIDD at $T = 0.5$, with D=1e-2 and $\tau = \frac{h}{20}$. 43

Table 2.6: Ratio in τ by SCEIDD at $T = 0.5$, with D=5e-3, $\tau = \frac{h}{20}$. 44

Table 2.7: Error by SCEIDD at $T = 0.5$, with $h = \frac{1}{200}$ and variable τ. 47

Table 2.8: Ratio in h by SCEIDD at $T = 0.5$, with D=5e-3 and $\tau = \frac{h}{30}$. 48

Table 2.9: Ratio in τ by SCEIDD at $T = 0.5$, with D=5e-3 and $\tau = \frac{h}{20}$. 49

Table 3.1: NACA 23012 coefficients, $\alpha \in [0°, 16°]$. 83

Table 3.2: Sections of NACA 23012 blade. 94

Table 3.3: Optimal operating point of NACA 23012. 96

Table 3.4: Optimal operating point of NACA 23012 (different initial point). . 97

Table 3.5: Joint optimal operating points of two NACA 23012. 105

Table 4.1: Deterministic optimal operating point of NACA 23012. 167

Table 4.2: Stochastic optimal operating point of NACA 23012. 168

Table 4.3: Joint optimal operating points of two NACA 23012. 168

List of Figures

Figure 2.1: Multiple subdomains.. 19

Figure 2.2: 6×1 subdomains (left) and 2×2 subdomains (right). 40

Figure 2.3: Sharp font approximated by our scheme SCEIDD at t=0.1. 51

Figure 2.4: Sharp font approximated by our scheme SCEIDD at t=0.2. 51

Figure 2.5: Computed moving sharp front with different schemes. 52

Figure 3.1: Interaction between the wind and a wind turbine. 56

Figure 3.2: C_p and C_T for an idealized wind turbine. 59

Figure 3.3: Velocities and angles on the wing section of the blade. 61

Figure 3.4: Chord length as a function of r, for different numbers of blades.. 65

Figure 3.5: Chord length as function of r, for different tip speed ratios. 66

Figure 3.6: Downstream rotation of the wake. 67

Figure 3.7: Relative velocities considering wake rotation. 67

Figure 3.8: Speeds in the plane of rotor. 68

Figure 3.9: Optimal chord length. 70

Figure 3.10: Optimal pitch angle. 71

Figure 3.11: Computational domain. 76

Figure 3.12: Cross-sectional airfoil element. 77

Figure 3.13: Cross-sectional airfoil element showing force vectors. 77

Figure 3.14: Coefficient of lift and drag. 80

Figure 3.15: Glide ratio as a function of the angle of attack. 82

Figure 3.16: Staggered grid. 88

Figure 3.17: Pattern search result for laminar flow $U_0 = 9$. 97

Figure 3.18: Pattern search result for laminar flow $U_0 = 10$. 98

Figure 3.19: Pattern search result for laminar flow $U_0 = 11$ 98

Figure 3.20: Wind flow in the plane of the rotor near the hub. 99

Figure 3.21: Wind flow in the plane of the rotor near and far from the hub. . 100

Figure 3.22: Wind flow in xz-plane passing through the hub. 100

Figure 3.23: Wind flow in xz-plane passing through the hub. 101

Figure 3.24: Schematic diagram of two turbines. 101

Figure 4.1: Generated power for various rotational velocity. 113

Figure 4.2: Staggered grid 119

Figure 4.3: Arbitrary mesh function at an arbitrary grid point. 120

Figure 4.4: Comparison of estimated solutions and exact solutions. 144

Figure 4.5: Comparison of estimated solutions and exact solutions. 145

Figure 4.6: Velocity profile for the lid driven cavity flow at z=0.5 plane. 146

Figure 4.7: Velocity profile for the lid driven cavity flow at y=0.5 plane. . . . 147

Figure 4.8: Velocity profile for the lid driven cavity flow at x=0.5 plane. . . . 148

Figure 4.9: Comparison of measured and computed power. 150

Figure 4.10: Mesh grid in the plane of the wind turbine. 152

Figure 4.11: Computed magnitude of vorticity at y-z planes for $U_0 = 10\frac{m}{s}$. . 153

Figure 4.12: Computed magnitude of vorticity behind the turbine. 154

Figure 4.13: Computed magnitude of vorticity at y-z planes in near wake. . . 155

Figure 4.14: Computed magnitude of vorticity at y-z planes in far wake. . . . 156

Figure 4.15: Computed magnitude of vorticity for different wind speed. 157

Figure 4.16: Distribution of axial factor in the rotor plane for $U_0 = 10$. 158

Figure 4.17: Distribution of axial factor in z-direction. 159

Figure 4.18: Distribution of axial factor along the blade (wake region). 160

Figure 4.19: Velocity profile in x-direction for wind speeds $U_0 = 3, \ldots, 6$. 161

Figure 4.20: Velocity profile in x-direction for wind speeds $U_0 = 7, \ldots, 10$. . . . 162

Figure 4.21: Velocity profile in x-direction for wind speeds $U_0 = 11, \ldots, 14$. . . 163

1 Introduction

1.1 Background

Unsteady convection-diffusion equations are important time-dependent partial differential equations that their numerical solutions arise in many important applications in science and engineering, such as simulation of underground water pollution, oil reservoir simulation, wind flow simulation, etc. (see [5, 8, 61, 64, 69]). In such problems, the convection term essentially dominates the diffusion term, leading to a nearly hyperbolic set of governing partial differential equations. Standard numerical methods are not capable of computing the solutions of such equations. They often introduce nonphysical oscillations into the approximated solutions, or they only have a first-order accuracy in space. To improve the accuracy of these schemes and avoid their oscillations, modified upwind schemes [48] that numerically simulate the direction of propagation of information in a flow field have been studied. Although modified upwind schemes suppress oscillations in numerical solutions, they could have a less promising performance for solving convection-diffusion equations in large-scale applications. In these applications, an extremely refined global mesh is needed which will increase the computational cost dramatically. Domain decomposition methods (DDMs) provide a feasible approach for handling these problems. DDMs reduce the computational cost by decomposing the global domain into smaller subdomains and solve sub-problems in different subdomains in parallel. For the parallel approximation of time-dependent parabolic equations, some explicit-implicit DDMs on non-overlapping subdomains were developed in [9, 17, 20, 45, 46]. The explicit-implicit domain decomposition (EIDD) method proposed by Kuznetsov [45] used a fully explicit scheme on the boundary of subdomains which causes numerical instability. To reduce the numerical instability, Dawson and Dupont [17] factorized the fully explicit scheme used on the boundary of subdomains into a partially explicit and partially implicit scheme. Their EIDD method achieved a better numerical stability; however, it was still not unconditionally stable. A penalized EIDD proposed by Black [9] achieved a numerically verified unconditional stability; nevertheless, it had a time step size restriction to attain a first-order temporal accuracy. To improve time step size restrictions of the EIDD methods, Du, Mu and Wu [20] proposed an alternative approach by using a multi-step explicit scheme on the interfaces of subdomains. To eliminate time step size restrictions of the EIDD methods, Zhuang and Sun [114] proposed a class of stabilized explicit-implicit domain decomposition algorithms by adding a stabilization step to the EIDD methods. Recently, a new technique called implicit correction in [47, 84, 111, 112]] is adopted to improve the parallel efficiency of EIDD methods by easing the time step-size restriction. The idea is to replace the predicted values on the interfaces with the new solutions computed by some implicit correction scheme, once the subdomain solutions are available at each time level. By adding the correction step to EIDD methods, the CEIDD algorithms exhibit much better numerical stability. Recently, the EIDD and CEIDD methods for the parabolic equations have been extended for the convection-diffusion equations in [19, 113]. Du and Liang [19] proposed an efficient EIDD methods by combining a splitting technique with the non-overlapping decomposition method. They used a multi-level explicit upstream scheme to compute the interface values on the boundaries of subdomains while the interior values of subdomains were computed by the splitting upstream one-dimensional implicit schemes. The CEIDD scheme proposed by Zhu [113] predicts the values on the interfaces by the linear combinations of the values at the current and previous time steps, computes the interior values in subdomains by an implicit modified upwind scheme and recomputes the values on the interfaces by an implicit scheme. However, the EIDD and CEIDD schemes developed in [19, 113] for convection-diffusion equations are only first-order in time step. It is an important and difficult task to develop time high-order non-overlapping domain decomposition schemes for convection-diffusion equations.

Numerical solutions of convection-diffusion equations arise in many important applications such as wind flow simulation which is essential for analyzing wind energy production of a wind farm. Wind energy, as an alternative to fossil fuels, is clean, plentiful, widely distributed, and it produces no greenhouse gas. It is an established source of energy, and its share in generating electricity has experienced a tremendous increase in the past decade. For example, Canada has experienced an average growth rate of 23% per year in generating electricity from wind energy in the past five years. Currently, wind energy is the fastest-growing source of electricity in the world and it is estimated that it will generate up to 18% of the world' s electricity by 2050. One of the keys of realizing this goal is to improve wind turbine' s performance in terms of power production. In this framework, the first step is how to analyze the performance of a single wind turbine. Analyzing and modeling a single wind turbine can be conveniently and elegantly conducted, using Blade Element Momentum (BEM) theory [12, 38, 54, 55, 66, 93]. The BEM is based on dividing the flow into annular control volumes, applying momentum balance and energy conservation in each control volume. The annuli are bounded by stream surfaces that enclose the rotor and extend from far upstream to far downstream. This method is simple to apply and has been popular for many years in analyzing the performance of a horizontal axis wind turbine. However, the BEM has limitations such as no aerodynamic interactions between different blade elements, and it usually underpredicts the power generated by a wind turbine (see [38, 60]). It is, therefore, necessary to use different methods to analyze the performance of a wind turbine. One such method is the vortex wake method [10,59,62,83,88,102] which is computationally expensive and another is the asymptotic acceleration potential method [36,100] which uses linearized flow equation for modeling of the airflow around wind turbines. Recently, the generalized actuator disk method [14,15,24,31,53,72,73,80,89,90,92,106] has been developed for analyzing a wind turbine. This method is a straightforward extension of the BEM method, and it uses tabulated airfoil data along with the conservation laws. The main difference is that, whereas the BEM is based on the assumption that there are no aerodynamic interactions between the flow in radial stream tubes, the generalized actuator disc method has no restriction on the kinematics of the flow since it is governed by the unsteady Navier-Stokes equations. The main limitation of the generalized actuator disc method is that it is valid for rotationally symmetric flow conditions since the forces at each spanwise section are distributed evenly. This also implies that the presence of the blades is taken as an integrated part in the circumferential direction, and hence the method cannot capture the influence of the tip vortices. To overcome the limitations of the generalized actuator disk model, actuator line model was developed by Sorensen [91]. This model combines three-dimensional Navier-Stokes equations with a technique in which the loading is distributed along lines representing the blades of the turbine in a fully three-dimensional domain. The kinematics of the wake is determined by solving Navier-Stokes equations numerically in one of three different formulations namely velocity-pressure [25,30,95,101,103,104,108], vorticity-vector-potential [23,29,51], or velocity-vorticity formulation [13,28,33,99,105,107,109,110], whereas the influence of the rotating blades on the flow field is included using tabulated airfoil data. The airfoil data and subsequent loading are determined iteratively by computing local angles of attack from the movement of the blades and the local flow field. This model has been used by other researchers [78,82,86,97,98] for studying the wake properties and analyzing the performance of a single turbine. However, it is quite challenging to analyze the performance of multiple turbines due to the wake interactions amongst the turbines.

Currently, wind turbines are operating at their own local optimum points to maximize their own performance. Many studies have shown that operating all turbines in a wind farm at their local optimum points leads to the suboptimal performance of the overall wind farm [43,58]. This is due to the wake generated by upstream wind turbines which alter the flow field and lead to a wind velocity deficit in downstream wind turbines [16,77,81,91]. As a consequence, if all wind turbines operate at their own local optimum points then downstream wind turbines cannot generate power as much as upstream wind turbines. For instance, Neustadter and

Spera [65] investigated the performance of three turbines separated by seven rotor diameters. They found that if all turbines operate at their own local optimum points then the power loss of downstream turbines can be as high as 10%. Another investigation by Rebecca [6] shows that the power loss of downstream wind turbines in full wake conditions can be as high as 30%, but when averaged over different wind directions, it is around 5-8%. These studies confirm that operating turbines at their local optimum points will lead to suboptimal performance of the overall wind farm. Therefore, in order to improve the performance of the overall wind farm, it is necessary to find the global optimum points of wind turbines by taking into account the impact of the wake on power production. Though Patricio [96] studied the total power optimization in a wind farm while considering the wake effect, he used an improvised BEM-alike method to model wind turbines' wake. Furthermore, he applied a rather inefficient grid search method to find optimum operating points of the upstream wind turbine while assuming that the downstream turbine is operating at its own local optimum points. Most of the work related to the power optimization in a wind farm study the power loss when the downstream turbines are operating at their local optimum points [6, 37, 39, 42, 65, 96] or wake of wind turbines is modeled via an improvised BEM-alike method. Little work has been done on explicitly optimizing the total power production of wind turbines while considering the wake impact.

One of the main characteristics of wind power generated by turbines in a wind farm is the inherent variability and unpredictability of the generation source which is incoming wind. Therefore, to further improve the performance of the overall wind farm, the random behavior of incoming wind speed must be taken into account when optimizing the total power and finding the global optimum points of wind turbines. To find a good probability distribution function to describe this random behavior of wind speed, a large number of studies have been done. Generally, the two-parameter Weibull distribution is widely used and accepted in the specialized literature on wind energy and other renewable energy sources [57, 79]. However, very little work has been done on explicitly optimizing the total power production of wind turbines while considering the randomness of wind speed as well as the wake impact.

Unsteady Navier-Stokes equations are not only important in modeling the airflow in a wind farm, but also play a vital role in providing solutions for a wide range of engineering problems [1, 26, 71]. In such problems, obtaining an accurate simulation of wind flow governed by the Navier-Stokes equations requires a great number of mesh points which can lead to the problem of solving large linear systems. It is, therefore, greatly beneficial to obtain solutions in reasonable time. However, even without real-time applications in mind, reducing the computational cost is always beneficial, as this enables us to study increasingly large and complex problems. There seems to be an agreement that direct methods are too expensive to handle them efficiently. Hence, an alternative approach is desired. Among the possible alternative paths that can be followed, domain decomposition comes to mind quite naturally. The task of approximating the wind flow in a wind farm could be split into small tasks that are be dealt with in parallel. The Schwartz-type (overlapping domains) decomposition algorithm for the numerical solution of the Navier-Stokes equations and other methods have been studied in [4, 27]. In the framework of explicit-implicit domain decomposition scheme for the Navier-Stokes equations, very little work has been done. It is, therefore, very desirable to develop an efficient explicit-implicit domain decomposition scheme to solve three-dimensional Navier-Stokes equations in reasonable time.

1.2 Work of the Thesis

In this thesis, we first develop an unconditional second-order corrected-explicit-implicit domain decomposition scheme (SCEIDD) over non-overlapping subdomains for the parallel approximation of convection-diffusion equations. This is achieved by combining a second-order extrapolation scheme, implicit correction technique, and modified upwind schemes. In the proposed SCEIDD scheme, the computational domain at first is decomposed into non-overlapping subdomains and then each subdomain is discretized by a nonuniformly partitioned mesh. Moreover, at each time step, we predict the values at the interface mesh points by a time second-order extrapolation scheme. Then, we approximate the interior values in each subdomain by a second-order implicit scheme where the time derivative is discretized by the linear combination of the backward Euler differences at the current and previous time steps and the convection-diffusion term is discretized by the modified upwind scheme. Finally, once the subdomain solutions are available, we correct the predicted values at the interface mesh points. The proposed SCEIDD scheme has three main features. First, it is unconditionally stable; hence, there is no restriction on time step size. Second, it has second-order accuracy in both time and space whereas previously developed domain decomposition schemes have the first-order accuracy in time. Third, in the proposed SCEIDD scheme, non-overlapping subdomains have simple geometry, while in comparison to the scheme proposed in [85], non-overlapping subdomains have complicated geometry. Consequently, less effort is needed for numerical simulation. We also analyze the stability and convergence properties of the SCEIDD scheme for convection-diffusion equations with variable coefficients. We prove that the proposed scheme is unconditionally stable, and it is second-order accurate in both time and space. Finally, we provide three different numerical experiments to verify the theoretical results. The goal of the first experiment is to show that the SCEIDD scheme is second-order in time and space. The goal of the second experiment is to show that the SCEIDD scheme maintains its accuracy as the number of subdomains increases. The goal of the third experiment is to show that the SCEIDD scheme estimates accurately the solution of convection-diffusion equations with a discontinuous initial solution. Furthermore, in all the experiments the SCEIDD scheme is compared with the EIPCMU2D scheme [113] which is first-order in time and developed recently for the parallel approximation of the convection-diffusion equations.

In this thesis, we then focus on the application of numerical PDEs in wind farm power optimization. We develop a model for wind farm power optimization while considering the wake interaction among wind turbines. The proposed model is a PDE-constrained optimization model with the objective of maximizing the total power of the wind turbines where the three-dimensional Navier-Stokes equations are the constraints. In this model, the three-dimensional Navier-Stokes equations are used to model the airflow as well as interacting wakes in the wind farm where the external forces in these equations represent the loading of wind turbines. Moreover, we develop an efficient numerical algorithm to solve the model accurately. This numerical algorithm is based on the pattern search method, the actuator line method and an efficient numerical scheme which is used to solve the three-dimensional Navier-Stokes equations in velocity-vorticity formulation. In the proposed numerical algorithm, we employ patten search method to find the global optimum operating points of wind turbines. Moreover, in the pattern search method, we evaluate the objective value, the total power of wind turbines, using the actuator line method and the three-dimensional Navier-Stokes equations solver. In this regard, the three-dimensional Navier-Stokes equations solver and the actuator line are connected in the following sense: 1) an input parameter of the three-dimensional Navier-Stokes equations solver, the external forces, is an output of the actuator line, and an input parameter of the actuator line, the wind speed at the plane of wind turbines, is an output of the three-dimensional Navier-Stokes equations solver. We iterate through the three-dimensional Navier-Stokes equations solver and the actuator line method until convergence. Moreover, in the actuator line method, the external forces which represent the loading of wind turbines are computed using tabulated airfoil data. Furthermore, we employ an efficient numerical scheme which uses the false-transient technique, backward Euler method and explicit scheme to solve the three-dimensional Navier-Stokes equations in a velocity-vorticity formulation. In this numerical scheme, the velocity Poisson equations are made parabolic using the false-transient technique and are solved along with the vorticity transport equations. The parabolic velocity Poisson equations are advanced in time using backward Euler method and are solved along with the continuity equation for velocities, thus ensuring a divergence-free velocity field. The vorticity transport equations in conservative form are solved using explicit scheme for the non-linear term and implicit scheme for viscosity term. Finally, we present two numerical case studies to test the efficiency and accuracy of the proposed numerical algorithm. We first apply the proposed numerical algorithm to find the optimal operating points of a single turbine as well as to investigate the near-wake and far-wake structures. These optimal operating points and the wake characteristic are consistent with the field-tested results. Then, we apply the proposed numerical algorithm to find the global optimal operating points of multiple turbines operating in a wind farm. We find that by operating wind turbines at their global optimal operating points, we can safely gain an additional 8% in the total power.

Finally, in this thesis, to push the proposed model to further realism, we relax the deterministic assumption for the incoming wind speed on the boundaries of wind farm and treat it as a stochastic variable. The developed model is ultimately a PDE-constrained stochastic optimization model; the decision variables, the objective function, and the constraints are the same as in the previously developed model. Moreover, we develop an efficient numerical algorithm to solve this model accurately and efficiently. This numerical algorithm is based on the Monte Carlo simulation method, the pattern search method, the actuator line method and the corrected-explicit-implicit domain decomposition scheme which we develop for the parallel approximation of three-dimensional Navier-Stokes equations. In the proposed numerical algorithm, we apply the pattern search method to find the operating points of wind turbines which optimize the total power. Furthermore, in the pattern search method, we compute the objective value, the total power, for a given decision variables, operating points of wind turbines, using Monte Carlo simulation method, the actuator line method and the corrected-explicit-implicit domain decomposition scheme. First, Monte Carlo simulation method is used for generating scenarios where the random samples are drawn from the doubly truncated Weibull distribution with the given probability distribution function. Then, for a given sample which is the speed of the incoming wind, we apply the actuator line method and the three-dimensional Navier-Stokes equations solver to compute the total power. Moreover, we develop a corrected-explicit-implicit domain decomposition scheme for the parallel approximation of the three-dimensional Navier-Stokes equations in a velocity-vorticity formulation. This is achieved by combining a second-order extrapolation scheme, an implicit correction technique, and the false-transient technique. In the proposed scheme, at each time step, we predict the values of velocity at interface mesh points by a time second-order extrapolation scheme. Then, we approximate the interior values of velocity in each subdomain using the backward Euler method, an explicit scheme, an implicit scheme and the false-transient method. Subsequently, once the subdomain solutions are available, we correct the predicted values of velocity at the interface mesh points. Moreover, we use similar steps to those of solving the velocity equations to solve the vorticity transport equations. Finally, the developed numerical algorithm, the parallel scheme and the model are validated by a benchmark used in the literature and the experimental data. It is shown that by taking into account the randomness of wind speed and optimizing the total power, we can improve the performance of wind turbines in a wind farm. We find that by optimizing the turbines' operation and taking into account the randomness of wind speed, we can gain an additional 9%, in total power.

2 Second-Order Domain Decomposition Scheme

2.1 Convection Diffusion Equations

In this chapter, we develop a second-order corrected-explicit-implicit domain decomposition scheme (SCEIDD) for the parallel approximation of convection-diffusion equations over multi-block sub-domains. The stability and error analysis of the SCEIDD scheme for convection-diffusion equations with variable coefficients are analyzed. We prove that the SCEIDD scheme has second-order accuracy in both time and space, and it has no stability condition. Moreover, numerical experiments are provided to verify the theoretical results.

2.2 Second-Order Corrected-Explicit-Implicit DD Scheme

Consider the following two-dimensional convection-diffusion equations:

$$\begin{cases} u_t = L(u) + f(X,t), & (X,t) \in \Omega \times (0,T], \\ u(X,t) = 0, & (X,t) \in \partial\Omega \times (0,T], \\ u(X,0) = u_0(X), & X \in \Omega, \end{cases} \tag{2.1}$$

where the operator $L(u)$, is defined as:

$$\begin{aligned} L(u) \;=\; & \frac{\partial}{\partial x}(a_1(x,y)\frac{\partial}{\partial x}u(x,y,t)) + \frac{\partial}{\partial y}(a_2(x,y)\frac{\partial}{\partial y}u(x,y,t)) \\ & -\frac{\partial}{\partial x}(b_1(x,y)u(x,y,t)) - \frac{\partial}{\partial y}(b_2(x,y)u(x,y,t)) \\ & -c(x,y)u(x,y,t). \end{aligned} \tag{2.2}$$

Here the variables $t \geq 0$, $X = (x,y)$, $T > 0$, $\Omega = (0,1) \times (0,1)$ and $\partial\Omega$ denote the time, spatial coordinates, final time, the computational domain and the boundary of the computational domain, respectively; $f(X,t)$ is the given source term, and $a_1(x,y), a_2(x,y)$ are the diffusion coefficients where $a_1(x,y), a_2(x,y) \geq a_0 > 0, (x,y) \in \bar{\Omega}$; $\overrightarrow{b}(x,y) = (b_1(x,y), b_2(x,y))$ is the velocity field, and $u_0(X)$ is the given initial condition. Let $t^n = n\tau$ be the time step where $\tau = T/N$, for some integer N, and let discretize the computational domain Ω by a uniformly partitioned mesh Ω_h. Th points in the mesh Ω_h are (x_i, y_j), $x_i = ih_x$, $1 \leq i \leq J_x - 1$, $y_j = jh_y$, $1 \leq j \leq J_y - 1$, where $h_x = 1/J_x$ and $h_y = 1/J_y$, for some integers $J_x > 0$ and $J_y > 0$, are the spatial step sizes. Now, we introduce a mesh function

$U^n_{i,j} = U(x_i, y_j, t_n)$ which approximates the exact solution at the mesh points, and it is used in the following notation:

$$\partial_t U^n_{i,j} = \frac{U^n_{i,j} - U^{n-1}_{i,j}}{\tau}, \quad \delta_x U^n_{i-\frac{1}{2},j} = \frac{U^n_{i,j} - U^n_{i-1,j}}{h_x}, \quad \delta_y U^n_{i,j-\frac{1}{2}} = \frac{U^n_{i,j} - U^n_{i,j-1}}{h_y}, \tag{2.3}$$

$$\delta_x(b_1 U^{n,ux})_{i,j} = \tfrac{1}{h_x}(b_{1i+\frac{1}{2},j} U^{n,ux}_{i+\frac{1}{2},j} - b_{1i-\frac{1}{2},j} U^{n,ux}_{i-\frac{1}{2},j}), \tag{2.4}$$

$$\delta_y(b_2 U^{n,uy})_{i,j} = \tfrac{1}{h_y}(b_{2i,j+\frac{1}{2}} U^{n,uy}_{i,j+\frac{1}{2}} - b_{2i,j-\frac{1}{2}} U^{n,uy}_{i,j-\frac{1}{2}}), \tag{2.5}$$

where

$$U^{n,ux}_{i+\frac{1}{2},j} = H(b_{1i+\frac{1}{2},j}) U^n_{i,j} + (1 - H(b_{1i+\frac{1}{2},j})) U^n_{i+1,j}, \tag{2.6}$$

$$U^{n,uy}_{i,j+\frac{1}{2}} = H(b_{2i.j+\frac{1}{2}}) U^n_{i,j} + (1 - H(b_{2i,j+\frac{1}{2}})) U^n_{i,j+1}, \tag{2.7}$$

and

$$\delta^2_{x,a_1^*} U^n_{i,j} = \tfrac{1}{h_x}(a^*_{1i+\frac{1}{2},j} \delta_x U^n_{i+\frac{1}{2},j} - a^*_{1i-\frac{1}{2},j} \delta_x U^n_{i-\frac{1}{2},j}), \tag{2.8}$$

$$\delta^2_{y,a_2^*} U^n_{i,j} = \tfrac{1}{h_y}(a^*_{2i,j+\frac{1}{2}} \delta_y U^n_{i,j+\frac{1}{2}} - a^*_{2i,j-\frac{1}{2}} \delta_y U^n_{i,j-\frac{1}{2}}), \tag{2.9}$$

where

$$a^*_{1i-\frac{1}{2},j} = \tfrac{2a^2_{1i-\frac{1}{2},j}}{2a_{1i-\frac{1}{2},j} + h_x |b_{1i-\frac{1}{2},j}|}, \tag{2.10}$$

$$a^*_{2i,j-\frac{1}{2}} = \tfrac{2a^2_{2i,j-\frac{1}{2}}}{2a_{2i,j-\frac{1}{2}} + h_y |b_{2i,j-\frac{1}{2}}|}. \tag{2.11}$$

Here the piecewise function $H(x)$ is defined by:

$$H(x) = \begin{cases} 1, & x \geq 0, \\ 0, & x < 0. \end{cases}$$

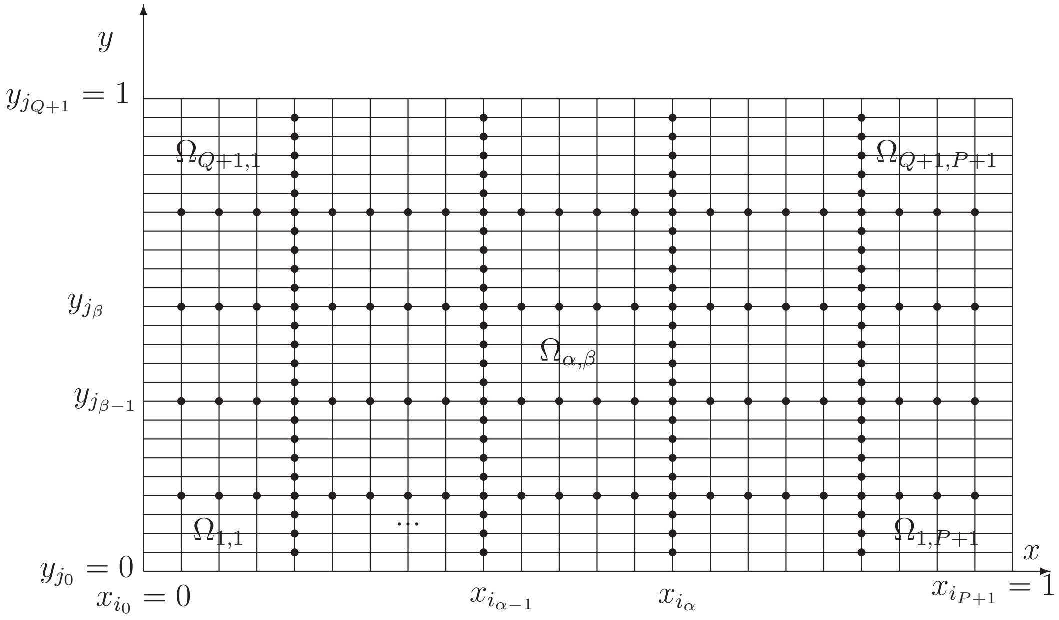


Figure 2.1: Multiple subdomains.

To develop a second-order finite difference scheme in time, we use the linear combination of the backward Euler differences at the current and previous time steps to discretize the time derivative $\frac{\partial u}{\partial t}$:

$$L_\tau U_{i,j}^n \;\; = \;\; \partial_t U_{i,j}^n + \frac{1}{2\tau}(U_{i,j}^n - 2U_{i,j}^{n-1} + U_{i,j}^{n-2}). \qquad (2.12)$$

To approximate the convection-diffusion term by a second-order finite difference scheme, we define the operator $L_h(U)$ which is based on the modified upwind scheme [49, 50, 61]:

$$L_h U_{i,j}^n \;\; = \;\; \delta_{x,a_1^*}^2 U_{i,j}^n + \delta_{y,a_2^*}^2 U_{i,j}^n - \delta_x(b_1 U^{n,ux})_{i,j} - \delta_y(b_2 U^{n,uy})_{i,j} - c_{i,j}U_{i,j}^n . (2.13)$$

Now, we decompose the computational domain $\Omega = (0,1) \times (0,1)$ into $(P +$

$1)(Q+1)$ non-overlapping subdomains as shown in Figure 1. In general, P and Q are related to the size of the problem and the number of processors in the computer platform. The subdomains may have different widths and heights. For the theoretical analysis, we assume that each subdomain has at least one mesh point, implying $2 \leq P+1 \leq \left\lfloor \frac{J_x}{2} \right\rfloor - 1$, $2 \leq Q+1 \leq \left\lfloor \frac{J_y}{2} \right\rfloor - 1$. Associated to the subdomains, there are $(P+1)(Q+1)$ interfaces. Let $\Gamma_h = \Gamma_h^1 \bigcup \Gamma_h^2$ be the set of all mesh points on the interfaces, and let $\Gamma_h^3 = \Gamma_h^1 \cap \Gamma_h^2$ be the set of intersection points of interface boundaries where Γ_h^1 and Γ_h^2 are given by:

$$\begin{aligned} \Gamma_h^1 &= \{(i_\alpha h_x, jh_y) \mid 4 \leq i_\alpha + 2 \leq i_{\alpha+1}, \quad 1 \leq \alpha \leq P, \quad 1 \leq j \leq (J_y - 1)\}, \\ \Gamma_h^2 &= \{(ih_x, j_\beta h_y) \mid 4 \leq j_\beta + 2 \leq j_{\beta+1}, \quad 1 \leq \beta \leq Q, \quad 1 \leq i \leq (J_x - 1)\}. \end{aligned} \tag{2.14}$$

Here, $i_0 = 0$, $i_{P+1} = J_x$, $j_0 = 0$ and $j_{Q+1} = J_y$. Therefore, Ω_h is decomposed into $(P+1)(Q+1)$ non-overlapping subdomains:

$$\Omega_{h\alpha,\beta} = \{(ih_x, jh_y) \mid \ i_{\alpha-1} < i < i_\alpha, \quad j_{\beta-1} < j < j_\beta\}.$$

Now, we propose a **second-order corrected-explicit-implicit domain decomposition scheme** (SCEIDD) over multi-block subdomains which consists of the following steps (for $n \geq 2$):

Step 1. Apply the following explicit scheme to predict the value of $U_{i,j}^n$ at the interface mesh points by:

$$\tilde{U}_{i,j}^n = 2U_{i,j}^{n-1} - U_{i,j}^{n-2} \qquad \text{on } \Gamma_h. \tag{2.15}$$

Step 2. Compute the value of $U^n_{i,j}$ at the interior points of subdomains by:

$$\begin{cases} \dfrac{U^n_{i,j}-U^{n-1}_{i,j}}{\tau}+\dfrac{(U^n_{i,j}-2U^{n-1}_{i,j}+U^{n-2}_{i,j})}{2\tau} = L_hU^n_{i,j}+f^n_{i,j} & \text{on} \quad \Omega_h\backslash\{\Gamma_h\cup\partial\Omega_h\}\ , \\ U^n_{i,j}=\tilde{U}^n_{i.j} & \text{on} \quad \Gamma_h\ . \end{cases} \tag{2.16}$$

Step 3. Correct the predicted value of $U^n_{i,j}$ at the interface mesh points by:

$$\begin{cases} \dfrac{U^n_{i,j}-U^{n-1}_{i,j}}{\tau}+\dfrac{(U^n_{i,j}-2U^{n-1}_{i,j}+U^{n-2}_{i,j})}{2\tau} = L_hU^n_{i,j}+f^n_{i,j} & \text{on } \Gamma_h\backslash\Gamma_h^3\ , \\ U^n_{i,j}=\tilde{U}^n_{i.j} & \text{on } \Gamma_h^3. \end{cases} \tag{2.17}$$

Step 4. Correct the predicted value of $U^n_{i,j}$ at the intersection points of interface boundaries by:

$$\frac{U^n_{i,j}-U^{n-1}_{i,j}}{\tau}+\frac{(U^n_{i,j}-2U^{n-1}_{i,j}+U^{n-2}_{i,j})}{2\tau} = L_hU^n_{i,j}+f^n_{i,j} \quad \text{on } \Gamma_h^3. \tag{2.18}$$

The boundary conditions are:

$$U^n_{i,j}=0, (x_i,y_j)\in\partial\Omega_h, i=1\ldots J_x, j=1\ldots J_y, \tag{2.19}$$

and the initial values are given by:

$$U^0_{i,j}=u_0(ih,jh), \text{ on } \ \Omega_h. \tag{2.20}$$

In the proposed scheme, for the first time step, we can compute the value o f $U^1_{i,j}$ by any scheme which has second-order accuracy both in time and space such as the Crank-Nicolson scheme:

$$\begin{cases} \partial_t U^1_{i,j} = L_h\dfrac{U^1_{i,j}+U^0_{i,j}}{2}+f^{\frac{1}{2}}_{i,j} & \text{on } \Omega_h, \\ U^1_{i,j}=0 & \text{on } \partial\Omega_h. \end{cases} \tag{2.21}$$

The SCEIDD scheme (2.15)-(2.21) is simple, and it can be used for a parallel implementation to tackle the high computational complexity when solving convection-diffusion equations in large-scale applications. The steps for parallel implementation of the scheme is given in Algorithm 1.

2.3 Stability Analysis of the SCEIDD Scheme

In this section, we analyze the stability of the SCEIDD scheme. Throughout this section, any subscript C will denote a generic positive constant that depends on the exact solution $u(x, y, t)$, the convection and diffusion coefficients. However, it is independent of the time step τ, the spatial steps h_x, h_y, and the number of subdomains (P+1)(Q+1). First, we give the definition of the inner product of two mesh functions and L^2 norm of a mesh function which will be used in the stability analysis of the SCEIDD scheme. The inner product of two mesh functions U^n and V^n is defined by $\langle U^n, V^n \rangle = \sum_{i=1}^{J_x} \sum_{j=1}^{J_y} U_{i,j}^n V_{i,j}^n h_x h_y$, and the L^2 norm of U^n is defined by $\|U^n\|^2 = \langle U^n, U^n \rangle$. Now, we define H^1 seminorms which are based on the L^2 norm:

$$|U^n|_{a^*,1}^2 = \|\sqrt{a_1^*}\delta_x U^n\|^2 + \|\sqrt{a_2^*}\delta_y U^n\|^2,$$

$$|U^n|_{b,1}^2 = \|\sqrt{b_1^+ h_x}\delta_x U^n\|^2 + \|\sqrt{b_2^+ h_y}\delta_y U^n\|^2,$$

Algorithm 1 Parallel SCEIDD algorithm for 2-D convection-diffusion problems

```
Set $U^0_{i,j} = u(ih_x, jh_y, n)$ for $i = 0, \ldots, J_x$, $j = 0, \ldots, J_y$.
Solve the linear system of equations (2.21) to obtain $U^1_{i,j}$, $i = 1, \ldots, J_x - 1$, $j = 1, \ldots, J_y - 1$.
Assign the subdomain $\Omega_{h\alpha,\beta}$ to the processor $P_{\alpha,\beta}$ ($\alpha = 1, \ldots, (P+1)$, $\beta = 1, \ldots, (Q+1)$) and the interface $\Gamma_h$ to the processor $I$.
while $n \leq N$ do
    At the processor $I$, predict the value of $U^n_{i,j}$ at the interface mesh points using (2.15), then pass the interface values associated with the subdomain $\Omega_{h\alpha,\beta}$ to the processor $P_{\alpha,\beta}$ ($\alpha = 1, \ldots, (P+1)$, $\beta = 1, \ldots, (Q+1)$).
    At the processors $P_{1,1}, \ldots,$ $P_{P+1,Q+1}$, solve the linear system of equations (2.16) to obtain the value of $U^n_{i,j}$ at the interior points of subdomains in parallel. Then, at each processor $P_{\alpha,\beta}$, pass the values of $U^n_{i,j}$, at the mesh points adjacent to the interface mesh points, to the processor $I$ for correction computation.
    At the processor $I$, correct the predicted values of $U^n_{i,j}$ at the interface mesh points and at the intersection points of interface boundaries by (2.17) and (2.18), respectively.
end while
```

where

$$
\begin{aligned}
\|\sqrt{a_1^*}\delta_x U^n\|^2 &= \sum_{i=1}^{Jx}\sum_{j=1}^{Jy} a^*_{1i-\frac{1}{2},j}(\delta_x U^n_{i-\frac{1}{2},j})^2 h_x h_y,\\
\|\sqrt{a_2^*}\delta_y U^n\|^2 &= \sum_{i=1}^{Jx}\sum_{j=1}^{Jy} a^*_{2i,j-\frac{1}{2}}(\delta_y U^n_{i,j-\frac{1}{2}})^2 h_x h_y,\\
\|\sqrt{b_1^+ h_x}\delta_x U^n\|^2 &= \sum_{i=1}^{Jx}\sum_{j=1}^{Jy} b^+_{1i-\frac{1}{2},j}(\delta_x U^n_{i-\frac{1}{2},j})^2 h_x^2 h_y,\\
\|\sqrt{b_2^+ h_y}\delta_y U^n\|^2 &= \sum_{i=1}^{Jx}\sum_{j=1}^{Jy} b^+_{2i,j-\frac{1}{2}}(\delta_y U^n_{i,j-\frac{1}{2}})^2 h_x h_y^2.
\end{aligned}
$$

In the above equations, $b^+(x,y) = b(x,y)H(b(x,y))$ and $b^-(x,y) = -b(x,y)(1 - H(b(x,y)))$, which is equivalent to the definition of positive and negative parts of a function. Before performing the stability analysis, we introduce the following notations which are used to obtain the compact form of the SCEIDD scheme:

$$
\tilde{Q}^n_{1ai,j} = \begin{cases} a^*_{1i-\frac{1}{2},j} h_x^{-2}(\tilde{U}^n_{i-1,j} - U^n_{i-1,j}) & \text{if} \quad (x_{i-1}, y_j) \in \Gamma^1_h, \\ a^*_{1i+\frac{1}{2},j} h_x^{-2}(\tilde{U}^n_{i+1,j} - U^n_{i+1,j}) & \text{if} \quad (x_{i+1}, y_j) \in \Gamma^1_h, \\ 0 & \text{Otherwise,} \end{cases}
$$

$$
\tilde{Q}^n_{2ai,j} = \begin{cases} a^*_{2i,j-\frac{1}{2}} h_y^{-2}(\tilde{U}^n_{i,j-1} - U^n_{i,j-1}) & \text{if} \quad (x_i, y_{j-1}) \in \Gamma^2_h, \\ a^*_{2i,j+\frac{1}{2}} h_y^{-2}(\tilde{U}^n_{i,j+1} - U^n_{i,j+1}) & \text{if} \quad (x_i, y_{j+1}) \in \Gamma^2_h, \\ 0 & \text{Otherwise,} \end{cases}
$$

$$\tilde{Q}^n_{1bi,j} = \begin{cases} b^+_{1i-\frac{1}{2},j} h_x^{-1}(\tilde{U}^n_{i-1,j} - U^n_{i-1,j}) & \text{if} \quad (x_{i-1}, y_j) \in \Gamma^1_h, \\ b^-_{1i+\frac{1}{2},j} h_x^{-1}(\tilde{U}^n_{i+1,j} - U^n_{i+1,j}) & \text{if} \quad (x_{i+1}, y_j) \in \Gamma^1_h, \\ 0 \quad \text{Otherwise}, \end{cases}$$

$$\tilde{Q}^n_{2bi,j} = \begin{cases} b^+_{2i,j-\frac{1}{2}} h_y^{-1}(\tilde{U}^n_{i,j-1} - U^n_{i,j-1}) & \text{if} \quad (x_i, y_{j-1}) \in \Gamma^2_h, \\ b^-_{2i,j+\frac{1}{2}} h_y^{-1}(\tilde{U}^n_{i,j+1} - U^n_{i,j+1}) & \text{if} \quad (x_i, y_{j+1}) \in \Gamma^2_h, \\ 0 \quad \text{Otherwise}. \end{cases}$$

Therefore, using the above equations, the SCEIDD scheme (2.15)-(2.20) can be written in a compact form:

$$L_\tau U^n_{i,j} = L_h U^n_{i,j} + f^n_{i,j} + \tilde{Q}^n_{1ai,j} + \tilde{Q}^n_{1bi,j} + \tilde{Q}^n_{2ai,j} + \tilde{Q}^n_{2bi,j}. \tag{2.22}$$

Lemma 2.3.1. *Let U^n be a mesh function satisfying the boundary condition $U^n_{i,j} = 0, (x_i, y_j) \in \partial\Omega_h$. Then it holds that:*

$$\begin{aligned} \langle \delta^2_{x,a_1^*} U^n + \delta^2_{y,a_2^*} U^n, \partial_t U^n \rangle &= -\frac{1}{2}\partial_t(|U^n|^2_{a^*,1}) - \frac{\tau}{2}|\partial_t U^n|^2_{a^*,1}, & (2.23) \\ \langle \frac{\partial_t U^n - \partial_t U^{n-1}}{2}, \partial_t U^n \rangle &= \frac{\tau}{4}\partial_t(\|\partial_t U^n\|^2) + \frac{1}{4}\|\partial_t U^n - \partial_t U^{n-1}\|^2, & (2.24) \\ \|U^n\|^2 &\leq C|U^n|^2_{a^*,1}. & (2.25) \end{aligned}$$

Proof: From the boundary condition and Abel's formulation we have:

$$\begin{aligned}\langle \delta^2_{x,a_1^*}U^n &+& \delta^2_{y,a_2^*}U^n, \partial_t U^n\rangle = \\ &-& \sum_{i=1}^{J_x-1}\sum_{j=1}^{J_y-1}\frac{1}{\tau}a^*_{1i+\frac{1}{2},j}\delta_x U^n_{i+\frac{1}{2},j}(a^*_{1i+\frac{1}{2},j}\delta_x U^n_{i+\frac{1}{2},j} - a^*_{1i-\frac{1}{2},j}\delta_x U^{n-1}_{i-\frac{1}{2},j})h_x h_y, \\ &-& \sum_{i=1}^{J_x-1}\sum_{j=1}^{J_y-1}\frac{1}{\tau}a^*_{2i,j+\frac{1}{2}}\delta_x U^n_{i,j+\frac{1}{2}}(a^*_{2i,j+\frac{1}{2}}\delta_x U^n_{i,j+\frac{1}{2}} - a^*_{2i,j-\frac{1}{2}}\delta_x U^{n-1}_{i,j-\frac{1}{2}})h_x h_y.\end{aligned} \tag{2.26}$$

Then, one can derive the equation (2.23) from the above equation by applying the following equality:

$$a(b-c) = \frac{1}{2}|a-c|^2 - \frac{1}{2}|a-b|^2 + \frac{1}{2}|b|^2 - \frac{1}{2}|c|^2 \quad \text{for any } a, b, c \in R. \tag{2.27}$$

Using (2.27), one can also derive the equation (2.24) which is straightforward. Now, we complete the proof by deriving the inequality (2.25). From the boundary condition, $U^n_{i,j} = 0$ on $\partial\Omega_h$, we have:

$$U^n_{i,j} = h_x\delta_x U^n_{\frac{1}{2},j} + \cdots + h_x\delta_x U^n_{i-\frac{1}{2},j}.$$

Using the Schwartz inequality, we obtain:

$$|U^n_{i,j}|^2 \le i h_x^2 \sum_{k=1}^{i}|\delta_x U^n_{k-\frac{1}{2},j}|^2 \le h_x \sum_{i=1}^{J_x}|\delta_x U^n_{i-\frac{1}{2},j}|^2.$$

Multiplying both sides of the above inequality by $h_x h_y$ and summing over i and j, we get:

$$\begin{aligned}\sum_{i=1}^{J_x}\sum_{j=1}^{Jy}|U^n_{i,j}|^2 h_x h_y &\le& \sum_{i=1}^{J_x}\sum_{j=1}^{J_y}(h_x\sum_{i=1}^{J_x}|\delta_x U^n_{i-\frac{1}{2},j}|^2)h_x h_y = \sum_{i=1}^{J_x} h_x(\sum_{i=1}^{J_x}\sum_{j=1}^{J_y}|\delta_x U^n_{i-\frac{1}{2},j}|^2 h_x h_y) \\ &\le& C|U^n|^2_{a^*,1}.\end{aligned}$$

Lemma 2.3.2. *Let U^n be a mesh function satisfying the boundary condition $U^n_{i,j} = 0, (x_i, y_j) \in \partial\Omega_h$. Then we have that:*

$$\begin{aligned}
\langle \delta_x(b_1U^{n,ux}) &+ \delta_y(b_2U^{n,uy}), \partial_t U^n\rangle = \frac{1}{2}\partial_t(|U^n|^2_{b,1}) + \frac{\tau}{2}|\partial_t U^n|^2_{b,1} \\
&+ \langle (\delta_x b_1)U^n, \partial_t U^n\rangle + \sum_{i,j} b_{1i+\frac{1}{2},j}\delta_x U^n_{i+\frac{1}{2},j}\partial_t U^n_{i,j}h_xh_y \\
&+ \langle (\delta_y b_2)U^n, \partial_t U^n\rangle + \sum_{i,j} b_{2i,j+\frac{1}{2}}\delta_y U^n_{i,j+\frac{1}{2}}\partial_t U^n_{i,j}h_xh_y.
\end{aligned} \tag{2.28}$$

Proof: Using the equations (2.4) and (2.6), we get:

$$\begin{aligned}
\langle \delta_x(b_1U^{n,ux}), \partial_t U^n\rangle &= \sum_{i,j}(b_{1i+\frac{1}{2},j}H(b_{1i+\frac{1}{2},j})\delta_x U^n_{i+\frac{1}{2},j}(\partial_t U^n_{i+1,j} - \partial_t U^n_{i,j})h_xh_y \\
&+ \frac{1}{h_x}\sum_{i,j}(b_{1i+\frac{1}{2},j}U^n_{i+1,j} - b_{1i-\frac{1}{2},j}U^n_{i,j})\partial_t U^n_{i,j}h_xh_y \\
&= \frac{1}{\tau}\sum_{i,j} b^+_{1i+\frac{1}{2},j}\delta_x U^n_{i+\frac{1}{2},j}(\delta_x U^n_{i+\frac{1}{2},j} - \delta_x U^{n-1}_{i+\frac{1}{2},j})h_x^2h_y \\
&+ \langle (\delta_x b_1)U^n, \partial_t U^n\rangle + \sum_{i,j} b_{1i+\frac{1}{2},j}\delta_x U^n_{i+\frac{1}{2},j}\partial_t U^n_{i,j}h_xh_y.
\end{aligned} \tag{2.29}$$

In a similar way, using the equations (2.5) and (2.7), we have that:

$$\begin{aligned}
\langle \delta_y(b_2U^{n,uy}), \partial_t U^n\rangle = \frac{1}{\tau}\sum_{i,j} b^+_{2i,j+\frac{1}{2}}\delta_y U^n_{i,j+\frac{1}{2}}(\delta_y U^n_{i,j+\frac{1}{2}} - \delta_y U^{n-1}_{i,j+\frac{1}{2}})h_xh_y^2 + \\
\langle (\delta_y b_2)U^n, \partial_t U^n\rangle + \sum_{i,j} b_{2i,j+\frac{1}{2}}\delta_y U^n_{i,j+\frac{1}{2}}\partial_t U^n_{i,j}h_xh_y.
\end{aligned} \tag{2.30}$$

Now, one can obtain the equation (2.28) by applying (2.27) in the equations (2.29) and (2.30), and adding the two resulting equations.

Theorem 2.3.1. *Let $U^n = \{U^n_{i,j} \mid n \geq 0,\ 0 \leq i \leq J_x,\ 0 \leq j \leq J_y\}$ be the solution of the scheme* (2.15)-(2.21). *Then it holds that:*

$$\|U^n\|^2 \leq C\Bigg(|U^0|^2_{a^*,1} + |U^0|^2_{b,1} + \tau\|f^{\frac{1}{2}}\|^2 + \tau\sum_{k=2}^{n}\|f^k\|^2\Bigg). \tag{2.31}$$

Proof: Let $\||U^n\||^2_{a^*,b} = |U^n|^2_{a^*,1} + |U^n|^2_{b,1} + \frac{\tau}{2}(\|\partial_t U^n\|^2)$. Then multiplying both sides of the equation (2.22) by $\partial_t U^n_{i,j} h_x h_y$, summing over i, j, and using Lemmas (2.3.1) and (2.3.2) in the resulting equation lead to:

$$\begin{aligned}
\frac{1}{2}\partial_t(\||U^n\||^2_{a^*,b}) \quad &+ \quad \frac{\tau^2}{4}\|\partial_t^2 U^n\|^2 = -\frac{\tau}{2}|\partial_t U^n|^2_{a^*,1} - \frac{\tau}{2}|\partial_t U^n|^2_{b,1} - \|\partial_t U^n\|^2\rangle \\
&- \quad \sum_{i,j} b_{1i+\frac{1}{2},j}\delta_x U^n_{i+\frac{1}{2},j}\partial_t U^n_{i,j} h_x h_y - \langle(\delta_y b_2)U^n, \partial_t U^n\rangle - \langle cU^n, \partial_t U^n\rangle \\
&- \quad \sum_{i,j} b_{2i,j+\frac{1}{2}}\delta_y U^n_{i,j+\frac{1}{2}}\partial_t U^n_{i,j} h_x h_y + \langle f^n, \partial_t U^n\rangle + \langle \tilde{Q}^n_{1a}, \partial_t U^n\rangle \\
&+ \quad \langle \tilde{Q}^n_{1b}, \partial_t U^n\rangle + \langle \tilde{Q}^n_{2a}, \partial_t U^n\rangle + \langle \tilde{Q}^n_{2b}, \partial_t U^n - \langle(\delta_x b_1)U^n, \partial_t U^n\rangle
\end{aligned} \tag{2.32}$$

where $\tilde{Q}^n_{1a}$ and $\tilde{Q}^n_{2a}$ are:

$$\langle \tilde{Q}^n_{1a}, \partial_t U^n\rangle \quad = \quad \sum_{(x_i,y_j)\in\Gamma_h^1} (a^*_{1i+\frac{1}{2},j} h_x^{-2}\partial_t U^n_{i+1,j} + a^*_{1i-\frac{1}{2},j} h_x^{-2}\partial_t U^n_{i-1,j})(\tilde{U}^n_{i,j} - U^n_{i,j}) h_x h_y, \tag{2.33}$$

$$\langle \tilde{Q}^n_{2a}, \partial_t U^n\rangle \quad = \quad \sum_{(x_i,y_j)\in\Gamma_h^2} (a^*_{2i,j+\frac{1}{2}} h_y^{-2}\partial_t U^n_{i,j+1} + a^*_{2i,j-\frac{1}{2}} h_y^{-2}\partial_t U^n_{i,j-1})(\tilde{U}^n_{i,j} - U^n_{i,j}) h_x h_y. \tag{2.34}$$

Using the following well-known inequality:

$$ab < \eta a^2 + (4\eta)^{-1} b^2 \quad a, b \in \mathbb{R},\ \eta \geq 0, \tag{2.35}$$

and equation (2.25), we can derive that:

$$\begin{aligned}-\langle(\delta_x b_1)U^n, \partial_t U^n\rangle &- \langle(\delta_y b_2)U^n, \partial_t U^n\rangle - \langle cU^n, \partial_t U^n\rangle \\ &\leq C\|U^n\| + \eta\|\partial_t U^n\|^2 \\ &\leq C\||U^n\||^2_{a^*,b} + \eta\|\partial_t U^n\|^2. \qquad (2.36)\end{aligned}$$

Using (2.35), one can also derive that:

$$\begin{aligned}&- \sum_{i,j} b_{1i+\frac{1}{2},j}\delta_x U^n_{i+\frac{1}{2},j}\partial_t U^n_{i,j}h_x h_y \\ &- \sum_{i,j} b_{2i,j+\frac{1}{2}}\delta_y U^n_{i,j+\frac{1}{2}}\partial_t U^n_{i,j}h_x h_y + \langle f^n, \partial_t U^n\rangle \\ &\leq C\||U^n\||^2_{a^*,b} + \eta\|\partial_t U^n\|^2 + C\|f^n\|^2. \qquad (2.37)\end{aligned}$$

Inserting the equation (2.36) and the above inequality into (2.32), we obtain:

$$\begin{aligned}&\frac{1}{2}\partial_t(\||U^n\||^2_{a^*,b}) \leq -\frac{\tau}{2}|\partial_t U^n|^2_{a^*,1} - \|\partial_t U^n\|^2 + \langle\tilde{Q}^n_{1a}, \partial_t U^n\rangle + \langle\tilde{Q}^n_{1b}, \partial_t U^n\rangle \\ &+\langle\tilde{Q}^n_{2a}, \partial_t U^n\rangle + \langle\tilde{Q}^n_{2b}, \partial_t U^n\rangle + C\||U^n\||^2_{a^*,b} + \eta\|\partial_t U^n\|^2 + C\|f^n\|^2. \qquad (2.38)\end{aligned}$$

Now, we analyze the term $a^*_{1i+\frac{1}{2},j}h_x^{-2}\partial_t U^n_{i+1,j}(\tilde{U}^n_{i,j} - U^n_{i,j})h_x h_y$ in $\langle\tilde{Q}^n_{1a}, \partial_t U^n\rangle$ at the mesh point $(x_i, y_j) \in \Gamma^1_h$. Using (2.27), we get:

$$\begin{aligned}\partial_t U^n_{i+1,j}(\tilde{U}^n_{i,j} - U^n_{i,j}) &= \tau\partial_t U^n_{i+1,j}(\partial_t U^{n-1}_{i,j} - \partial_t U^n_{i,j}) \\ &= \frac{\tau}{2}(|\partial_t U^{n-1}_{i,j}|^2 - |\partial_t U^n_{i,j}|^2 - |\partial_t U^n_{i+1,j} - \partial_t U^{n-1}_{i,j}|^2 \\ &+ |\partial_t U^n_{i+1,j} - \partial_t U^n_{i,j}|^2) \qquad (x_i, y_j) \in \Gamma^1_h. \qquad (2.39)\end{aligned}$$

In similar way, we have:

$$\begin{aligned}\partial_t U^n_{i-1,j}(\tilde{U}^n_{i,j} - U^n_{i,j}) &= \tau \partial_t U^n_{i-1,j}(\partial_t U^{n-1}_{i,j} - \partial_t U^n_{i,j}) \\ &= \frac{\tau}{2}(|\partial_t U^{n-1}_{i,j}|^2 - |\partial_t U^n_{i,j}|^2 - |\partial_t U^n_{i-1,j} - \partial_t U^{n-1}_{i,j}|^2 \\ &+ |\partial_t U^n_{i-1,j} - \partial_t U^n_{i,j}|^2) \qquad (x_i, y_j) \in \Gamma^1_h. \end{aligned} \tag{2.40}$$

Inserting equations (2.39) and (2.40) into (2.33) yields:

$$\begin{aligned}\langle \tilde{Q}^n_{1a}, \partial_t U^n \rangle &\leq \sum_{(x_i,y_j)\in\Gamma^1_h} a^*_{1i+\frac{1}{2},j} h_x^{-2} \frac{\tau}{2}(|\partial_t U^{n-1}_{i,j}|^2 - |\partial_t U^n_{i,j}|^2) h_x h_y \\ &+ \sum_{(x_i,y_j)\in\Gamma^1_h} a^*_{1i+\frac{1}{2},j} h_x^{-2} \frac{\tau}{2} |\partial_t U^n_{i+1,j} - \partial_t U^n_{i,j}|^2 h_x h_y \\ &+ \sum_{(x_i,y_j)\in\Gamma^1_h} a^*_{1i-\frac{1}{2},j} h_x^{-2} \frac{\tau}{2}(|\partial_t U^{n-1}_{i,j}|^2 - |\partial_t U^n_{i,j}|^2) h_x h_y \\ &+ \sum_{(x_i,y_j)\in\Gamma^1_h} a^*_{1i-\frac{1}{2},j} h_x^{-2} \frac{\tau}{2} |\partial_t U^n_{i-1,j} - \partial_t U^n_{i,j}|^2 h_x h_y. \end{aligned} \tag{2.41}$$

Now, we define:

$$W^n_{1\Gamma^1_h} = \sum_{(x_i,y_j)\in\Gamma^1_h} \frac{\tau^2}{2}(a^*_{1i-\frac{1}{2},j} + a^*_{1i+\frac{1}{2},j}) h_x^{-2} |\partial_t U^n_{i,j}|^2 h_x h_y.$$

Then the equation (2.41) becomes:

$$\begin{aligned}\langle \tilde{Q}^n_{1a}, \partial_t U^n \rangle \leq & -\frac{W^n_{1\Gamma^1_h} - W^{n-1}_{1\Gamma^1_h}}{\tau} + \sum_{(x_i,y_j)\in\Gamma^1_h} a^*_{1i+\frac{1}{2},j} h_x^{-2} \frac{\tau}{2}(|\partial_t U^n_{i+1,j} - \partial_t U^n_{i,j}|^2) h_x h_y \\ &+ \sum_{(x_i,y_j)\in\Gamma^1_h} a^*_{1i-\frac{1}{2},j} h_x^{-2} \frac{\tau}{2}(|\partial_t U^n_{i-1,j} - \partial_t U^n_{i,j}|^2) h_x h_y. \end{aligned}$$

Using (2.27) in the above inequality, we obtain:

$$\begin{aligned}\langle \tilde{Q}^n_{1a}, \partial_t U^n \rangle \leq & -\frac{W^n_{1\Gamma^1_h} - W^{n-1}_{1\Gamma^1_h}}{\tau} + \sum_{(x_i,y_j)\in\Gamma^1_h} a^*_{1i+\frac{1}{2},j} h_x^{-2} \frac{\tau}{2}(|\partial_t U^n_{i+1,j} - \partial_t U^n_{i,j}|^2) h_x h_y \\ &+ \sum_{(x_i,y_j)\in\Gamma^1_h} a^*_{1i-\frac{1}{2},j} h_x^{-2} \frac{\tau}{2}(|\partial_t U^n_{i-1,j} - \partial_t U^n_{i,j}|^2) h_x h_y + 2\eta \|\partial_t U^n\|^2. \end{aligned} \tag{2.42}$$

In similar way, we can show that:

$$\langle \tilde{Q}^n_{2a}, \partial_t U^n\rangle \leq -\frac{W^n_{2\Gamma^2_h} - W^{n-1}_{2\Gamma^2_h}}{\tau} + \sum_{(x_i,y_j)\in\Gamma^2_h} a^*_{2i,j+\frac{1}{2}} h_x^{-2}\frac{\tau}{2}(|\partial_t U^n_{i,j+1} - \partial_t U^n_{i,j}|^2)h_x h_y$$
$$+ \sum_{(x_i,y_j)\in\Gamma^2_h} a^*_{2i,j-\frac{1}{2}} h_x^{-2}\frac{\tau}{2}(|\partial_t U^n_{i,j-1} - \partial_t U^n_{i,j}|^2)h_x h_y + 2\eta\|\partial_t U^n\|^2, \quad (2.43)$$

where

$$W^n_{1\Gamma^2_h} = \sum_{(x_i,y_j)\in\Gamma^2_h} \frac{\tau^2}{2}(a^*_{2i,j-\frac{1}{2}} + a^*_{2i,j+\frac{1}{2}})h_x^{-2}|\partial_t U^n_{i,j}|^2 h_x h_y. \quad (2.44)$$

Inserting (2.42) and (2.43) into (2.38) yields:

$$\begin{aligned}
\frac{1}{2}\partial_t(\||U^n\||^2_{a^*,b}) \leq & -\frac{\tau}{2}|\partial_t U^n|^2_{a^*,1} - \|\partial_t U^n\|^2 \\
& + \langle \tilde{Q}^n_{1b}, \partial_t U^n\rangle + \langle \tilde{Q}^n_{2b}, \partial_t U^n\rangle + C\||U^n\||^2_{a^*,b} + \eta\|\partial_t U^n\|^2 + C\|f^n\|^2 \\
& - \frac{W^n_{1\Gamma^1_h} - W^{n-1}_{1\Gamma^1_h}}{\tau} + \sum_{(x_i,y_j)\in\Gamma^1_h} a^*_{1i+\frac{1}{2},j} h_x^{-2}\frac{\tau}{2}(|\partial_t U^n_{i+1,j} - \partial_t U^n_{i,j}|^2)h_x h_y \\
& + \sum_{(x_i,y_j)\in\Gamma^1_h} a^*_{1i-\frac{1}{2},j} h_x^{-2}\frac{\tau}{2}(|\partial_t U^n_{i-1,j} - \partial_t U^n_{i,j}|^2)h_x h_y + 2\eta\|\partial_t U^n\|^2 \\
& - \frac{W^n_{2\Gamma^2_h} - W^{n-1}_{2\Gamma^2_h}}{\tau} + \sum_{(x_i,y_j)\in\Gamma^2_h} a^*_{2i,j+\frac{1}{2}} h_x^{-2}\frac{\tau}{2}(|\partial_t U^n_{i,j+1} - \partial_t U^n_{i,j}|^2)h_x h_y \\
& + \sum_{(x_i,y_j)\in\Gamma^2_h} a^*_{2i,j-\frac{1}{2}} h_x^{-2}\frac{\tau}{2}(|\partial_t U^n_{i,j-1} - \partial_t U^n_{i,j}|^2)h_x h_y + 2\eta\|\partial_t U^n\|^2.
\end{aligned}$$

In the above inequality the following term is negative:

$$\begin{aligned}
-\frac{\tau}{2}|\partial_t U^n|^2_{a^*,1} &+ \sum_{(x_i,y_j)\in\Gamma_h^1} a^*_{1i+\frac{1}{2},j} h_x^{-2}\frac{\tau}{2}(|\partial_t U^n_{i+1,j} - \partial_t U^n_{i,j}|^2) h_x h_y \\
&+ \sum_{(x_i,y_j)\in\Gamma_h^1} a^*_{1i-\frac{1}{2},j} h_x^{-2}\frac{\tau}{2}(|\partial_t U^n_{i-1,j} - \partial_t U^n_{i,j}|^2) h_x h_y \\
&+ \sum_{(x_i,y_j)\in\Gamma_h^2} a^*_{2i,j+\frac{1}{2}} h_x^{-2}\frac{\tau}{2}(|\partial_t U^n_{i,j+1} - \partial_t U^n_{i,j}|^2) h_x h_y \\
&+ \sum_{(x_i,y_j)\in\Gamma_h^2} a^*_{2i,j-\frac{1}{2}} h_x^{-2}\frac{\tau}{2}(|\partial_t U^n_{i,j-1} - \partial_t U^n_{i,j}|^2) h_x h_y \le 0.
\end{aligned}$$

Therefore, we obtain:

$$\begin{aligned}
\frac{1}{2}\partial_t(\||U^n\||^2_{a^*,b}) &\le -\|\partial_t U^n\|^2 - \frac{W^n_{1\Gamma_h^1} - W^{n-1}_{1\Gamma_h^1}}{\tau} - \frac{W^n_{2\Gamma_h^2} - W^{n-1}_{2\Gamma_h^2}}{\tau}\rangle \\
&+ \langle \tilde{Q}^n_{1b} + \tilde{Q}^n_{2b}, \partial_t U^n\rangle + C\||U^n\||^2_{a^*,b} \\
&+ \eta\|\partial_t U^n\|^2 + C\|f^n\|^2 + 4\eta\|\partial_t U^n\|^2. \qquad (2.45)
\end{aligned}$$

Using (2.27), we can derive that:

$$\langle \tilde{Q}^n_{1b} + \tilde{Q}^n_{2b}, \partial_t U^n\rangle \le C(W^n_{1\Gamma_h^1} + W^n_{2\Gamma_h^2}) + \eta\|\partial_t U^n\|^2. \qquad (2.46)$$

Let $E^n = \frac{1}{2}(\||U^n\||^2_{a^*,b}) + W^n_{1\Gamma_h^1} + W^n_{2\Gamma_h^2}$. Then, using (2.46), (2.45) becomes:

$$\partial_t(E^n) \le -\|\partial_t U^n\|^2 + CE^n + 6\eta\|\partial_t U^n\|^2 + C\|f^n\|^2. \qquad (2.47)$$

Assuming η is small, then we get:

$$\partial_t(E^n) \le CE^n + C\|f^n\|^2. \qquad (2.48)$$

Summing n from 2 to n leads to:

$$E^n \le E^1 + C\tau\sum_2^n E^k + C\tau\sum_2^n \|f^k\|^2. \qquad (2.49)$$

By the discrete Gronwall inequality, we have:

$$E^n \leq \exp(CT)\Bigg(E^1 + C\tau \sum_{2}^{n} \|f^k\|^2\Bigg). \tag{2.50}$$

Now, we complete the proof by approximating E^1 in the following. Because the value of $U_{i,j}^1$ in the SCEIDD scheme is computed by the Crank-Nicolson scheme, we have:

$$\frac{1}{2}\|\partial_t U^1\|^2 + (2 - \frac{1}{2})\|\partial_t U^1\|^2 = \langle L_h(U^1 + U^0), \partial_t U^1\rangle + \langle f^{\frac{1}{2}}, \partial_t U^1\rangle. \tag{2.51}$$

Because:

$$\langle \delta^2_{x,a_1^*} U^1 + \delta^2_{x,a_1^*} U^0, \partial_t U^1\rangle \;=\; -\sum_{i,j} \frac{1}{\tau} a^*_{1i+\frac{1}{2},j}(\delta_x U^1_{i+\frac{1}{2},j} + \delta_x U^0_{i+\frac{1}{2},j})(\delta_x U^1_{i+\frac{1}{2},j} - U^0_{i+\frac{1}{2},j}),$$

it follows that:

$$\langle \delta^2_{x,a_1^*}(U^1 + U^0) + \delta^2_{y,a_2^*}(U^1 + U^0), \partial_t U^1\rangle \;=\; -\frac{1}{\tau}(|U^1|^2_{a^*,1} - |U^0|^2_{a^*,1}). \tag{2.52}$$

In similar way, using (2.35), it can be derived that:

$$\begin{aligned}\langle -\delta_x(b_1 U^{1,ux} + b_1 U^{0,ux}) \;&-\; \delta_y(b_2 U^{1,uy} + b_2 U^{0,uy}), \partial_t U^1\rangle \leq -\frac{|U^1|^2_{b,1} - |U^0|^2_{b,1}}{\tau}\\ &+\; C(|U^1|^2_{a^*,1} + |U^0|^2_{a^*,1}) + \eta\|\partial_t U^1\|^2.\end{aligned}$$

Using the above inequality, (2.52), part 3 of Lemma 2.3.1, and (2.35) lead to:

$$\langle L_h(U^1 + U^0), \partial_t U^1\rangle \leq -\partial t(|U^1|^2_{a^*,1} + |U^1|^2_{b,1}) + C(|U^1|^2_{a^*,1} + |U^0|^2_{a^*,1}) + \eta\|\partial_t U^1\|^2.$$

Inserting the above inequality in (2.51), applying (2.35) to the last term $(f^{\frac{1}{2}}, \partial_t U^1\rangle)$ and eliminating the negative term yield:

$$\partial t(|U^1|^2_{a^*,1} + |U^1|^2_{b,1}) + \frac{1}{2}\|\partial_t U^1\|^2 \leq C(|U^1|^2_{a^*,1} + |U^0|^2_{a^*,1} + \|f^{\frac{1}{2}}\|^2),$$

and it follows that:

$$|U^1|^2_{a^*,1} + |U^1|^2_{b,1} + \frac{\tau}{2}\|\partial_t U^1\|^2 \leq C\Bigg(|U^0|^2_{a^*,1} + |U^0|^2_{b,1} + \tau\|f^{\frac{1}{2}}\|^2\Bigg). \tag{2.53}$$

Using the above equation, we can get:

$$E^1 \leq C\Bigg(|U^0|^2_{a^*,1} + |U^0|^2_{b,1} + \tau\|f^{\frac{1}{2}}\|^2\Bigg). \tag{2.54}$$

Inserting the equation (2.54) into (2.50) yields:

$$E^n \quad \leq \quad \exp(CT)\Bigg(C\Bigg(|U^0|^2_{a^*,1} + |U^0|^2_{b,1} + \tau\|f^{\frac{1}{2}}\|^2\Bigg) + C\tau\sum_{k=2}^{n}\|f^k\|^2\Bigg).$$

Using part 3 of Lemma 3.1 yields:

$$\|U^n\|^2 \quad \leq \quad C\Bigg(\Bigg(|U^0|^2_{a^*,1} + |U^0|^2_{b,1} + \tau\|f^{\frac{1}{2}}\|^2\Bigg) + \tau\sum_{k=2}^{n}\|f^k\|^2\Bigg).$$

2.4 Error Analysis

In this section, we perform the error analysis for the SCEIDD scheme, and we prove that it has second-order accuracy in both time and space. Let $\tilde{\Psi}^n_{i,j}$ and $\Psi^n_{i,j}$ be the local truncation error of explicit scheme (2.15) and the local truncation error of implicit schemes (2.16)-(2.18), respectively. These truncation errors, $\tilde{\Psi}^n_{i,j}$ and $\Psi^n_{i,j}$, satisfy the following equations:

$$u^n_{i,j} = 2u^{n-1}_{i,j} - u^{n-2}_{i,j} - \tilde{\Psi}^n_{i,j}, \qquad \text{on } \Gamma_h, \tag{2.55}$$

$$\begin{cases} L_\tau u^n_{i,j} = L_h u^n_{i,j} - \Psi^n_{i,j} & \text{on } \Omega_h \backslash \{\Gamma_h \cup \partial\Omega_h\}, \\ \quad u^n_{i,j} = u^n_{i.j} & \text{on } \Gamma_h, \end{cases} \tag{2.56}$$

$$\begin{cases} L_\tau u^n_{i,j} = L_h u^n_{i,j} - \Psi^n_{i,j} & \text{on } \Gamma_h \backslash \Gamma^3_h, \\ \quad u^n_{i,j} = u^n_{i.j} & \text{on } \Gamma^3_h, \end{cases} \tag{2.57}$$

$$L_\tau u^n_{i,j} = L_h u^n_{i,j} - \Psi^n_{i,j} \qquad \text{on } \Gamma^3_h, \tag{2.58}$$

$$\partial_t u^1_{i,j} = L_h \frac{u^1_{i,j} + u^0_{i,j}}{2} - \Psi^{\frac{1}{2}}_{i,j} \quad \text{on } \Omega_h. \tag{2.59}$$

In the following Lemma, we estimate the local truncation errors.

Lemma 2.4.1. *Let $\Psi^n_{i,j}$, $\tilde{\Psi}^n_{i,j}$ and $\Psi^{\frac{1}{2}}_{i,j}$ be the local truncation errors of the SCEIDD scheme. Then we have that:*

$$\begin{aligned} |\Psi^{\frac{1}{2}}_{i,j}| \leq & \ C(\tau^2 + h_x^2 + h_y^2), \qquad && (x_i, y_j) \in \Omega_h, \\ |\Psi^n_{i,j}| \leq & \ C(\tau^2 + h_x^2 + h_y^2), \qquad && (x_i, y_j) \in \Omega_h, \\ |\tilde{\Psi}^n_{i,j}| \leq & \ C\tau^2, \qquad && (x_i, y_j) \in \Gamma_h. \end{aligned} \tag{2.60}$$

Proof: By the Taylor expansion, we get:

$$\begin{aligned} L_h u_{i,j}^n &= \delta_{x,a_1^*}^2 u_{i,j}^n + \delta_{y,a_2^*}^2 u_{i,j}^n - \delta_x (b_1 u^{n,ux})_{i,j} - \delta_y (b_2 u^{n,uy})_{i,j} - c_{i,j} u_{i,j}^n \\ &= L(u)|_{i,j}^n + O(h_x^2 + h_y^2), \qquad (2.61) \\ L_\tau u_{i,j}^n &= \partial_t u_{i,j}^n + \frac{1}{2\tau}(u_{i,j}^n - 2u_{i,j}^{n-1} + u_{i,j}^{n-2}) = u_t|_{i,j}^n + O(\tau^2). \qquad (2.62) \end{aligned}$$

From (2.1) and the above equation, we have:

$$\begin{aligned} \Psi_{i,j}^n = L_\tau u_{i,j}^n - L_h u_{i,j}^n - f_{i,j}^n &= u_t|_{i,j}^n - L(u)|_{i,j}^n - f_{i,j}^n + O(\tau^2) + O(h_x^2 + h_y^2) \\ &= O(\tau^2 + h_x^2 + h_y^2). \qquad (2.63) \end{aligned}$$

Therefore, $|\Psi_{i,j}^n| \leq C(\tau^2 + h_x^2 + h_y^2)$. From the equation (2.15):

$$\tilde{\Psi}_{i,j}^n = u_{i,j}^n - (2u_{i,j}^{n-1} - u_{i,j}^{n-2}) = O(\tau^2). \qquad (2.64)$$

Thus, $|\tilde{\Psi}_{i,j}^n| \leq C(\tau^2)(n \geq 2)$. Since the Crank-Nicolson scheme has second-order accuracy in time and space, therefore the proof of the lemma is complete. Now, we analyze the error of the SCEIDD scheme which is defined as the difference between the approximated and the exact solution. Let $e_{i,j}^n = U_{i,j}^n - u_{i,j}^n$ and $\tilde{e}_{i,j}^n = \tilde{U}_{i,j}^n - u_{i,j}^n$. Then from the equations (2.15)-(2.18) and (2.55)-(2.59), we have:

$$\tilde{e}_{i,j}^n = 2e_{i,j}^{n-1} - e_{i,j}^{n-2} - \tilde{\Psi}_{i,j}^n, \quad \text{on } \Gamma_h, \qquad (2.65)$$

$$\begin{cases} L_\tau e_{i,j}^n = L_h e_{i,j}^n - \Psi_{i,j}^n & \text{on } \Omega_h \backslash \{\Gamma_h \cup \partial\Omega_h\}, \\ e_{i,j}^n = \tilde{e}_{i,j}^n & \text{on } \Gamma_h, \end{cases} \qquad (2.66)$$

$$\begin{cases} L_\tau e^n_{i,j} = L_h e^n_{i,j} - \Psi^n_{i,j} & \text{on } \Gamma_h \backslash \Gamma^3_h, \\ \\ \quad e^n_{i,j} = \tilde{e}^n_{i,j} & \text{on } \Gamma^3_h, \end{cases} \tag{2.67}$$

$$L_\tau e^n_{i,j} = L_h e^n_{i,j} - \Psi^n_{i,j} \qquad \text{on } \Gamma^3_h, \tag{2.68}$$

$$\partial_t e^1_{i,j} = L_h \frac{e^1_{i,j} + e^0_{i,j}}{2} - \Psi^{\frac{1}{2}}_{i,j} \quad \text{on } \Omega_h, \tag{2.69}$$

where $e^n_{i,j} = 0, (x_i, y_j) \in \partial\Omega_h$ and $e^0_{i,j} = 0$. Similar to the stability analysis, one can show that:

$$\begin{aligned} \|e^n\|^2 \le C\exp(CT)\Bigg(|e^0|^2_{a^*,1} &+ |e^0|^2_{b,1} + \tau\|\Psi^{\frac{1}{2}}\|^2 \\ &+ \tau\sum_{k=2}^{n}\|\Psi^k\|^2 + \tau\sum_{k=2}^{n}|||\tilde{\Psi}^k|||^2\Bigg), \end{aligned} \tag{2.70}$$

where

$$\|\tilde{\Psi}^k\||^2 = \sum_{(x_i,y_j)\in\Gamma^1_h} Ch_x^{-4}|\tilde{\Psi}^k_{i,j}|^2 h_x h_y + \sum_{(x_i,y_j)\in\Gamma^2_h} Ch_y^{-4}|\tilde{\Psi}^k_{i,j}|^2 h_x h_y.$$

Let $\frac{\tau}{h_x^2} = \lambda_x$ and $\frac{\tau}{h_y^2} = \lambda_y$. Then, using (2.64), we obtain that:

$$\|e^n\| \le C(\tau^2 + h_x^2 + h_y^2).$$

We proved that the SCEIDD scheme has second-order accuracy in time and space which is stated in the following theorem.

Theorem 2.4.1. *(Convergence) If the solution of problem* (2.1) *is sufficiently smooth, then numerical solution $U_{i,j}^n$ generated by the SCEIDD scheme* (2.15)-(2.21) *converges to the exact solution with error of $O\left(\tau^2 + h_x^2 + h_y^2\right)$ in L^2 norm.*

2.5 Numerical Experiment

In this section, we present numerical experiments for two-dimensional convection-diffusion equations to verify the theoretical results obtained in two previous sections. Three different numerical experiments have been performed to study the stability and accuracy of the SCEIDD scheme in the sense of L^2 error norm:

$$\|e^n\|_2 = \sqrt{\sum_i \sum_j (u(x_i, y_j, t_n) - U^n(x_i, y_j))^2 h_x h_y},$$

where $u(x_i, y_j, t_n)$ and $U^n(x_i, y_j)$ are the exact solution and the approximated solution, respectively.

Example 1. In this example, we consider the transport of a rotating Gaussian pulse in a two-dimensional square domain which has been widely used to test numerical schemes developed for convection-diffusion equations. For this example, the variable velocity field is given by $b_1 = -4y$, $b_2 = 4x$, the diffusion coefficients are taken as $a_1(x, y) = a_2(x, y) = D$ where D is a positive constant, the reaction coefficient is given by $c(x, y) = 0$, and the initial configuration of the Gaussian pulse is given by:

$$u_0(x,y) = \exp\left(-\frac{(x-x_c)^2+(y-y_c)^2}{2\sigma^2}\right). \tag{2.71}$$

Here, (x_c, y_c) and σ are the location of the center and standard deviation, respectively. For this example, the exact solution is given by:

$$u(x,y,t) = \frac{2\sigma^2}{2\sigma^2+4Dt}\exp\left(-\frac{(x^*-x_c)^2+(y^*-y_c)^2}{2\sigma^2+4Dt}\right), \tag{2.72}$$

where

$$x^* = (\cos 4t)x + (\sin 4t)y, \quad y^* = -(\sin 4t)x + (\cos 4t)y. \tag{2.73}$$

The boundary condition and source term f are decided by the above exact solution, and additional data are given as follows: $\Omega = [-0.5, 0.5] \times [-0.5, 0.5]$, $(x_c, y_c) = (0, 0)$, $\sigma = 0.0447$.

To show that the SCEIDD scheme is unconditionally stable, we apply it to solve the convection-diffusion equation (2.1) with the exact solution and other parameters provided in Example 1, where the computational domain is decomposed into 2×2, 6×1, and 10×1 subdomains, as shown in Figure 2.2. Moreover, the numerical experiments are carried out for different time steps, different diffusion coefficients, D=5×10^{-3}, D=1×10^{-2}, while keeping the spatial step, $h = \frac{1}{200}$, and the final time, $T = 0.5$, fixed. From the numerical results, displayed in Tables 2.2 and 2.1, it is evident that, as we increase the time step while keeping the spatial step size

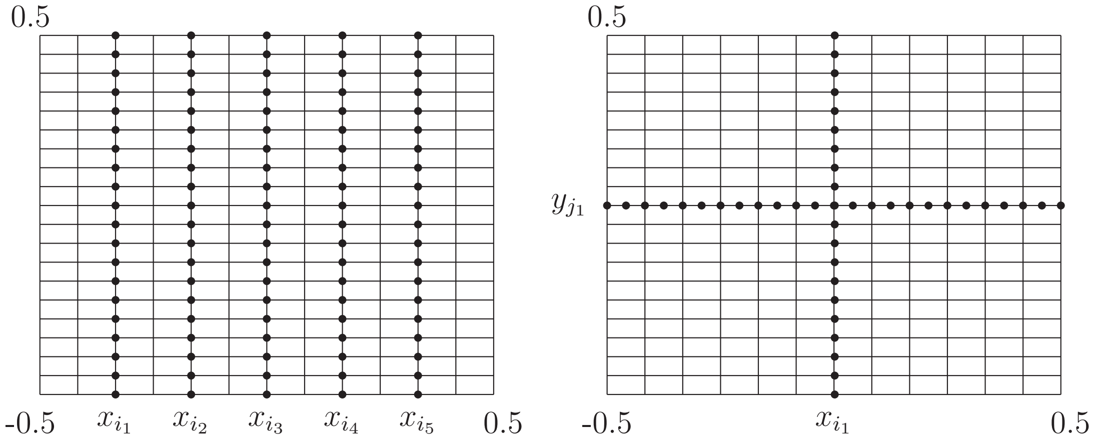


Figure 2.2: 6×1 subdomains (left) and 2×2 subdomains (right).

fixed, the error of the SCEIDD scheme remain relatively small. For instance, from Table 2.2, for very large time step size $\tau = \frac{1}{10}$, the error of the SCEIDD scheme is 7.3575×10^{-4}, 9.2359×10^{-4} and 5.5947×10^{-3} when the computational domain is decomposed into 2×2, 6×1 and 10×1 subdomains, respectively. These results verify that the SCEIDD scheme is unconditionally stable.

To find the convergence rate of the SCEIDD scheme in space, we let $T = 0.5$, $\tau = \frac{h}{30}$ and apply the SCEIDD scheme with various spatial steps, $\frac{1}{50}$, $\frac{1}{250}$, $\frac{1}{350}$, and $\frac{1}{550}$ to approximate the solution of the problem (2.1) with parameters provided in Example 1. Moreover, the simulation is performed for different diffusion coefficients, $D = 5 \times 10^{-3}$ and $D = 1 \times 10^{-2}$, and for different number of subdomains. From Tables

Table 2.1: Error by SCEIDD at $T = 0.5$, with D=1e-2 and $h = \frac{1}{200}$.

SD	τ	1/800	1/200	1/50	1/10
2×2	Error	7.6909e-05	7.5314e-05	8.3839e-04	1.1021e-03
6×1	Error	7.6901e-05	7.5267e-05	8.2473e-04	3.5690e-03
10×1	Error	7.6890e-05	7.5108e-05	8.5775e-04	6.6340e-03

Table 2.2: Error by SCEIDD at $T = 0.5$, with D=5e-3 and $h = \frac{1}{200}$.

SD	τ	1/800	1/200	1/50	1/10
2×2	Error	2.0570e-04	2.0436e-04	2.0102e-04	7.3575e-04
6×1	Error	2.0567e-04	2.0398e-04	1.9691e-04	9.2359e-04
10×1	Error	2.0505e-04	1.9453e-04	2.6569e-04	5.5947e-03

Table 2.3: Ratio in h by SCEIDD at $T = 0.5$, with D=1e-2 and $\tau = \frac{h}{30}$.

SD	h	1/150	1/250	1/350	1/550
6×1	Error	1.7395e-04	6.5501e-05	3.4209e-05	1.4050e-05
	Ratio in space		1.912	1.9306	1.9688
10×1	Error	1.7866e-04	6.8657e-05	3.6105e-05	1.4933e-05
	Ratio in space		1.8722	1.9101	1.9533
2×2	Error	1.7482e-04	6.6332e-05	3.4692e-05	1.4275e-05
	Ratio in space		1.8971	1.9263	1.9647

Table 2.4: Ratio in h by SCEIDD at $T = 0.5$, with D=5e-3 and $\tau = \frac{h}{30}$.

SD	h	1/150	1/250	1/350	1/550
6×1	Error	3.3896e-04	1.2997e-04	6.8343e-05	2.8199e-05
	Ratio in space		1.8766	1.9102	1.9586
10×1	Error	3.4833e-04	1.3599e-04	7.1938e-05	2.9799e-05
	Ratio in space		1.8413	1.8925	1.9499
2×2	Error	3.4713e-04	1.3372e-04	7.0556e-05	2.9209e-05
	Ratio in space		1.8675	1.9001	1.9512

Table 2.5: Ratio in τ by SCEIDD at $T = 0.5$, with D=1e-2 and $\tau = \frac{h}{20}$.

	SD	τ	1/3000	1/5000	1/7000	1/11000
SCEIDD	6×1	Error	9.8361e-05	3.7583e-05	1.9761e-05	8.1976e-06
		Ratio in time		1.8834	1.9105	1.9467
	10×1	Error	9.9266e-05	3.8147e-05	2.0166e-05	8.4056e-06
		Ratio in time		1.8722	1.8945	1.9361
	2×2	Error	9.8734e-05	3.7809e-05	1.9920e-05	8.2857e-06
		Ratio in time		1.8791	1.9045	1.9408
EIPCMU2D		Error	7.6103e-04	4.5872e-04	3.2753e-04	2.0057e-04
	(2×2)	Ratio in time		0.9910	1.0012	1.0850

Table 2.6: Ratio in τ by SCEIDD at $T = 0.5$, with D=5e-3, $\tau = \frac{h}{20}$.

	SD	τ	1/3000	1/5000	1/7000	1/11000
SCEIDD	6×1	Error	3.4855e-04	1.3603e-04	7.1971e-05	2.9918e-05
		Ratio in time		1.8419	1.8920	1.9421
	10×1	Error	3.5688e-04	1.4015e-04	7.4535e-05	3.1152e-05
		Ratio in time		1.8298	1.8766	1.9301
	2×2	Error	3.5210e-04	1.3771e-04	7.2935e-05	3.0394e-05
		Ratio in time		1.8377	1.8890	1.9366
EIPCMU2D		Error	8.2265e-04	4.9784e-04	3.5572e-04	2.2416e-04
	(2×2)	Ratio in time		0.9832	0.9990	1.0217

2.4 and 2.3, we observe that the SCEIDD scheme has second-order convergence rate in space. These numerical results are in agreement with the theoretical results obtained in the Theorem 2.4.1. Now, we test the accuracy of the SCEIDD scheme in time and compare its accuracy with the accuracy of EIPCMU2D scheme [113] which is recently developed for the parallel approximation of convection-diffusion equations. We let $T = 0.5$, $h = 20\tau$ and apply the SCEIDD and EIPCMU2D scheme with various time steps to solve the problem (2.1) with parameters provided in Example 1. Again, the simulation is performed for different diffusion coefficients, $D = 5 \times 10^{-3}$ and $D = 1 \times 10^{-2}$, and for the different number of subdomains. The numerical results are presented in Tables 2.6 and 2.5. From these Tables, we observe that SCEIDD scheme is second-order in time while EIPCMU2D scheme is only first order in time.

Example 2. For this example, the variable velocity field is given by $b_1 = y$, $b_2 = 0$, the diffusion coefficients are taken as $a_1(x, y) = a_2(x, y) = D$ where D is a positive constant, the reaction coefficient is given by $c(x, y) = 0$, and the exact solution is given by:

$$u_(x, y, t) = \exp(-t)\sin(x)\cos(y). \tag{2.74}$$

In this example, the computational domain is taken as $\Omega = [0, 1] \times [0, 1]$, and the initial value $u_0(x, y)$, the boundary condition and the source term f are decided by the above exact solution.

Numerical results for this example, presented in Table 2.7, are obtained at final time $T = 0.5$, for $D = 5 \times 10^{-3}$, using different time steps and for different number of subdomains while keeping the spatial step fixed, $h = \frac{1}{200}$. From Table 2.7, it is evident that, for a large time step, the SCEIDD scheme maintain its accuracy when the computational domain is decomposed into many subdomains. For instance, for very large time step $\tau = \frac{1}{10}$, the error of the SCEIDD scheme is 2.5539×10^{-2} when the computational domain is decomposed into 200 subdomains. These results show that the SCEIDD scheme not only is unconditionally stable but also maintain its accuracy as the number of subdomains increases.

Now, we test the accuracy of the SCEIDD scheme in space for Example 2. In this regard, we present the numerical results in Table 2.8. These numerical results are obtained at final time $T = 0.5$, for $D = 5 \times 10^{-3}$, using different spatial steps and for the different number of subdomains while keeping the time step fixed, $\tau = \frac{h}{30}$. From these numerical results, we observe the SCEIDD scheme has second-order convergence rate in space. We also note that the SCEIDD scheme maintain its second-order convergence rate in space as the number of subdomains increases.

Finally, we investigate the accuracy of the SCEIDD scheme in time for Example 2. In this regard, the numerical results are displayed in Table 2.9. These numerical results are obtained at final time $T = 0.5$, for $D = 5 \times 10^{-3}$, using different time steps and for the different number of subdomains while keeping the spatial step

Table 2.7: Error by SCEIDD at $T = 0.5$, with $h = \frac{1}{200}$ and variable τ.

SD	τ	1/800	1/200	1/50	1/10
2×2	Error	3.7970e-05	5.6152e-05	4.3802e-04	5.6764e-03
4×4	Error	3.9488e-05	8.4541e-05	8.7937e-04	1.0495e-02
6×6	Error	4.1168e-05	1.0127e-04	9.8923e-04	1.4327e-02
10×10	Error	4.3693e-05	1.6476e-04	2.1118e-03	1.8745e-02
20×20	Error	5.1037e-05	2.9811e-04	4.1097e-03	2.5539e-02

fixed, $h = 20\tau$. From these results, we note that the SCEIDD scheme is second-order in time. Furthermore, it is noted that the SCEIDD scheme maintains its second-order accuracy in time as the number of subdomain increases.

Example 3. We consider the moving sharp front problem in a two-dimensional square domain. In this example, the velocity field is given by $b_1 = 1$, $b_2 = 1$, the diffusion coefficients are taken as $a_1 = a_2 = 1e - 3$, the reaction coefficient is given by $c(x, y) = 0$, the initial value is given by:

$$u_0(x,y) = \begin{cases} 1, & 0 \le x, y \le 0.2 \\ 0, & \text{Otherwise}, \end{cases} \tag{2.75}$$

and the boundary conditions are given by:

$$u(0, y, t) = 1,\ y \in [0, 1], \quad u(x, 0, t) = 1,\ x \in [0, 1], \quad t \in (0, T],$$

Table 2.8: Ratio in h by SCEIDD at $T = 0.5$, with D=5e-3 and $\tau = \frac{h}{30}$.

SD	h	1/150	1/250	1/350	1/550
2x2	Error	5.9566e-05	2.3982e-05	1.2701e-05	5.2897e-06
	Ratio in space		1.7810	1.8890	1.9380
4x4	Error	5.9823e-05	2.4184e-05	1.2834e-05	5.3523e-06
	Ratio in space		1.7730	1.8830	1.9350
6x6	Error	6.0057e-05	2.4328e-05	1.2928e-05	5.4067e-06
	Ratio in space		1.7690	1.8791	1.9287
10x10	Error	6.0387e-05	2.4562e-05	1.3091e-05	5.4877e-06
	Ratio in space		1.7610	1.8702	1.9236
20x20	Error	6.1055e-05	2.4853e-05	1.3291e-05	5.5753e-06
	Ratio in space		1.7595	1.8602	1.9220

Table 2.9: Ratio in τ by SCEIDD at $T = 0.5$, with D=5e-3 and $\tau = \frac{h}{20}$.

	SD	τ	1/3000	1/5000	1/7000	1/11000
SCEIDD	2x2	Error	6.0368e-05	2.4450e-05	1.2993e-05	5.4358e-06
		Ratio in time	0	1.7693	1.8789	1.9280
	4x4	Error	6.0599e-05	2.4648e-05	1.3128e-05	5.5019e-06
		Ratio in time	0	1.7610	1.8722	1.9241
	6x6	Error	6.0825e-05	2.4802e-05	1.3250e-05	5.5708e-06
		Ratio in time	0	1.7561	1.8633	1.9170
	10x10	Error	6.1077e-05	2.4994e-05	1.3384e-05	5.6365e-06
		Ratio in time	0	1.7491	1.8562	1.9134
	20x20	Error	6.1267e-05	2.5108e-05	1.3469e-05	5.6795e-06
		Ratio in time	0	1.7463	1.8510	1.9105
EIPCMU2D	20×20	Error	2.6855e-04	1.7476e-04	1.2790e-04	8.2670e-05
		Ratio in time		0.8410	0.9278	0.9655

$$\frac{\partial u(1,y,t)}{\partial x} = 0,\ y \in [0,1], \qquad \frac{\partial u(x,1,t)}{\partial y} = 0,\ x \in [0,1], \quad t \in (0,T].$$

This example involves the propagation of a sharp front through the computational domain. Therefore, the numerical simulation of this example is difficult and challenging. To test the accuracy of the SCEIDD scheme for this example, we apply it to simulate the propagation of sharp front using $\tau = h_x = h_y = \frac{1}{150}$ while the computational domain is decomposed into 4 blocks. The surface plot of approximated solutions at $t = 0.1$ and $t = 0.2$ are displayed in Figures 2.3 and 2.4. From these figures, it is evident that the SCEIDD scheme simulates the propagation of a sharp front with good accuracy. Moreover, we compare the solution curves of three different numerical schemes. Figure 2.5 shows the solution curves of SCEIDD, UPWIND, EIPCMU2D and the reference solution at $t = 0.2$ on a sectional plane at $y = 0.5$. It is clear that the SCEIDD scheme has good accuracy compared to the UPWIND and EIPCMU2D schemes.

2.6 Conclusion

We developed a second-order corrected-explicit-implicit domain decomposition scheme (SCEIDD) for the parallel approximation of convection-diffusion equations over multi-block subdomains in two dimensions. This scheme is unconditionally stable, and it is second-order accurate in time as well as space. The proposed SCEIDD

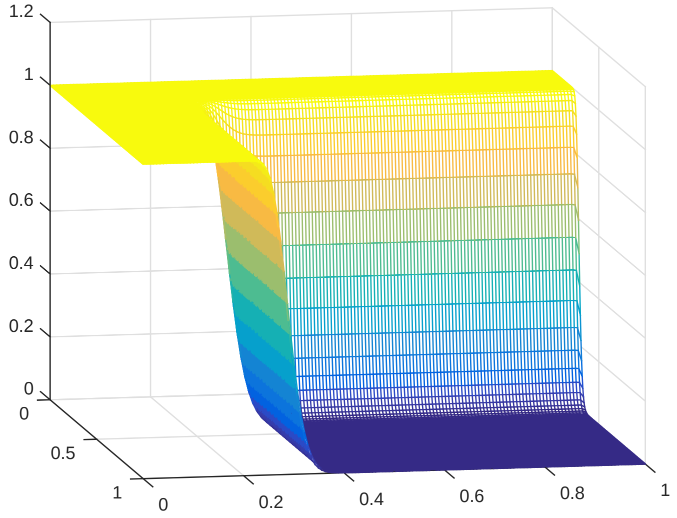


Figure 2.3: Sharp font approximated by our scheme SCEIDD at t=0.1.

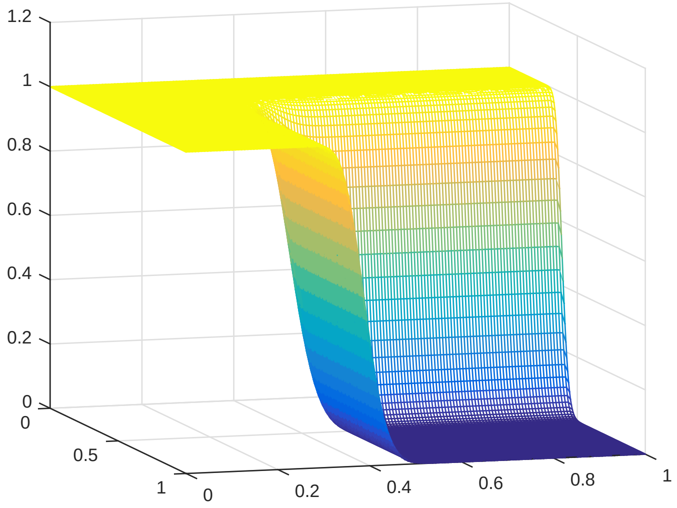


Figure 2.4: Sharp font approximated by our scheme SCEIDD at t=0.2.

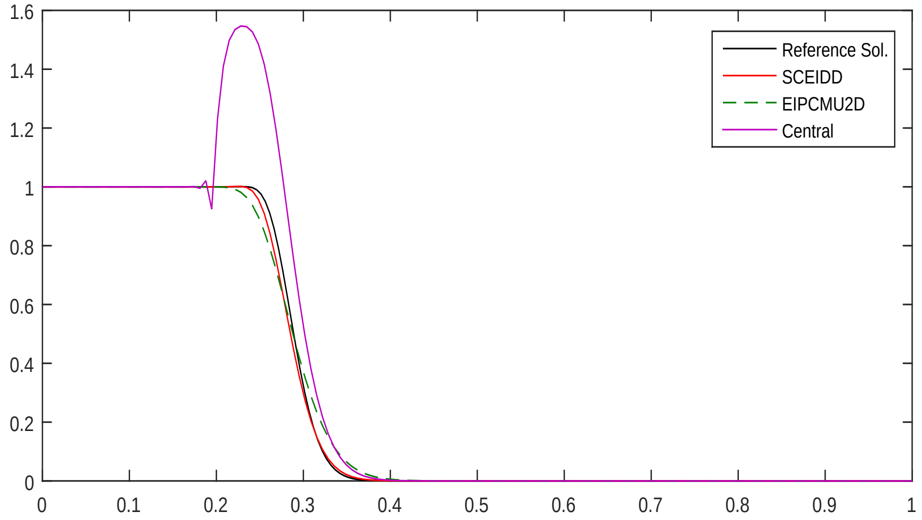


Figure 2.5: Computed moving sharp front with different schemes.

scheme maintains the advantages of CEIDD schemes, including good parallelism, the localization of communication, the flexibility of domain partitioning. Moreover, this scheme is designed over non-overlapping subdomains with simpler structures. Therefore, less effort is needed for numerical simulation. We also performed the stability and convergence analysis for the SCEIDD scheme, and we proved that it is second-order in time and space. Further, we carried out numerical experiments for three different examples to show that the developed SCEIDD scheme approximates the solutions of convection-diffusion equations with good accuracy and high efficiency. Moreover, in all the experiments the SCEIDD scheme is compared with the EIPCMU2D scheme [113] which is first-order in time and developed recently for the parallel approximation of the convection-diffusion equations.

3 Power Optimization of Wind Turbines

3.1 Introduction

In this chapter, we focus on analyzing, designing and power calculation of a single wind turbine as well as modeling of the airflow in a wind farm for the total power optimization. In the case of a single wind turbine, we address the following questions:

1. how much power can be generated?
2. what is the optimal design?
3. how to calculate the power of a given rotor?

In the case of multiple wind turbines, we focus on:

1. modeling the airflow to optimize the total power while considering wake interactions among wind turbines.

2. finding the global optimal operating points of multiple wind turbines by considering the wake effect on the power generated by wind turbines.

In the following section, we derive a set of equations that incorporate the physical parameters influencing the power generation of a single wind turbine.

3.2 Power Production

How much energy is in the wind and how much of the wind's energy can be converted into useful electrical energy? In 1919, Albert Betz theoretically determined the maximum amount of kinetic energy that can be extracted by a wind turbine. He considered a wind turbine as a circular disc through which the wind flows with an incoming wind speed of U_0 and leaves the rotor plane with a speed of U_1, see Figure 3.1. If U_1 is almost as large as U_0, the turbine will not extract much kinetic energy. On the other hand, if U_1 is very low, then not much air will pass through the turbine, resulting in less energy extraction. This suggests that there is an optimum value for the reduction of the wind speed. Below, we derive the relation among the optimum wind speed at the plane of the rotor, U_1, upstream wind speed, U_0, and downstream wind speed, U_2.

After the wind passes the rotor, the air speed would be reduced to U_2 and there would be a pressure drop. The initial pressure is p_1 and as air moves towards the rotor, the pressure rises to a pressure p_+. After passing the rotor, the pressure

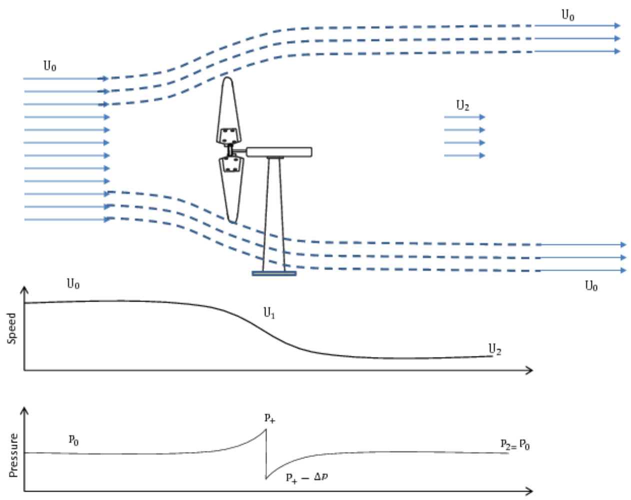


Figure 3.1: Interaction between the wind and a wind turbine.

suddenly falls by an amount of Δp. Therefore, immediately behind the turbine the pressure is $p_- = p_+ - \Delta p$, and further downstream the pressure again rises to $p_2 = p_0$. Curves for the wind speed and the pressure are shown in Figure 3.1.

Since the flow is frictionless and no work is done, the Bernoulli equation can be applied on both sides of the rotor to find the relation between the pressure p and the speed U:

$$\frac{1}{2}\rho U^2 + p = p_{\text{tot}}. \tag{3.1}$$

If we apply (3.1) for the flow upstream of the rotor, we get:

$$p_0 + \frac{1}{2}\rho U_0^2 = p_+ + \frac{1}{2}\rho U_1^2. \tag{3.2}$$

If we apply (3.1) downstream of the rotor plane, we get:

$$p_+ - \Delta p + \frac{1}{2}\rho U_1^2 = p_0 + \frac{1}{2}\rho U_2^2. \tag{3.3}$$

Subtracting (3.3) from (3.2) yields:

$$\Delta p = \frac{1}{2}\rho(U_0^2 - U_2^2). \tag{3.4}$$

If we look at one square meter of the rotor plane, the mass flow equals ρU, the momentum equals the mass times the velocity and the pressure equals the force per surface area. Thus, the differential pressure can be calculated as:

$$\Delta p = \rho U_1(U_0 - U_2). \tag{3.5}$$

From (3.4) and (3.5), we get:

$$U_1 = \frac{1}{2}(U_0 + U_2). \tag{3.6}$$

Therefore, the wind speed in the plane of the rotor is the average of the upstream and downstream velocities. Now that we have the expression for the optimum wind speed at the plane of the rotor as a function of ultimate wake velocities, we can now derive the expression for the power extracted by the turbine as a function of upstream velocity. The power of the turbine equals the change in the kinetic energy in the air:

$$P = \frac{1}{2}\rho U_1(U_0^2 - U_2^2)A. \tag{3.7}$$

Here, A is the surface area swept by the rotor. The axial force (thrust) on the rotor can be calculated as:

$$T = \Delta p A. \tag{3.8}$$

If one defines the axial induction factor, a, as the fractional decrease in wind velocity between the free stream and the rotor plane:

$$a = 1 - \frac{U_1}{U_0}, \tag{3.9}$$

then, $U_1 = (1-a)U_0$. The quantity, aU_0, is often referred to as the induced velocity at the rotor, in which case velocity of the wind at the rotor is a combination of the free stream velocity and the induced wind velocity. Using (3.9) and (3.6), we get:

$$U_2 = (1-2a)U_0. \tag{3.10}$$

From (3.10), the wind speed behind the rotor slows down as the axial induction factor increases. If $a = \frac{1}{2}$, then the wind has slowed to zero velocity behind the rotor. In this case, this simple theory proposed by Betz is no longer applicable. Using the above equation, (3.7) and (3.8), we derive equations for the generated power and force as a function of upstream velocity:

$$\begin{aligned} P &= \;2\rho a(1-a)^2 U_0^3 A, \\ T &= \;\;2\rho a(1-a) U_0^2 A. \end{aligned} \tag{3.11}$$

Wind turbine rotor performance is usually characterized by its power coefficient,

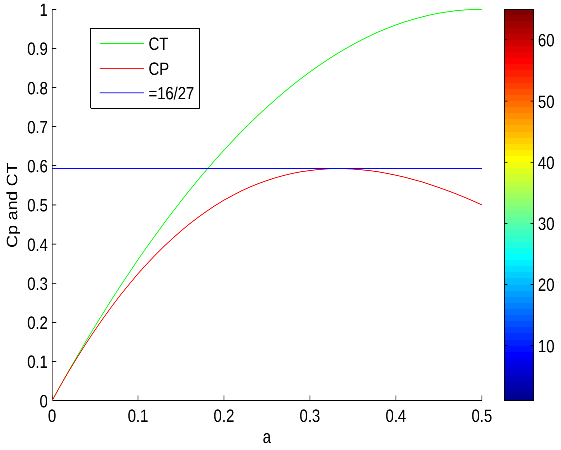


Figure 3.2: C_p and C_T for an idealized wind turbine.

C_p:

$$C_p = \frac{\text{Rotor Power}}{\text{Power in the wind}} = \frac{P}{\frac{1}{2}\rho U_0^3 A}. \tag{3.12}$$

This non-dimensional power coefficient represents the fraction of the power in the wind that is extracted by the rotor. Combining (3.11) and (3.12), we get:

$$\begin{aligned} C_p =& \quad 4a(1-a)^2, \\ C_T =& \quad 4a(1-a). \end{aligned} \tag{3.13}$$

Consequently, the equation (3.11) can be written in terms of C_p and C_T:

$$\begin{aligned} P =& \quad \frac{1}{2}\rho C_p U_0^3 A, \\ T =& \quad \frac{1}{2}\rho C_T U_0^2 A. \end{aligned} \tag{3.14}$$

In Figure 3.2, curves for C_p and C_T are shown. The maximum C_p is determined by taking the derivative of the power coefficient, the equation (3.13), with respect to

a and setting it equal to zero, yielding $a = \frac{1}{3}$. Therefore, the power coefficient, C_p, has an optimum at about 0.593 (exactly $\frac{16}{27}$) at an axial interference factor of 0.333 (exactly $\frac{1}{3}$). According to Betz, we have:

$$P = \frac{1}{2}\rho C_{p,\text{Betz}} U_0^3 A, \quad \text{with} \quad C_{p,\text{Betz}} = \frac{16}{27}. \tag{3.15}$$

Therefore, the maximum possible extractable energy is 59.3% of the total available wind energy from a given volume of wind. Wind turbine designers around the world try to design their wind energy extractor (blades) to catch up with this limit. In the next section, we show how to design the blades in order to extract maximum possible energy from the wind.

3.3 Rotor Design

3.3.1 Airfoil Theory

Figure 3.3 shows a typical wing section of the blade. The air hits the blade at an angle of α which is called the angle of attack. The reference line for this angle on the blade is the chord line. As the air hits the blade at the angle of α, an aerodynamic force is produced. This force can be broken down into two components, lift and drag. The lift force F_L, per definition, is perpendicular to the wind direction, and it can be calculated as:

$$F_L = C_L \frac{1}{2}\rho W_1^2 (bc), \tag{3.16}$$

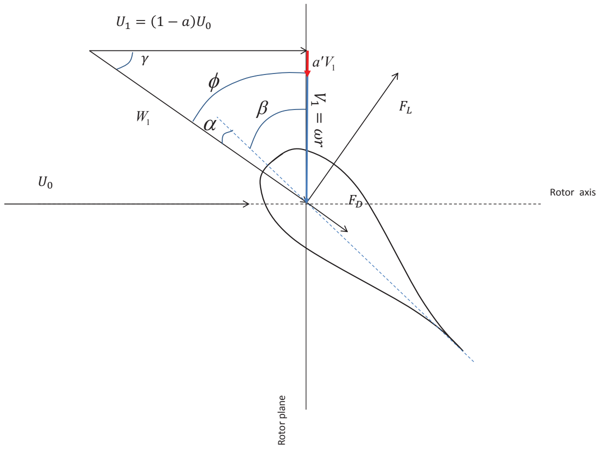


Figure 3.3: Velocities and angles on the wing section of the blade.

where C_L is the coefficient of lift, ρ is the density of air, W_1 is the relative wind speed, b is the width of the blade section, and c is the length of the chord line. Similarly, for the drag force, we have:

$$F_D = C_D \frac{1}{2} \rho W_1^2 (bc). \tag{3.17}$$

The lift and drag coefficient both depend on the angle of attack and it will be discussed in section 3.6. With the help of airfoil theory, the optimized values of blade pitch and chord length that create the most efficient blade geometry is derived in the following subsection.

3.3.2 Pitch Angle and Chord Length after Betz

Figure 3.3 shows the velocities and the angles in a given distance, r, from the rotor axis. To design the rotor, we have to define the pitch angle β and the chord length c. Both of them depend on the given radius that we are looking at; hence, we sometimes write $\beta(r)$ and $c(r)$. The blade, as shown in Figure 3.3, is moving upwards, thus the wind speed, seen from the blade, is moving downwards with a relative speed of $(1 + a^{'})V_1$, where:

$$V_1 = \omega r. \tag{3.18}$$

Here, ω is the angular speed of the rotor given by:

$$\omega = 2\pi n, \tag{3.19}$$

where n is the rotational speed of the rotor in round per second. Betz does not include rotation of the wind, i.e. $a^{'} = 0$, which yields:

$$W_1^2 = V_1^2 + U_1^2. \tag{3.20}$$

If we define the tip speed ratio as:

$$X = \frac{V_{\text{tip}}}{U_0} = \frac{\omega R}{U_0}, \tag{3.21}$$

then we have:

$$\gamma(r) = \arctan \frac{3rX}{2R}, \tag{3.22}$$

and

$$\Phi(r) = \arctan \frac{2R}{3rX}. \tag{3.23}$$

In the above equations, $a = \frac{1}{3}$ is used, as derived in the previous section, the power coefficient achieves its maximum at this vale. Therefore, the pitch angle is obtained by:

$$\beta(r)_{\text{Betz}} = \arctan \frac{2R}{3rX} - \alpha_D, \tag{3.24}$$

where α_D is the design angle of attack which is used for the blade design. Most often, this angle is chosen to be close to the angle that gives maximum glide ratio, see Figure 3.15; hence, this implies that α_D is in the range from 5° to 10°. However, this angle is sometimes reduced near the tip of the blade.

If we look at one blade element in the distance r from the rotor axis with the thickness dr, then the lift force is, see formula (3.16) and (3.17):

$$dF_L = C_L \frac{1}{2} \rho W_1^2 c dr, \tag{3.25}$$

and the drag force is:

$$dF_D = C_D \frac{1}{2} \rho W_1^2 c dr. \tag{3.26}$$

By decomposing the lift and the drag forces into the axial and the tangential components, we derive the following equation for the torque at the plane of the rotor:

$$d\Psi = \frac{1}{2} \rho W_1^2 c dr C_x, \tag{3.27}$$

where

$$C_x = C_L sin(\phi) - C_D cos(\phi). \tag{3.28}$$

Similarly, for the thrust at the rotor plane, we have:

$$dT = \frac{1}{2}\rho W_1^2 cdrC_y, \tag{3.29}$$

where

$$C_y = C_L cos(\phi) + C_D sin(\phi). \tag{3.30}$$

Now, in the design situation, we have $C_L \geq\geq C_D$, then (3.27) becomes:

$$d\Psi = \frac{1}{2}\rho W_1^2 cdrC_L cos(\gamma). \tag{3.31}$$

From the definition of the power, we have:

$$dP = d\Psi r\omega. \tag{3.32}$$

If we have B blades, then (3.31) including (3.32) give:

$$dP = B\frac{1}{2}\rho W_1^2 cdrC_L cos(\gamma)r\omega. \tag{3.33}$$

According to Betz, the blade element would also give:

$$dP = \frac{16}{27}\frac{1}{2}\rho U_0^3(2\pi rdr). \tag{3.34}$$

Using $U_0 = \frac{3}{2}W_1\cos(\gamma)$ and $V_1 = W_1\sin(\gamma)$, then (3.34) and (3.33) give:

$$c(r)_{\text{Betz}} = \frac{16\pi R}{9BC_{L,D}}\frac{1}{X\sqrt{X^2(\frac{r}{R})^2 + \frac{4}{9}}}, \tag{3.35}$$

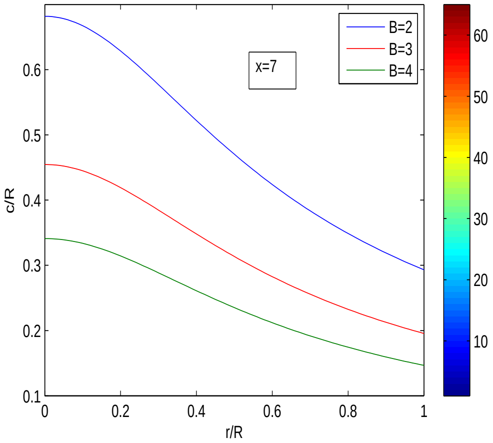


Figure 3.4: Chord length as a function of r, for different numbers of blades.

where $C_{L,D}$ is the lift coefficient at the chosen design angle of attack, $\alpha_{A,D}$. Tip speed ratio of about $X = 7$ is optimal and three blades seem to be state of the art. Figure 3.4 and 3.5 show the results of formula (3.35) concerning the chord length according to Betz.

3.3.3 Pitch Angle and Chord Length Considering Wake Rotation

In the previous analysis, using linear momentum theory, it was assumed that no rotation was imparted to the airflow. The previous analysis can be extended to the case where the rotating rotor generates angular momentum, which can be related to rotor torque. According to the conservation of angular momentum, the torque in the wind turbine shaft can only be created if there is a rotation in the downstream

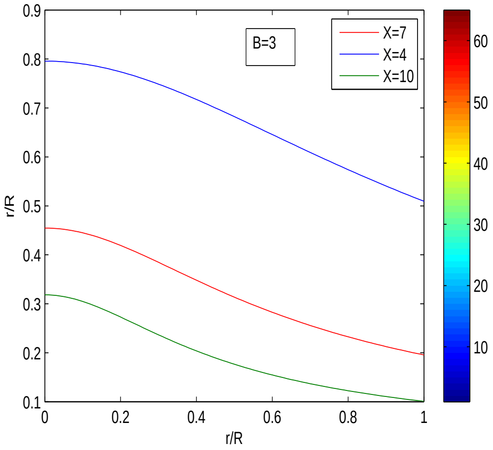


Figure 3.5: Chord length as function of r, for different tip speed ratios.

wake in the opposite direction of the rotor's rotation. Figure 3.6 shows downstream rotation of the wake that the wake rotates in the opposite direction to the rotor. By taking into account the torque producing the wake in the opposite direction of the rotor's rotation, the relative tangential speed of the rotating blade is $V_1 + a^{'}V_1$, as shown in Figure 3.7. From Figure 3.8, we have the following equations:

$$W_1 = W_0 \cos(\phi_1 - \phi), \tag{3.36}$$

$$U_1 = W_1 \sin(\phi). \tag{3.37}$$

By combining above two equations, we get:

$$U_1 = W_0 \cos(\phi_1 - \phi) \sin(\phi). \tag{3.38}$$

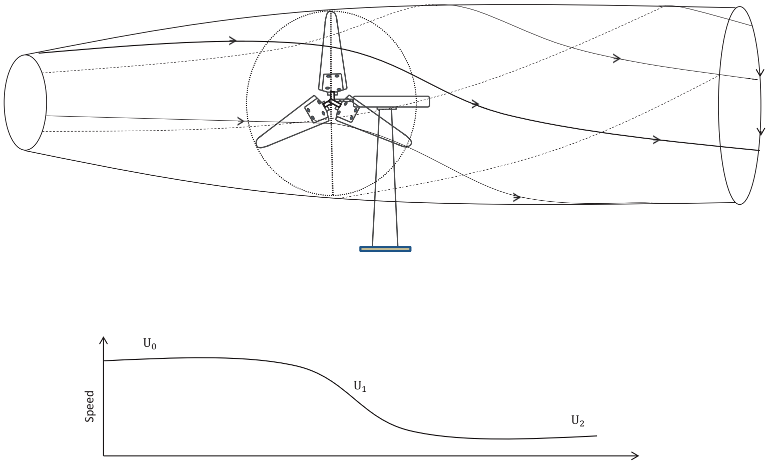


Figure 3.6: Downstream rotation of the wake.

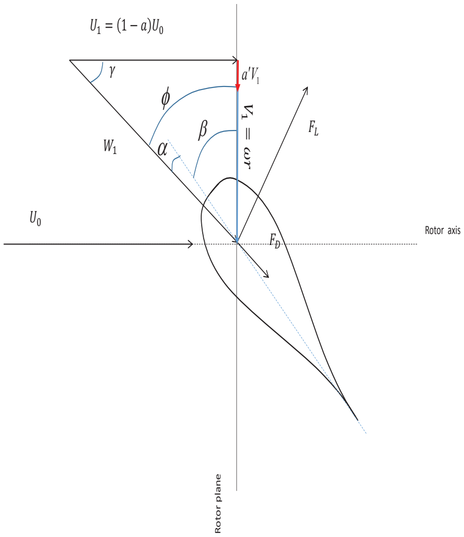


Figure 3.7: Relative velocities considering wake rotation.

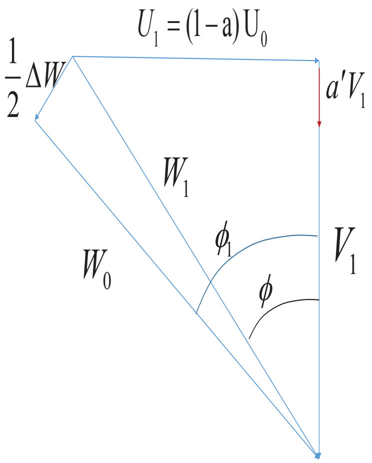


Figure 3.8: Speeds in the plane of rotor.

From Figure 3.8, we also have:

$$\Delta W = 2W_0 \sin(\phi_1 - \phi). \tag{3.39}$$

From the conservation of momentum, we have:

$$dF_L = \Delta W dq, \tag{3.40}$$

where dq is the math flow through the ring element in the radius r with the width dr:

$$dq = 2\rho\pi r dr U_1. \tag{3.41}$$

By using (3.38), (3.39) and the definition of power which equals the torque multiplied by angular velocity:

$$\begin{aligned} dp &= dM\omega \\ &= dF_L \sin(\phi) r\omega \\ &= \Delta w dq \sin(\phi) r\omega \\ &= r^2 \omega \rho 2\pi dr W_0^2 \sin(2(\phi_1 - \phi)) \sin^2(\phi_1). \end{aligned} \tag{3.42}$$

We have now a relation for the power of the ring element as a function of the flow angle. If we take the derivative of the above equation and solve it for flow angle, then we have:

$$\phi_{\max} = \frac{2}{3}\phi_1 = \frac{2}{3}\arctan(\frac{R}{Xr}), \tag{3.43}$$

and for the pitch anlge:

$$\beta_{\text{Schmitz}}(r) = \frac{2}{3}\arctan(\frac{R}{Xr}) - \alpha_D. \tag{3.44}$$

Using equations (3.40), (3.39), (3.38) and (3.43), we can derive:

$$dF_L = 2W_0^2 2\rho\pi r dr \sin^2(\frac{\phi_1}{3}) \cos^2(\frac{\phi_1}{3}). \tag{3.45}$$

From the air foil theory, we have:

$$dF_L = \frac{1}{2}\rho W_0^2 B c dr C_L \cos^2(\frac{\phi_1}{3}). \tag{3.46}$$

By combining above two equations:

$$c_{\text{Schmitz}}(r) = \frac{1}{B}\frac{16\pi r}{C_L} \sin^2(\frac{\phi_1}{3}). \tag{3.47}$$

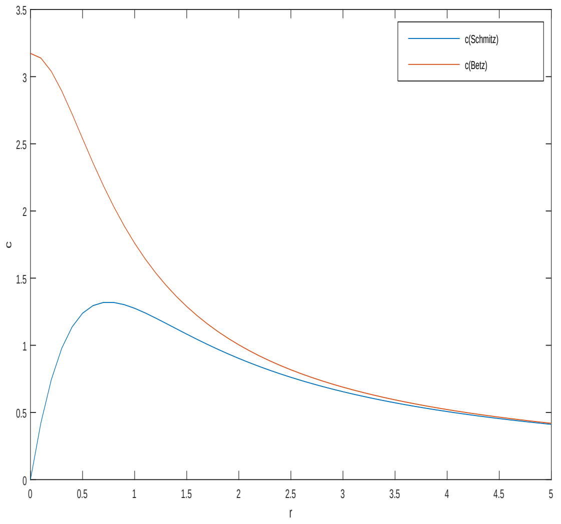


Figure 3.9: Optimal chord length.

Using equation (3.43), the above equation becomes:

$$c_{\text{Schmitz}}(r) = \frac{1}{B}\frac{16\pi r}{C_L}\sin^2(\frac{1}{3}\arctan(\frac{R}{Xr})). \tag{3.48}$$

Figures 3.9 and 3.10 show the comparison between Betz and Schmitz' s formula for the design of the optimal pitch angle and the chord length. From these figures, it appears that Betz and Schmitz' s theory differs on designing the optimal pitch angle and the chord length near the hub, but near the tip there are no differences between them. So far, we have focused on the maximum power that can be generated by a wind turbine and the design of an optimal rotor. In the following section, we focus on the power calculation of a given rotor.

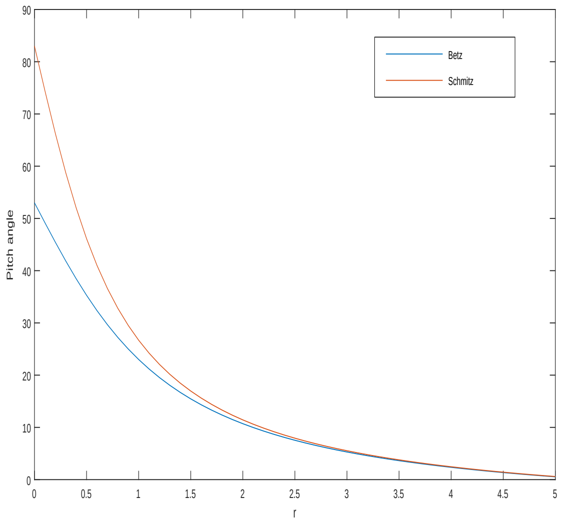


Figure 3.10: Optimal pitch angle.

3.4 Blade Element Theory

The BEM method solves for the torque and thrust using the law of momentum, angular momentum, and the Bernoulli's equation. Let U_0 be the given upstream velocity, $U_1(r)$ be the unknown downstream velocity, $U_2(r)$ be the unknown velocity at the rotor plane, see Figure 3.1. Then, by the law of momentum and angular momentum, one can get:

$$dT(r) = 2\pi r \rho U_1(r)(U_0 - U_2(r))Bdr, \tag{3.49}$$

and

$$d\Psi(r) = 2\pi r^2 \rho U_1(r) V_2(r) Bdr. \tag{3.50}$$

On the other hand, from the airfoil theory, see the equations (3.27) and (3.29), we have:

$$\begin{aligned} dT(r) &= \frac{1}{2}\rho W_1^2 cBC_y dr, & (3.51)\\ d\Psi(r) &= \frac{1}{2}\rho W_1^2 cBC_x r dr. & (3.52) \end{aligned}$$

Hence, using the above equations, (3.49) and (3.50), we have:

$$\begin{aligned} \frac{a}{1-a} &= \frac{cBC_y}{8\pi r\sin^2(\phi)}, & (3.53)\\ \frac{a^{'}}{a^{'}+1} &= \frac{cBC_x}{8\pi r\sin(\phi)\cos(\phi)}. & (3.54) \end{aligned}$$

In the above equations, we have used:

$$\tan(\phi(r)) = \frac{1-a(r)}{1+a^{'}(r)}\frac{U_0}{r\Omega}, \quad \alpha(r) = \phi(r) - \beta(r), \tag{3.55}$$

$$W_1(r) = \frac{U_0(1-a(r))}{\sin(\phi(r))} = \frac{r\Omega(1+a^{'}(r))}{\cos(\phi(r))}. \tag{3.56}$$

By solving equations (3.53) and (3.54) for $a^{'}$ and a, we get:

$$a(r) = \frac{1}{\frac{4\sin^2(\phi(r))}{\sigma(r)C_x(r)}+1}, \tag{3.57}$$

$$a^{'}(r) = \frac{1}{\frac{4\sin(\phi(r))\cos(\phi(r))}{\sigma(r)C_y(r)}-1}, \tag{3.58}$$

where

$$\sigma(r) = \frac{c(r)B}{2\pi r}, \tag{3.59}$$

$$C_x(r) = C_L(\alpha(r))\cos(\phi(r)) + C_D(\alpha(r))\sin(\phi(r)), \tag{3.60}$$

$$C_y(r) = C_L(\alpha(r))\sin(\phi(r)) - C_D(\alpha(r))\cos(\phi(r)). \tag{3.61}$$

For rotors with few blades, it can be shown that a better approximation of a and $a^{'}$ is:

$$a(r) = \frac{1}{\frac{4F_{\text{corr}}(r)\sin^2(\Phi(r))}{\sigma(r)C_y(r)} + 1}, \tag{3.62}$$

$$a^{'}(r) = \frac{1}{\frac{4F_{\text{corr}}(r)\sin(\Phi(r))\cos(\Phi(r))}{\sigma(r)C_x(r)} - 1}, \tag{3.63}$$

where F_{corr} is given in the equation (3.80). This simple momentum theory breaks down when $a(r)$ is greater than $a_c = 0.2$. In this case, one shall apply:

$$a(r) = \frac{1}{2}\Bigg(2 + K_a(r)(1 - 2a_c) - \sqrt{(2 + K_a(r)(1 - 2a_c))^2 + 4(K_a(r)a_c^2 - 1)}\Bigg), \tag{3.64}$$

where

$$K_a(r) = \frac{4F_{\text{corr}}(r)\sin^2(\Phi(r))}{\sigma(r)C_x(r)}. \tag{3.65}$$

We solve the nonlinear system with two unknowns $a(r)$ and $a^{'}(r)$ by an iterative method in Algorithm 2. With the solution of $a(r)$ and $a^{'}(r)$, one can compute the thrust force, the torque and the generated power by the wind turbine on one ring element at radius r.

The BEM method is simple to apply and has been popular for many years in analyzing the performance of a wind turbine. Nevertheless, the BEM has limitations such as no aerodynamic interactions between different blade elements, and it usually underpredicts the power generated by a wind turbine (see [38,60]). Therefore, It is

Algorithm 2 Solve for $a(r), a^{'}(r)$ at radius r

```
Given $\Omega$, $\beta$, $U_0$; $k \leftarrow 0; a^{(k)}(r) \leftarrow 0; a^{'(k)}(r) \leftarrow 0;$
repeat
    $k \leftarrow k + 1;$
    evaluate $\tan(\phi(r))$, $\alpha(r)$ in (3.55);
    evaluate $C_L$ in (3.78) and $C_D$ in (3.79);
    evaluate $C_x$ in (3.61) and $C_y$ in (3.60);
    evaluate $a^{(k)}(r)$ in (3.57) and $a^{'(k)}(r)$ in (3.58);
    if $a^{(k)}(r) \geq 0.2$ then
        evaluate $a^{(k)}(r)$ in (3.64);
    end if
until $|a^{(k)}(r) - a^{(k-1)}(r)| < \epsilon; |a^{'(k)}(r) - a^{'(k-1)}(r)| < \epsilon$
```

necessary to use different methods to analyze the performance of a wind turbine. One such method is the actuator line method which is developed by Sorensen et al. [91] for modeling and analyzing a single turbine. However, it is quite challenging to analyze the performance of multiple turbines due to the wake interactions amongst the turbines. In the following section, we focus on modeling the airflow to optimize the total power while considering the wake interactions among the wind turbines.

3.5 Modeling of the Flow for Power Optimization in Wind Farm

For a wind farm with N turbines, the model for power optimization in its condensed form is the following:

$$\begin{aligned} &\underset{\substack{\beta_1,\dots,\beta_N \\ \Omega_1,\dots,\Omega_N}}{\text{Max}} \quad \sum_{i=1}^{N} P_i \\ &\qquad\qquad \text{subject to}\,(3.67)\text{ and} \\ &\qquad\qquad \beta_{\min} \leq \beta_i \leq \beta_{\max}, \\ &\qquad\qquad \Omega_{\min} \leq \Omega_i \leq \Omega_{\max}, \\ &\qquad\qquad \forall i \in \{1, 2, \dots, N\}. \end{aligned} \tag{3.66}$$

Here, the decision variables are the rotational speed Ω_i and the pitch angle β_i of the ith turbine, for $i = 1, \cdots, N$. The parameters $\beta_{\min}$, $\beta_{\max}$, $\Omega_{\min}$ and $\Omega_{\max}$ are the physical limits of the adjustable pitch angle and the rotational speed. The

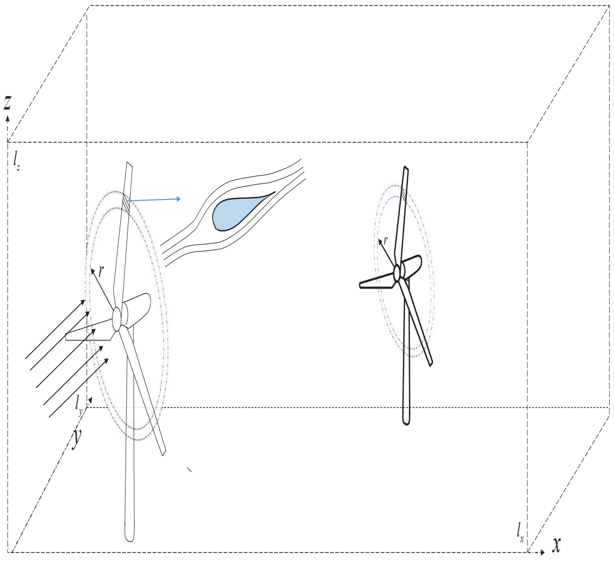


Figure 3.11: Computational domain.

objective function is to maximize the total power where P_i is the power generated by the ith turbine which is a complicated nonlinear function of the direct decision variables as well as other derived quantities from the PDE system (3.67). Due to its complexity, we derive it in detail in section 3.6. The constraints are the three-dimensional Navier-Stokes equations which govern the relationship between the wind flow field and the external forces placed in the computational domain, see the schematic diagram in Figure 3.11. These equations on a rectangular domain,

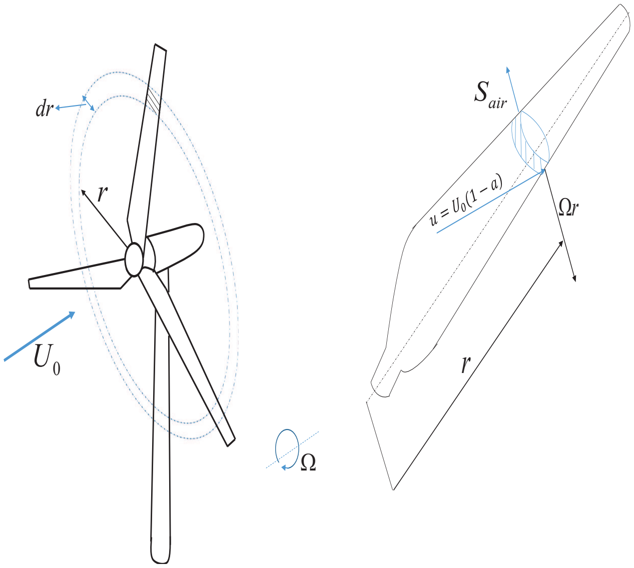


Figure 3.12: Cross-sectional airfoil element.

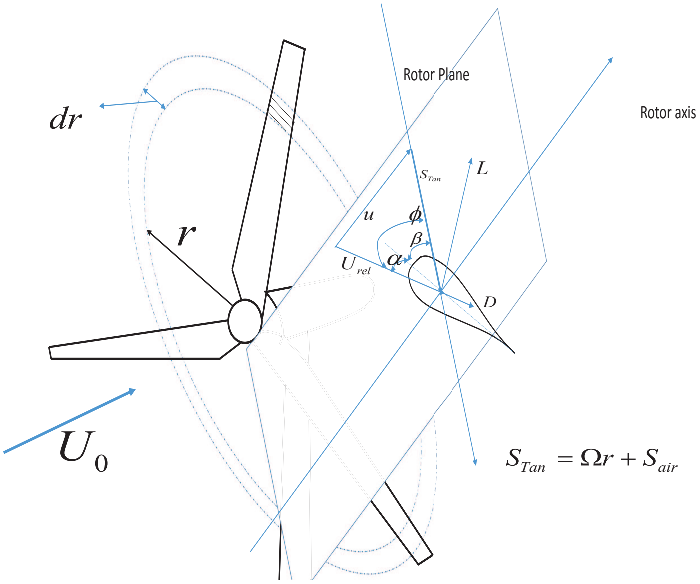


Figure 3.13: Cross-sectional airfoil element showing force vectors.

$D = [0, l_x] \times [0, l_y] \times [0, l_z]$, are given by:

$$\begin{cases} \dfrac{\partial U}{\partial t} + UU_x + VU_y + Wu_z = -\dfrac{1}{\rho}p_x + \nu(U_{xx} + U_{yy} + U_{zz}) + f_u(t, x, y, z), \\ \dfrac{\partial V}{\partial t} + UV_x + VV_y + WV_z = -\dfrac{1}{\rho}p_y + \nu(V_{xx} + V_{yy} + V_{zz}) + f_v(t, x, y, z), \\ \dfrac{\partial W}{\partial t} + UW_x + VW_y + WW_z = -\dfrac{1}{\rho}p_z + \nu(W_{xx} + W_{yy} + W_{zz}) + f_w(t, x, y, z), \end{cases} \tag{3.67}$$

with the following boundary conditions:

$$\begin{cases} U(t,0,\cdot,\cdot) = U_0, \quad V(t,0,\cdot,\cdot) = 0, \quad W(t,0,\cdot,\cdot) = 0, \\ U_x(t,l_x,\cdot,\cdot) = 0, \quad V_x(t,l_x,\cdot,\cdot) = 0, \quad W_x(t,l_x,\cdot,\cdot) = 0, \\ U_y(t,\cdot,0,\cdot) = 0, \quad V_y(t,\cdot,0,\cdot) = 0, \quad W_y(t,\cdot,0,\cdot) = 0, \\ U_y(t,\cdot,l_y,\cdot) = 0, \quad V_y(t,\cdot,l_y,\cdot) = 0, \quad W_y(t,\cdot,l_y,\cdot) = 0, \\ U_z(t,\cdot,\cdot,0) = 0, \quad V_z(t,\cdot,\cdot,0) = 0, \quad W_z(t,\cdot,\cdot,0) = 0, \\ U_z(t,\cdot,\cdot,l_z) = 0, \quad V_z(t,\cdot,\cdot,l_z) = 0, \quad W_z(t,\cdot,\cdot,l_z) = 0, \end{cases} \tag{3.68}$$

the initial conditions:

$$U(0,\cdot,\cdot,\cdot) = U_0, \quad V(t,\cdot,\cdot,\cdot) = 0, \quad W(t,\cdot,\cdot,\cdot) = 0, \tag{3.69}$$

and the continuity equation:

$$U_x + V_y + W_z = 0. \tag{3.70}$$

Here, ν is the viscosity, ρ is the density of the air, U_0 is the atmosphere wind speed. The unknown variables U, V and W are the wind speed in x, y and z-direction, respectively. The unknown variable p is the pressure, and the external forces are given by f_u, f_v and f_w. One can view (3.67) as a blackbox of the optimization model (3.66) where the set of operating control variables $(\beta_i, \Omega_i), i = 1, \cdots, N$ is the input, and the output is the total generated power. We remind the reader that inside this blackbox f_u, f_v and f_w are non-stationary blade forces which depend on the flow velocities U, V and W. From (3.67), it is evident the solution of U, V and W depends on f_u, f_v and f_w. The next two sections are devoted to disentangle this complicated relationship. Section 3.6 describes this facet of the relation: assume U, V and W are given for a turbine, then how to compute the blade forces f_u, f_v and f_w. Section 3.7 describes this facet of the relation: assume f_u, f_v and f_w are given, then how to compute U, V and W.

3.6 Blade Aerodynamics and Power Production

Figure 3.12 shows a cross-sectional airfoil element on the blade at radius r in the plane of the rotor. A blade is viewed as a continuous stack of airfoils from the hub to the tip. In a modern blade, the airfoils deform from the hub to the tip smoothly, but gradually. The aerodynamic forces acting on an airfoil is related to the local relative velocity. From Figure 3.13, the relative velocity at radius r from the hub is:

$$U_{rel}^2(r) = (U(r))^2 + (S_{Tan}(r))^2, \tag{3.71}$$

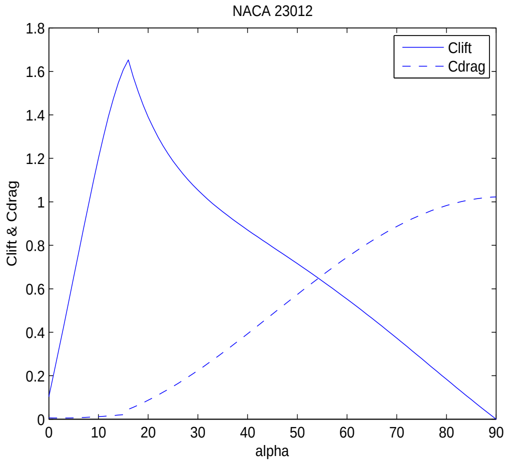


Figure 3.14: Coefficient of lift and drag.

where

$$S_{Tan}(r) = \Omega r + S_{air}(r). \tag{3.72}$$

In the above equations, $U(r)$ and $S_{Tan}(r)$ are the axial speed and the relative tangential speed at radius r from the hub, respectively. In the equation (3.72), S_{air} is the wind speed in tangential direction which is computed by the orthogonal projection of V and W onto the tangential direction. One has to solve the Navier-Stokes equations (3.67) to find U, V and W. We remind that Ω is a decision variable in the joint optimization model (3.66). The angle between U_{rel} and the rotor plane is:

$$\phi(r) = \tan^{-1}(\frac{U(r)}{S_{Tan(r)}}). \tag{3.73}$$

The angle of attack between the relative wind velocity and the airfoil chord line is:

$$\alpha(r) = \phi(r) - \beta(r), \tag{3.74}$$

where $\beta(r)$ is the local pitch angle between the airfoil and the rotor plane at the radius r. From the airfoil theory, the lift force $L(r)$, perpendicular to the relative wind direction, and the drag force $D(r)$ are:

$$L(r) = \frac{1}{2}\rho U_{rel}^2(r)c(r)b(r)C_L(\alpha(r)), \tag{3.75}$$

$$D(r) = \frac{1}{2}\rho U_{rel}^2(r)c(r)b(r)C_D(\alpha(r)), \tag{3.76}$$

where $c(.)$ is the chord length of the airfoil and $b(.)$ is the width of the blade section. In the above equations, $C_L(.)$ and $C_D(.)$ are the lift and drag coefficients at radius r which depend on the local angle of attack $\alpha(\cdot)$, see Figure 3.14. The ratio of these two coefficients $GR \equiv C_L/C_D$ is called the glide ratio. Normally, we are interested in at high glide ratio for wind turbines as well as for air planes. Figures 3.15 and 3.14 show the curves for the glide ratio, the lift and the drag coefficients. These curves are usually obtained by testing wing profiles in wind tunnels. However, for numerical studies, it is convenient to have the curves as functions. For NACA 23012 profile [18] and for $0^\circ < \alpha(.) < 16^\circ$, $C_L(.)$ and $C_D(.)$ are characterized by the following polynomials:

$$C_{L,D}(\alpha(r)) = k_0 + k_1\alpha(r) + k_2\alpha(r)^2 + k_3\alpha(r)^3 + k_4\alpha(r)^4, \tag{3.77}$$

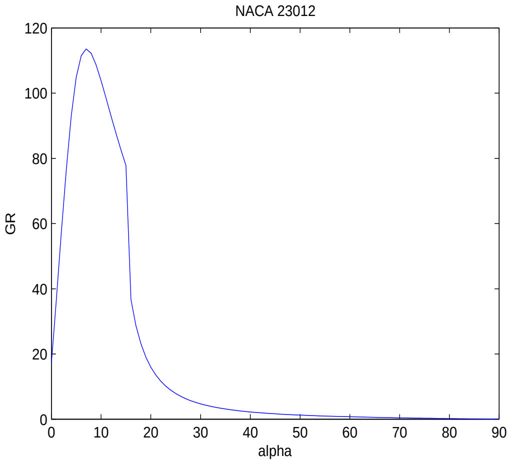


Figure 3.15: Glide ratio as a function of the angle of attack.

where the polynomial coefficients are shown in Table 3.1. For an angle larger than the critical angle of attack, 16°, we will apply the following correction:

$$C_L(\alpha(r)) = A_1 \sin(2\alpha(r)) + A_2 \frac{\cos^2(\alpha(r))}{\sin(\alpha(r))}, \tag{3.78}$$

$$C_D(\alpha(r)) = B_1 \sin^2(\alpha(r)) + B_2 \cos(\alpha(r)) + C_{Ds}, \tag{3.79}$$

where

$$\begin{aligned}
A_1 &= \frac{B_1}{2}, \\
A_2 &= (C_{Ls} - C_{\mathrm{D,max}} \sin(\alpha_{st}) \cos(\alpha_{st})) \frac{\sin(\alpha_{st})}{\cos^2(\alpha_{st})}, \\
B_1 &= C_{\mathrm{D,max}}, \\
B_2 &= \frac{1}{\cos(\alpha_{st})}(C_{Ds} - C_{\mathrm{D,max}} \sin^2(\alpha_{st})).
\end{aligned}$$

Table 3.1: NACA 23012 coefficients, $\alpha \in [0°, 16°]$.

	$C_L(.)$	$C_D(.)$
k_0	$1.0318e-1$	$6.0387e-3$
k_1	$1.0516e-1$	$-3.6282e-4$
k_2	$1.0483e-3$	$5.4269e-5$
k_3	$7.3487e-6$	$6.5341e-6$
k_4	$-6.5827e-6$	$-2.8045e-7$

Here C_{Ls} and C_{Ds} are the lift and drag coefficients at stall angle of attack α_{st}, and $C_{\text{D,max}}$ is the maximal value of the drag coefficient which is approximately 1, see Figure 3.14. We also need another correction due to the cross-flow effect at the blade tip. To take into account this effect, we employ the correction formula $C_L(.) = \frac{C_L(.)}{F_{\text{corr}}}$ and $C_D(.) = \frac{C_D(.)}{F_{\text{corr}}(.)}$ where:

$$F_{\text{corr}}(r) = \frac{2}{\pi} \arccos(\exp(-B\frac{R-r}{2r\sin(\phi(r))})). \qquad (3.80)$$

In the above equation, B and R are the number of blades and the rotor radius, respectively. Now, we project the lift and drag force onto the axial and the tangential direction to get their components in these directions:

$$F_x(r) = L(r)\cos(\phi(r)) + D(r)\sin(\phi(r)), \tag{3.81}$$

$$F_y(r) = L(r)\sin(\phi(r)) - D(r)\cos(\phi(r)). \tag{3.82}$$

Consequently, for the ith turbine with B blades, the thrust force $dT_i(r)$, the torque $d\Psi_i(r)$ and the power $dP_i(r)$ are:

$$dT_i(r) = F_x(r)Bdr, \tag{3.83}$$

$$d\Psi_i(r) = F_y(r)Bdr, \tag{3.84}$$

$$dP_i(r) = d\Psi_i(r)(\Omega r). \tag{3.85}$$

Here, $dT_i(r)$ is along axial direction, and it is used to compute the source term f_u in the Navier-Stokes equations (3.67). The torque $\Psi_i(r)$ is along the tangential direction, and one must project it onto y and z-direction to compute the source terms f_v and f_w in the Navier-Stokes equations (3.67). From the equation (3.85), the generated power for the ith turbine with blade length R is:

$$P_i = \int_0^R dP(r) = \int_0^R F_y(r)B\Omega r dr. \tag{3.86}$$

Using equations (3.75) and (3.76), the power generated by the ith turbine is:

$$P_i = \frac{\rho B \Omega}{2} \int_0^R \Bigg(U_{rel}^2(r)c(r)b(r) \\ \Bigg(C_L(\alpha(r))\sin(\phi(r)) - C_D(\alpha(r))\cos(\phi(r)) \Bigg) \Bigg) r dr. \tag{3.87}$$

Summing P_i from $i = 1$ to N will give the objective function of the optimization model (3.66) which is the total power generated by all N turbines.

3.7 Numerical Solution of the Navier-Stokes Equations

By using the curl operator to equations (3.67), one gets the following dimensionless form of the vorticity transport equation:

$$\frac{\partial \omega}{\partial t} + \mathbf{U}.\nabla\omega = \omega.\nabla\mathbf{U} + \frac{1}{Re}\nabla^2\omega + \nabla \times \mathbf{f}_\epsilon, \tag{3.88}$$

where $\mathbf{U} = (U, V, W)$ and $\omega = (\xi, \eta, \zeta)$. The vorticity vector ω is defined as:

$$\omega = \nabla \times \mathbf{U}. \tag{3.89}$$

By taking the curl of equation (3.89) and using the continuity equation (3.70), the following velocity Poisson equation can be obtained:

$$\nabla^2\mathbf{U} = -\nabla \times \omega. \tag{3.90}$$

We seek the solutions for equations (3.88) and (3.90) with the boundary and initial conditions given in (3.68) and (4.4). In order to ensure accuracy in the prediction of velocities and vorticities, a staggered grid system as displayed in Figure 3.16 is used in the present numerical scheme. Let U^n, V^n, W^n, ξ^n , η^n and ζ^n be the numerical approximation of U, V, W, ξ ,η and ζ at time step n, then we calculate the solution in new time step by the following steps:

1. The velocity Poisson equations, obtained as a result of taking curl of the vorticity, are made parabolic using the false-transient technique [7, 32, 56, 63]:

$$\alpha \frac{\partial U^n}{\partial t} - \nabla^2 \mathbf{U^n} - \nabla \times \omega^{\mathbf{n-1}} = 0, \tag{3.91}$$

where α is a relaxation parameter. Central finite differencing scheme is used to approximate the second order derivatives that leads to a large linear system to be solved. There are many approaches to solve the linear system in each time step, for instance, Multigrid method, incomplete LU factorization approach, Generalized Minimum Residual technique with an appropriate preconditioning. Here, we use Generalized Minimum Residual technique, but in a sparse format [75].

After we calculate velocities U and V using (3.91), then velocity W is calculated from the continuity equation as given below:

$$\frac{\partial^2 W^n}{\partial z^2} = -\frac{\partial}{\partial z}(\frac{\partial U^n}{\partial x} + \frac{\partial V^n}{\partial y}). \tag{3.92}$$

Since the velocities U and V are already known, the resulting set of equations from (3.92) can be solved using the TDMA (Tridiagonal matrix algorithm). The main advantage of the present numerical solution procedure is that it assures a divergence-free solution for the velocity field, in addition to achieving a higher numerical accuracy and a significant reduction in the computational time.

2. The vorticity transport equations are discretized in time using the explicit scheme for the nonlinear term and implicit scheme for the linear term. The non-

linear terms are discretized explicitly to avoid the solution of a nonlinear system. This introduces limitations on the length of the time step which is proportional to the spacial resolution (CFL Condition):

$$\frac{\partial \omega^{\mathbf{n}}}{\partial t} + \mathbf{U^n}.\nabla \omega^{\mathbf{n-1}} = \omega^{\mathbf{n-1}}.\nabla \mathbf{U^n} + \frac{1}{Re}\nabla^2 \omega^{\mathbf{n}} + \nabla \times \mathbf{f}_\epsilon. \tag{3.93}$$

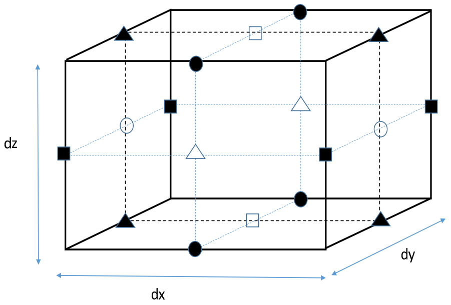
The central finite differencing scheme is used to approximate the first and second derivatives which leads to a large linear system, and it is solved by the Generalized Minimum Residual technique. In the equation (3.93), the source term, $\mathbf{f}_\epsilon = (f_{\epsilon u}, f_{\epsilon v}, f_{\epsilon w})$ acts as a singular vorticity source along the rotor blades. To avoid singular behavior, $\mathbf{f}_\epsilon$ is formed by taking the convolution of the computed normal load, $\mathbf{f} = (f_u, f_v, f_w)$, and a regularization kernel, η_ϵ, as shown below [91]:

$$\mathbf{f}_\epsilon = \mathbf{f} \otimes \eta_\epsilon, \tag{3.94}$$

where

$$\eta_\epsilon = \frac{1}{\epsilon^3 \pi^{\frac{3}{2}}} exp(-(\frac{r}{\epsilon})^2). \tag{3.95}$$

We remind the reader that f_u is computed directly using the equation (3.83). However, one needs to project the torque in the equation (3.84) which is along the tangential direction onto y and z-direction to compute f_v and f_w. Finally, for a given pitch angle and rotational velocity, to compute U, V, W and $\mathbf{f}$ simultaneously, we propose a time-stepping procedure which at every instant assures a time-true solution. This procedure is summarized in Algorithm 3.



Velocity	U	V	W
	○	△	○

Vorticity	ξ	η	ς
	●	▲	■

Figure 3.16: Staggered grid.

Algorithm 3 Iterations between Navier-Stokes equation and blade Forces

```
for a given $\beta_i, \Omega_i, i = 1, \cdots, N$, let $n = 0$, $U^n(0, x, y, z) = U_0$,$V^n(0, x, y, z) = 0$, $W^n(0, x, y, z) = 0,\ \forall(x, y, z) \in D$.
repeat
    Using $U^n$, $V^n$, $W^n$ at the rotor plane and the equations (3.71)-(3.87), compute $f_u^n$, $f_v^n$, $f_w^n$ and the generated power.
    Solve the Navier-Stokes equations to get $U^{n+1}$, $V^{n+1}$ and $W^{n+1}$.
until $|U^{n+1} - U^n| < \epsilon$, $|v^{n+1} - V^n| < \epsilon$ and $|W^{n+1} - W^n| < \epsilon$
```

3.7.1 Generalized Minimal Residual Algorithm

One of the most effective iterative methods for solving large sparse symmetric positive definite linear systems of equations is the conjugate gradient method [11], [34]. Several different generalizations of this method have been presented in the recent years to deal with nonsymmetric problems and symmetric indefinite problems [3, 21, 22, 40, 68, 70]. Recently, Youcef Saad developed an efficient numerical algorithm called Generalized Minimum Residual algorithm for solving nonsymmetric linear systems which is based on the Arnoldi process [2, 74, 76]. This generalized method, summarized in Algorithm 4, is to intend to solve the following system:

$$Ax = b. \tag{3.96}$$

In Algorithm 4, $\bar{H}_k$ is a $(k+1) \times k$ matrix which is the same as $H_k = V_k^T A V_k$, except for an additional row whose only nonzero element is $h_{k+1,k}$ in the $(k+1, k)$ position.

3.8 Joint Optimization Algorithm

The joint optimization model (3.66) is a PDE-constrained optimization model with the objective of maximizing the total power output of the wind farm. This objective function is the sum of the power produced by the individual turbines, P_i, which is given in the equation (3.87). From this equation, we note that P_i is a complicated

Algorithm 4 Generalized Minimal Residual Algorithm

```
1: Choose $x_0$ and compute $r_0 = b - Ax_0$ and $v_1 = \frac{r_0}{\|r_0\|}$.
2: for $j = 1, 2, 3, \ldots, m$ do
3:     $h_{i,j} = (Av_j, v_i),\ i = 1, 2, \ldots, j,$
4:     $\hat{v}_{j+1} = Av_j - \sum_{i=1}^{j} h_{i,j} v_i,$
5:     $h_{j+1,j} = \|\hat{v}_{j+1}\|$ and $v_{j+1} = \frac{\hat{v}_{j+1}}{h_{j+1,j}}$.
6: end for
7: Form the approximate solution: $x_m = x_0 + V_m y_m$ where $y_m$ minimizes $\|\beta e_1 - \bar{H}_m y\|$, $y \in R^m$.
8: Restart: compute $r_m = b - Ax_m$; if satisfied then stop else compute $x_0 = x_m$, $v_1 = \frac{r_m}{\|r_m\|}$ and go to 2.
```

nonlinear function of the direct decision variables as well as the wind speed at the plane of the rotor which is the solution of Navier-Stokes equations (3.67). Therefore, it is almost impossible to find the derivative of the objective function. Hence, we adopt the pattern search algorithm [67] which is a derivative free method to solve the joint optimization model (3.66). The pattern search method starts with an arbitrary feasible initial point, i.e., a set of operating configuration, $\beta_i, \Omega_i, i = 1, \cdots, N$, and seek a better set of operating configuration to gain more power. For the model (3.66), let x be the vector of all decision variables, D_{feasible} be the feasible region:

$$\begin{aligned} x &\equiv [\beta_1, \cdots, \beta_N, \Omega_1, \cdots, \Omega_N]^T, &(3.97)\\ D_{\text{feasible}} &= \{x \mid \beta_{1\min} \leq x_1 \leq \beta_{1\max}, \ldots, \Omega_{N\min} \leq x_{2N} \leq \Omega_{N\max}\}, &(3.98)\end{aligned}$$

and f be the objective function:

$$f \equiv \sum_{i=1}^{N} P_i, \tag{3.99}$$

then the joint optimization algorithm based on the pattern search method in its matrix notation is presented in Algorithm 5. The parameters used in this algorithm are the convergence tolerance $\gamma_{\text{tol}} = 1 \times 10^{-6}$, the contraction parameter $\theta = 0.5$, the aggressive parameter $\eta = 2$, the sufficient increase function $\mu(s) = s^3/2$ and the direction set $d \equiv \{p_i, i = 1, \cdots, n+1\}$. In this direction set, the search directions are given by $p_i = \frac{1}{2n}E - E_i$, for $i = 1, \cdots, n$, and $p_{n+1} = \frac{1}{2n}E$ where E is the n dimensional vector of all ones and E_i is the ith column of the unit matrix of

Algorithm 5 Joint Optimization

```
initialize the parameters $\gamma_{\text{tol}}$, $\theta$, $\eta$, $\mu(\cdot)$, guess $x_0$ and set $\gamma_0 \geq \gamma_{tol}$.
evaluate $f(x_0)$ by Algorithm 3 and (3.87).
for $k = 1, 2, 3, \ldots$ do
    if $\gamma_k \leq \gamma_{\text{tol}}$ then
        return.
    end if
    for $p_k \in d$ do
        if $(x_k + \gamma_k p_k) \notin D_{\text{feasible}}$ then
            $x_k + \gamma_k p_k = x_b$.
        end if
        evaluate $f(x_k + \gamma_k p_k)$ by Algorithm 3 and (3.87).
        if $f(x_k + \gamma_k p_k) > f(x_k) + \mu(\gamma_k)$ then
            $x_{k+1} \leftarrow x_k + \gamma_k p_k$ and $\gamma_{k+1} \leftarrow \gamma_k \eta$.
            break.
        else
            $x_{k+1} \leftarrow x_k$ and $\gamma_{k+1} \leftarrow \gamma_k \theta$.
        end if
    end for
end for
```

size n. Note that the search direction does not come from the gradient, but rather from a predetermined direction set d. Moreover, one member of d is a improving direction [67].

3.9 Numerical Simulation

In this section, we present two numerical case studies to test the efficiency and accuracy of the proposed numerical algorithm. In these numerical case studies, we use a three-bladed wind turbine with a rotor diameter of 10 meters where the blade sections consist of NACA 23012 series airfoils. The chord length and the manufactured twist angle of this turbine are obtained using the formulas given in (3.48) and (3.44).

To capture the gradients of the flow field, grid points are concentrated near the blade tips and stretched in the axial direction as well as in the y and z-direction. In the case of one turbine, the resulting grid consists of 76 grid points in the axial direction, 80 points in the y-direction and 92 points in the z-direction. In the case of two turbines, the resulting grid consists of 96 grid points in the axial direction, 80 points in the y-direction and 92 points in the z-direction. In the axial direction, the grid spacing ranges from $dx = 0.02$ at the rotor plane to about $dx = 1.9476$ in the far wake and in the y-direction, the spacing takes values from $dy = 0.02$ near the tip to about $dy = 1.2150$ at the lateral boundary. Moreover,

Table 3.2: Sections of NACA 23012 blade

relative radius	0.188	0.313	0.438	0.563	0.688	0.813	0.938
pitch	24.2	14.7	9.4	6.0	3.8	2.2	1.0
relative chord	0.259	0.212	0.169	0.138	0.116	0.100	0.087

in the z-direction, the spacing takes values from $dz = 0.0346$ near the tip to about $dz = 1.4863$ at the lateral boundary. The computations are carried out on a $100m \times 60m \times 60m$ computational domain and at an effective Reynolds number of $Re = U_0 * R/\nu = 5000$. The grid spacing and the value of the used Reynolds number are of course a compromise between the accuracy and computing costs. To ensure that the flow is fully developed in most of the wake, we carry out the computations until t=10 with a time step of $dt = 1 \times 10^{-3}$ which corresponds to 10000 time steps.

3.9.1 One Turbine

We apply the developed numerical Algorithm 5 to find the optimal operating points of NACA 23012 when the incoming wind speed is $9\frac{m}{s}$, $10\frac{m}{s}$ and $11\frac{m}{s}$. The results are tabulated in Table 3.3. From this table, the optimal pitch angle and the rotational speed are $\beta = 1.1215°$ and $\Omega = 1.684$ rad/s when the incoming wind speed is $9\frac{m}{s}$. At this optimal operating point, the generated power is $P = 20.5110$kw, the thrust is 5.6873 kilonewton, and the torque is 0.80744 kilonewton. These results are consistent with the field-tested results in [89]. Moreover, at this optimal operating point, the averaged angle of attack is $\alpha = 13.046°$ which yields high glide ratio, see Figure 3.15.

We now examine the performance of the developed numerical algorithm with different initial guesses. In this regard, we apply Algorithm 5 to find the optimal operating points of NACA 23012 using different initial guesses. The results are tabulated in Table 3.4. From this table, we observe that Algorithm 5 starting from the initial guesses $\beta_0 = 0°$ and $\Omega_0 = 1$ rad/s finds the optimal solutions of $\beta = 1.1213°$ and $\Omega = 1.6837$ rad/s when the incoming wind speed is $9\frac{m}{s}$. The magnitude of difference between these optimal points and those optimal points associated with the incoming wind speed of $9\frac{m}{s}$ in Table 3.3 is very small. Therefore, Algorithm 5 produces consistent optimal solutions, and it is not sensitive to the selected initial guesses used. Moreover, these results imply that the optimal operating points of a single turbine are unique.

We also investigate the efficiency of the developed numerical algorithm in terms of the number of objective function evaluations required for our algorithm to reach convergence. In this regard, the best objective function value, the number of objective function evaluations and the mesh size at each iteration of Algorithm 5 are plotted for $U_0 = 9$, $U_0 = 10$ and $U_0 = 11$ in Figures 3.17, 3.18 and 3.19. For

Table 3.3: Optimal operating point of NACA 23012.

Wind speed	init.	β	Ω	α	T	Ψ	P
U_0=9 $\frac{m}{s}$	(-3, 2)	1.1215	1.684	13.046	5.6873	0.80744	20.5110
U_0=10 $\frac{m}{s}$	(1, 1.9)	1.375	1.800	13.1870	6.5458	1.0100	30.0060
U_0=11 $\frac{m}{s}$	(2, 1)	1.4146	2.0500	13.4190	8.0906	1.2410	40.5448

example, from Figure 3.17, it is evident that Algorithm 5 converges to the stationary point 20.5110 as the sequence of the mesh size parameter tends to zero. From this figure, we also note that the number of objective function evaluations is 30. With this number of objective function evaluations, Algorithm 5 seems to find the optimal operating points of a single turbine in a reasonable time. However, as the number of turbine increases the objective function evaluation becomes substantially expensive. Therefore, for future work we are planning to develop a robust optimization algorithm to speed up the optimization process.

Finally, we examine the efficiency of Algorithm 3 in simulating the flow in ultimate wakes of a wind turbine. In this regard, we apply Algorithm 3 to simulate the flow around NACA 23012 operating at its optimal operating points $\beta = 1.375°$ and $\Omega = 1.800$ rad/s when the incoming wind speed is $10\frac{m}{s}$. The results are

Table 3.4: Optimal operating point of NACA 23012 (different initial point).

Wind speed	init.	β	Ω	α	T	Ψ	P
U_0=9 $\frac{m}{s}$	(0, 1)	1.1213	1.6837	13.0468	5.6874	0.80747	20.5113
U_0=10 $\frac{m}{s}$	(-2, 1)	1.3754	1.802	13.1881	6.5459	1.0102	30.0061
U_0=11 $\frac{m}{s}$	(-2, 1.7)	1.4160	2.0550	13.4120	8.0902	1.2403	40.5435

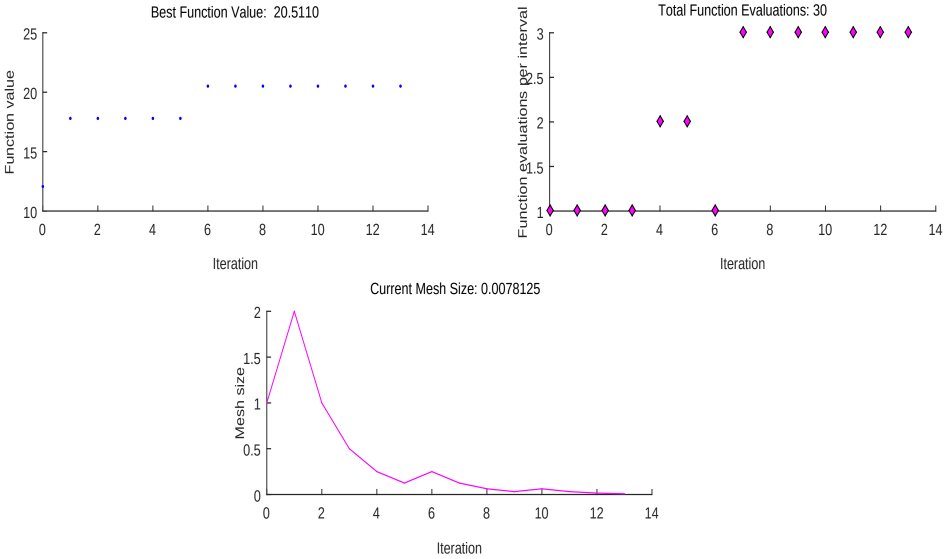


Figure 3.17: Pattern search result for laminar flow $U_0 = 9$.

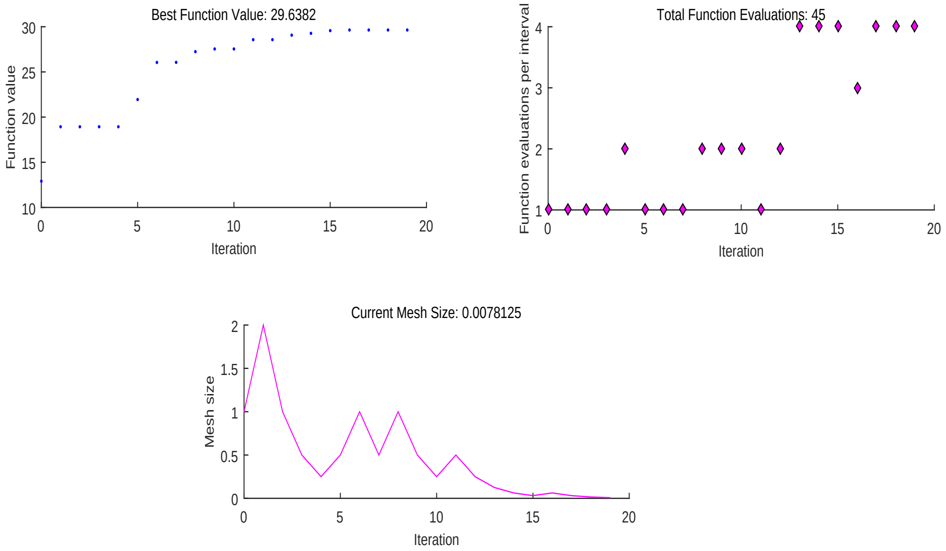


Figure 3.18: Pattern search result for laminar flow $U_0 = 10$.

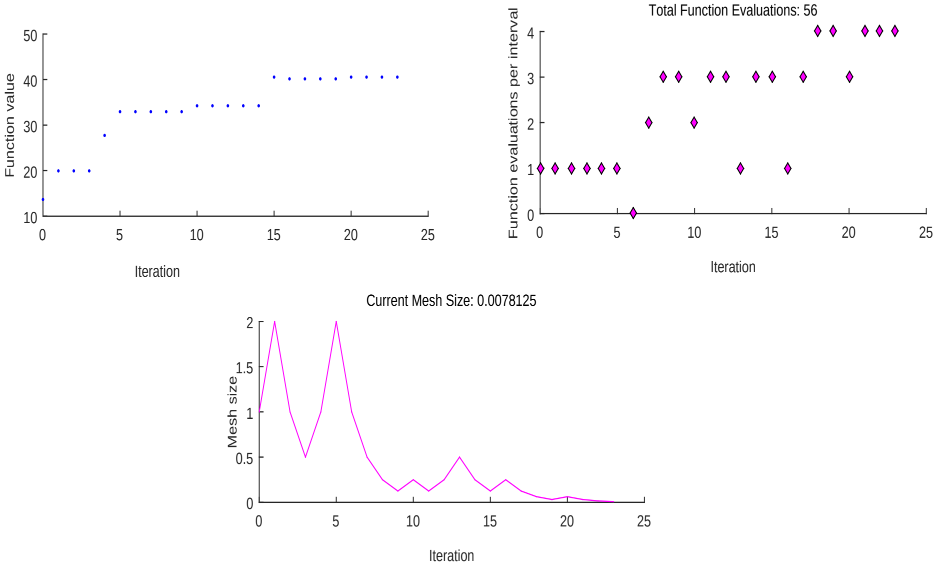


Figure 3.19: Pattern search result for laminar flow $U_0 = 11$

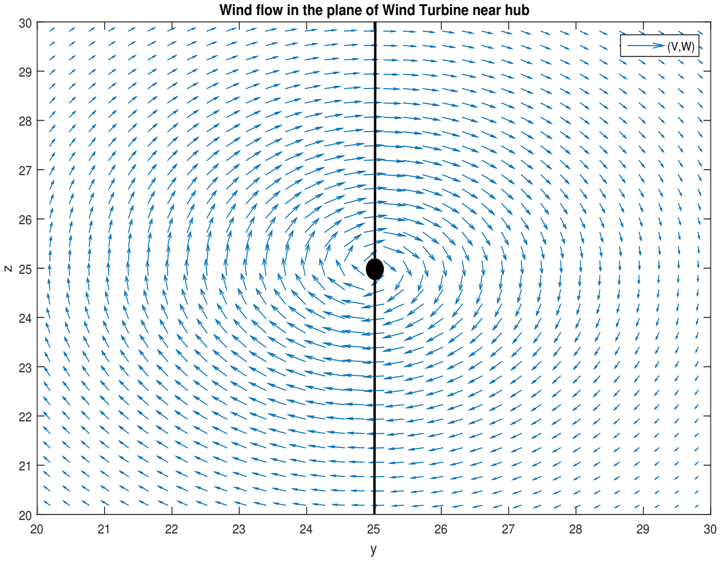


Figure 3.20: Wind flow in the plane of the rotor near the hub.

displayed in Figures 3.20, 3.21, 3.23 and 3.22. The flow patterns shown in these figures are consistent with the field-tested results in [89]. Moreover, from Figure 3.23, we notice that the wake of wind turbine continues to expand in the axial direction as well as in the z-direction as it develops further downstream. It is also important to note that, 6 rotor diameters away from the plane of the wind turbine, the wake has not yet fully recovered. Therefore, a wind turbine placed downstream of other wind turbines will experience diminished power production compared to free-standing wind turbines. In conclusion, in order to improve the performance of overall wind farm, it is essential to optimize the total power while considering the wake effect.

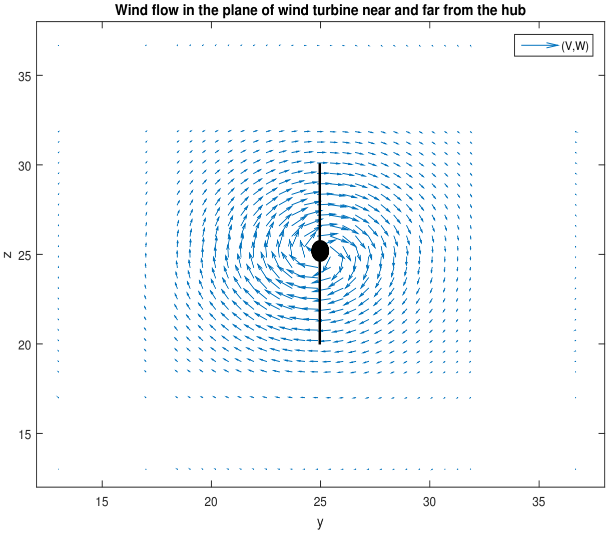


Figure 3.21: Wind flow in the plane of the rotor near and far from the hub.

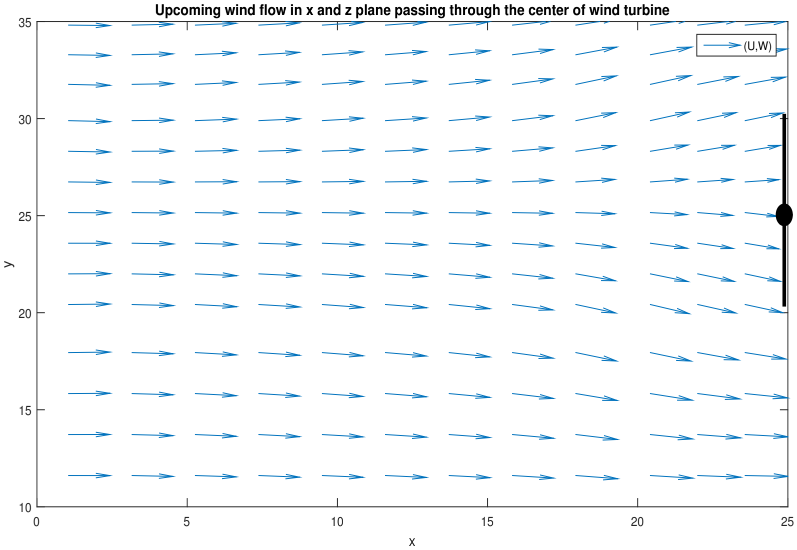


Figure 3.22: Wind flow in xz-plane passing through the hub.

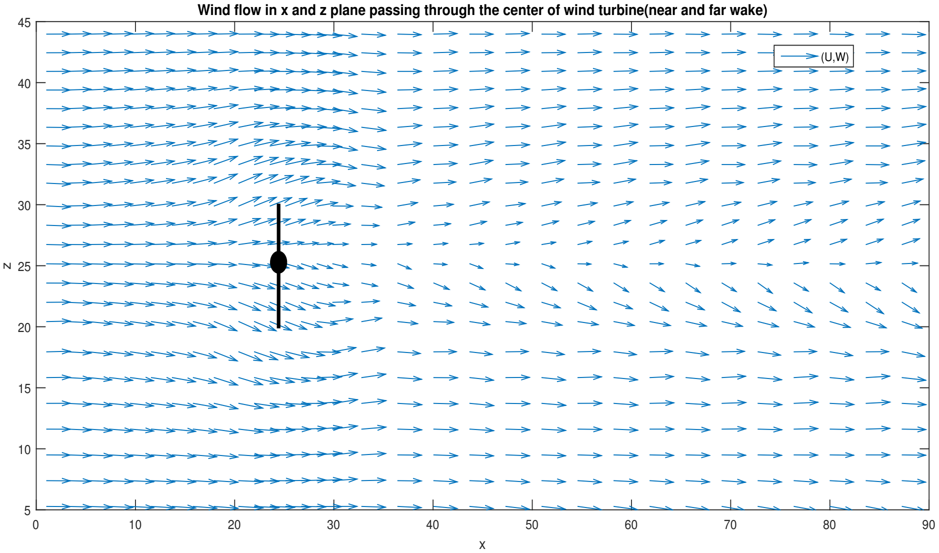


Figure 3.23: Wind flow in xz-plane passing through the hub.

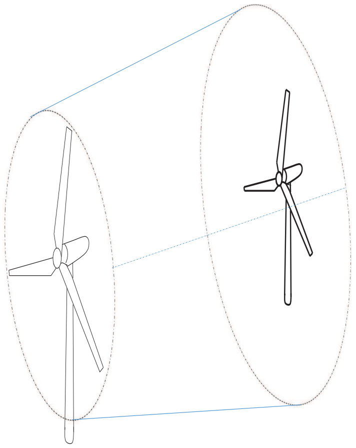


Figure 3.24: Schematic diagram of two turbines.

3.9.2 Two Turbines

The schematic diagram in Figure 3.24 shows two three-bladed turbines where the downstream turbine is positioned in the wake of the upstream turbine. These two turbines are four rotor diameters apart, and their blade sections consist of NACA 23012 series airfoils. We now apply the developed numerical Algorithm 5 to find the global optimum operating point of these two turbines where the wake of upstream turbine interfere with the downstream turbine. The results are tabulated in Table 3.5. From this table, the global optimum point of upstream and downstream turbine are $(\beta, \Omega) = (1.1646, 1.5915)$ and $(\beta, \Omega) = (0.7470, 1.3086)$, respectively. Moreover, from this table, we also note that the upstream turbine generates 18.575kw power and the downstream turbine generates 17.107kw power when they operate at their global optimum point. In this case, the total power output of wind turbines is 35.682kw. In contrast, when both turbines operate at their own optimum point, i.e., $(\beta, \Omega) = (1.1215, 1.6840)$, as shown in Table 3.3, then the upstream turbine generates 16.495kw power and the downstream turbine generates 16.226kw power. In this case, the total power output of wind turbines is 32.721kw. Therefore, by optimizing the turbines' operation while considering the wake effect, we can gain an additional 9.05% in the total power since $35.682/32.721 = 1.0905$. Moreover, we also note that this extra gain in the total power does not lead to increased loading on wind turbines. For example, from Table 3.5, the thrust of the downstream turbine operating at its global optimum point and in the wake of upstream turbine is $T = 5.2265$ kN which is less than the thrust of a free-standing turbine $T = 5.6873$ kN, see Table 3.3, operating at its own optimal point with no exposure to the wake of another turbine.

From Table 3.3, we note that a free-standing wind turbine operating at its optimum point $(\beta, \Omega) = (1.1215, 1.6840)$ generates 20.5110kw power when the incoming wind speed is $9\frac{m}{s}$. In contrast, if two turbines are grouped in a wind farm, and they operate at the optimum point of a free-standing turbine, then the upstream and downstream turbine generate 16.495kw and 16.226kw power, respectively. This power reduction of the downstream turbine is due to the wake generated by upstream wind turbine which alters the flow field and leads to a wind velocity deficit in the downstream wind turbine. The power reduction of the upstream wind turbine can be justified due to the thrust generated by the downstream turbine which reacts to the common flow field, and leads to a wind velocity deficit in the upstream wind turbine. In this perspective, the wake zone concept could be extended. It is not only the turbine at front affects the performance of the turbines at rear; rather, all turbines affect each other via the common flow field in which they are immersed.

Finally, from Table 3.3, we note that a free-standing wind turbine achieves its highest aerodynamic performance at the angle of attack 13.046 degrees. This angle of attack leads to a high glide ratio as shown in Figure 3.15. However, from Table 3.5, when two turbines are grouped in a wind farm they achieve their highest aerodynamic performance at the angle of attack higher than 13.046 degrees. For example, from Table 3.5, the angle of attack of the upstream and downstream turbine are 17.489 and 16.4512 degrees, respectively, when the incoming wind speed is $11\frac{m}{s}$, and they operate at the global optimum point. Moreover, from Table 3.5, it is evident that the angles of attack for all tested cases are consistently bigger, around 16 degrees. This angle of attack, 16 degrees, is the critical angle of attack for the lift coefficient, and it leads to higher lift coefficient, see Figure 3.14. In conclusion, the developed Algorithm 5 moves towards maximizing the lift coefficients when optimizing the total power.

3.10 Conclusion

In this chapter, we studied how to optimize power production of multiple wind turbines by considering the wake interactions among them. We modeled the intricate interference of multiple turbines through the actuator line method and the Navier-Stokes equations. We find that by optimizing the turbines' operation, we can safely gain an additional 8% in the total power. We find that not only the turbine at the front affects the production of a turbine at the rear, but all turbines affect each other by exerting forces into the flow field in which all turbines are immersed. In

Table 3.5: Joint optimal operating points of two NACA 23012.

Wind speed	turbine	β	Ω	α	T	Ψ	P
U_0=9 $\frac{m}{s}$	1st, joint	1.1646	1.5915	13.1200	5.1708	0.75555	18.575
	2nd, joint	0.7470	1.3086	22.3690	5.2265	0.75073	17.107
	1st	1.1215	1.6840	11.3001	5.1469	0.64332	16.495
	2nd	1.1215	1.6840	11.1630	5.1180	0.62954	16.226
					Gain:	9.04%	
U_0=10 $\frac{m}{s}$	1st, joint	0.5979	1.5252	15.1863	6.3486	0.98574	26.8790
	2nd, joint	0.082	1.3939	18.1510	6.3911	0.96538	25.8350
	1st	1.3750	1.8000	11.4591	5.9452	0.81589	24.5200
	2nd	1.3750	1.8000	11.3320	5.9133	0.80071	24.2460
					Gain:	8.11%	
U_0=11 $\frac{m}{s}$	1st, joint	0.331	1.7222	17.4891	8.2425	1.1991	35.7501
	2nd, joint	0.8275	1.6114	16.4512	7.2552	1.1972	34.6912
	1st	1.4160	2.0500	11.6740	7.3602	1.0041	33.3061
	2nd	1.4160	2.0500	11.5511	7.3233	0.9861	32.8601
					Gain:	7.3680%	

contrast to the case of a single turbine, the numerical results show that, at the optimal operating point for multiple wind turbines, the angle of attack does not yield the highest glide ratio. We also find that the optimal angle of attack deviates the most from the point where the glide ratio is maximum for the turbine at the rear. This work paves a way for a larger scale power production optimization and more accurate wind farm layout optimization.

4 Stochastic Power Optimization Using DD

4.1 Introduction

In the previous chapter, we optimize the total power in a wind farm assuming a deterministic value for the incoming wind. In this chapter, we relax this assumption and treat the incoming wind as a stochastic variable. We essentially focus on analyzing and modeling stochastic optimization of wind turbines' output in a wind farm subject to Navier-Stokes equations. Obtaining accurate simulation of wind flows governed by Navier-Stokes equations requires a great number of mesh points which can lead to the problem of solving large linear systems. Therefore, to obtain the solutions in a reasonable time, we develop a corrected-explicit-implicit domain decomposition scheme for the parallel approximation of the three-dimensional Navier-Stokes equations in a velocity-vorticity formulation. To validate the proposed parallel scheme, we apply it to estimate the solution of the lid-driven cavity problem which has been used as a benchmark problem for many numerical methods due to its simple geometry and complicated flow behaviors. To further validate the developed parallel scheme in the context of wind turbines' power production, we test our model against an experimental data that is available for a commercial wind turbine called WindSpot. We estimate the power generation of wind spot for different wind speeds using the parallel simulation of the Navier-Stokes equations and the actuator line method. It will be shown that our power estimates match the experimental data for the WindSpot very well. Since the results of our validation are acceptable, we proceed to apply the developed parallel scheme for the Navier-Stokes equations and the actuator line method in the context of stochastic optimization of wind turbines' output in a wind farm.

4.2 Modeling of the Flow for Stochastic Power Optimization

For a wind farm with N turbines, the production stochastic optimization model in its condensed form is the following:

$$\begin{aligned} &\underset{\substack{\beta_1,\ldots,\beta_N\\ \Omega_1,\ldots,\Omega_N}}{\text{Max}} \quad \sum_{i=1}^{N} E^{U_0}[P_i] \\ &\qquad\qquad \text{subject to}\,(4.2),\ \text{and} \\ &\qquad\qquad \beta_{\min} \le \beta_i \le \beta_{\max}, \\ &\qquad\qquad \Omega_{\min} \le \Omega_i \le \Omega_{\max}, \\ &\qquad\qquad \forall i \in \{1, 2, \ldots, N\}, \end{aligned} \tag{4.1}$$

where the incoming wind speed on the inlet boundary, U_0, is a random variable and the objective function is the expected total generated power from the wind farm. This optimization problem (4.1) is to select $2n$ decision variables, $(\beta_1,\Omega_1, \ldots, \beta_n,\Omega_n)$ in a 2n-dimensional rectangular box, where Ω_i is the speed at which the ith turbine rotates and β_i is the angle at which the blades of the ith turbine are being pitched. Moreover, these decision variables are bounded $\Omega_{\min} \leq \Omega_i \leq \Omega_{\max}$ and $\beta_{\min} \leq \beta_i \leq \beta_{\max}$ where $\Omega_{\min}$, $\Omega_{\max}$, $\beta_{\min}$ and $\beta_{\max}$ are the lower and upper bounds on the rotational speed and the pitch angle. These lower bounds are imposed due to the fact that if the wind turbine rotates slowly, or the blades are pitched at a low angle then most of the wind will pass unperturbed through the gaps between the blades which will result in power reduction. The upper bounds are imposed due to the fact that if the turbine turns too fast, or the blades are pitched at a high angle then it will act as a solid wall to the wind which will result in power reduction. Moreover, in the model (4.1), P_i is the power generated by the ith turbine that has a complicated nonlinear relationship with the decision variables and the wind speed which is the solution of the Navier-Stokes equations (4.2); it is derived in the previous chapter in section 3.6. Finally, in the model (4.1), the constraints are the Navier-Stokes equations which govern the relationship between the wind flow field and the external forces. These equations on a rectangular domain, $D = [0, l_x] \times [0, l_y] \times [0, l_z]$, are given by:

$$\begin{cases} \dfrac{\partial U}{\partial t} + UU_x + VU_y + Wu_z = -\dfrac{1}{\rho} p_x + \nu(U_{xx} + U_{yy} + U_{zz}) + f_u(t, x, y, z), \\ \dfrac{\partial V}{\partial t} + UV_x + VV_y + WV_z = -\dfrac{1}{\rho} p_y + \nu(V_{xx} + V_{yy} + V_{zz}) + f_v(t, x, y, z), \\ \dfrac{\partial W}{\partial t} + UW_x + VW_y + WW_z = -\dfrac{1}{\rho} p_z + \nu(W_{xx} + W_{yy} + W_{zz}) + f_w(t, x, y, z), \end{cases} \tag{4.2}$$

with the following boundary conditions:

$$\begin{cases} U(t, 0, \cdot, \cdot) = U_0, \quad V(t, 0, \cdot, \cdot) = 0, \quad W(t, 0, \cdot, \cdot) = 0, \\ U_x(t, l_x, \cdot, \cdot) = 0, \quad V_x(t, l_x, \cdot, \cdot) = 0, \quad W_x(t, l_x, \cdot, \cdot) = 0, \\ U_y(t, \cdot, 0, \cdot) = 0, \quad V_y(t, \cdot, 0, \cdot) = 0, \quad W_y(t, \cdot, 0, \cdot) = 0, \\ U_y(t, \cdot, l_y, \cdot) = 0, \quad V_y(t, \cdot, l_y, \cdot) = 0, \quad W_y(t, \cdot, l_y, \cdot) = 0, \\ U_z(t, \cdot, \cdot, 0) = 0, \quad V_z(t, \cdot, \cdot, 0) = 0, \quad W_z(t, \cdot, \cdot, 0) = 0, \\ U_z(t, \cdot, \cdot, l_z) = 0, \quad V_z(t, \cdot, \cdot, l_z) = 0, \quad W_z(t, \cdot, \cdot, l_z) = 0, \end{cases} \tag{4.3}$$

the initial conditions:

$$U(0, \cdot, \cdot, \cdot) = U_0, \quad V(t, \cdot, \cdot, \cdot) = 0, \quad W(t, \cdot, \cdot, \cdot) = 0, \tag{4.4}$$

and the continuity equation:

$$U_x + V_y + W_z = 0, \tag{4.5}$$

where ν is the viscosity, ρ is the air density and U_0 is the incoming wind speed which is a random variable.

4.3 Value of Stochastic Programming for Power Production

There are two approaches to find optimal operating points of wind turbines while considering the random wind speed on the boundary of the wind farm. The first approach is to solve a deterministic model, and the second approach is to solve a stochastic model. The first approach replaces the random incoming wind speed by its expectation, while the second approach explicitly includes the randomness of the incoming wind speed. Both approaches are aimed to find the optimal operating points of wind turbines. However, the second approach has advantages over the first one. Here, we illustrate the advantage of the stochastic model over the deterministic model through an example, and we show that ignoring the random characteristics of the incoming wind speed may limit the usefulness of the optimal solutions. For simplicity, we consider one turbine operating within the wind farm where the incoming wind speed has a discrete distribution with finite number of values $7\frac{m}{s}$, $8\frac{m}{s}$, $9\frac{m}{s}$, $10\frac{m}{s}$ and $11\frac{m}{s}$ that are equally likely to be observed. For these wind speeds, the power generated by the turbine for a fixed pitch angle and various rotational velocities are given in Figure 4.1. Here, we use the deterministic approach to find the optimal rotational speed. In this case, we assume the incoming wind speed is $9\frac{m}{s}$ which is the expectation of $7\frac{m}{s}$, $8\frac{m}{s}$, $9\frac{m}{s}$, $10\frac{m}{s}$ and $11\frac{m}{s}$. Since the incoming wind speed is $9\frac{m}{s}$, the optimal rotational velocity of the wind turbine is 1.684, see Table 3.3. Now, if we operate the wind turbine at this optimal point, then the expected power output is:

$$\frac{9.6531 + 15.2134 + 22.3351 + 29.4452 + 37.6782}{5} = 22.865, \tag{4.6}$$

where 9.911, 15.435, 22.409, 29.170 and 37.156 are the power generated by the turbine for the incoming wind speed of $7\frac{m}{s}$, $8\frac{m}{s}$, $9\frac{m}{s}$, $10\frac{m}{s}$ and $11\frac{m}{s}$, respectively. Therefore, if the wind turbine operates at the optimal rotational velocity obtained by the deterministic approach, then the expected power output will be 22.865kw. Now, we investigate the expected power output of the turbine at different rotational velocity using the power output curves, see Figure 4.1. From this figure, at rotational velocity 1.684, the power output curves for the incoming wind speed $10\frac{m}{s}$ and $11\frac{m}{s}$ are steep while other three curves are almost flat. Therefore, at rotational velocity slightly away from 1.684, the expected power output may increases. To find out, we compute the expected power output of the turbine operating at the rotational velocity 2.00:

$$\frac{9.0259 + 14.8221 + 22.996 + 30.0612 + 39.0965}{5} = 23.2003.$$

Therefore, if we operate the wind turbine at rotational velocity 2, then we gain 1.47% more power (since 23.2003=1.47*22.8650). Due to the simplifying assumption, this gain is small but it is large for the optimization model 4.1 as numerical results confirm this.

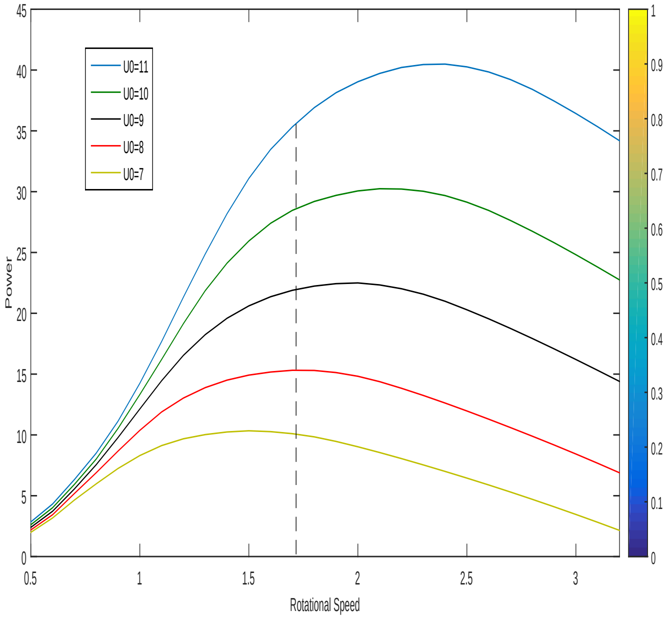


Figure 4.1: Generated power for various rotational velocity.

4.4 Scenario-Based Approximation Model

To solve the stochastic optimization model (4.1), we first need to model the random behavior of the incoming wind speed. In this regard, a large number of studies have been conducted to find a good probability distribution to model the randomness of the wind speed. Generally, the two-parameter Weibull distribution is widely used and accepted in the specialized literature on wind energy and other renewable energy sources [57, 79]. Here, we use the doubly truncated Weibull distribution with the following probability distribution function to model the random behavior of the incoming wind speed:

$$f(x) = \begin{cases} \dfrac{g(x)}{G(b) - G(a)} & a \leq x \leq b, \\ \\ 0 & \text{Otherwise,} \end{cases} \tag{4.7}$$

where g is the non-truncated Weibull distribution with shape parameter k_c and scale parameter s_c, and G is the cumulative distribution function. It can be shown that the expected value of the doubly truncated Weibull distribution is equal to:

$$\begin{aligned} &E_f(X) = \\ &\frac{s_c \exp((\frac{a}{s_c})^{k_c})}{1 - \exp(-(\frac{b}{s_c})^{k_c})} \Big(\gamma(\frac{1}{k_c} + 1, (\frac{b}{s_c})^{k_c}) - \gamma(\frac{1}{k_c} + 1, (\frac{a}{s_c})^{k_c})\Big), \end{aligned} \tag{4.8}$$

where γ is the incomplete gamma function.

To simulate the uncertainty of the incoming wind speed in the stochastic optimization model (4.1), we use the Monte Carlo simulation method. The advantage of applying this method is that the required number of samples for a given level of accuracy is independent of the size of the model (4.1). We apply the Monte Carlo simulation method to generate finite number of scenarios, $(U_0^1, U_0^2, \ldots, U_0^M)$, according to the doubly truncated Weibull distribution (4.8), where each scenario $U_0^j, j = 1, \ldots, M$ represents the incoming wind speed. The scenario generation process is summarized in Algorithm 4.9. Consequently, we obtain the following scenario-based approximation model to the stochastic optimization model (4.1):

$$\begin{aligned}
\underset{\substack{\beta_1,\ldots,\beta_N \\ \Omega_1,\ldots,\Omega_N}}{\text{Max}} \quad & \sum_{i=1}^{N} \frac{\sum_{j=1}^{M} P_i^j}{M} \\
& P_i^j,\ i = 1\ldots\ N,\ \text{subject to (4.2) under scenario } U_0^j \text{ for } j = 1\ldots M, \\
& \beta_{\min} \leq \beta_i \leq \beta_{\max}, \\
& \Omega_{\min} \leq \Omega_i \leq \Omega_{\max}, \\
& \forall i \in \{1, 2, \ldots, N\}.
\end{aligned} \tag{4.9}$$

Algorithm 6 Scenario generation

```
for $j = 1, 2, 3, \ldots, M$ do
    repeat
        $U_0^j = s_c(-\ln(1 - z_j))^{\frac{1}{k_c}}$ where $z_j$ is a random number generated from the uniform distribution.
    until $a \leq U_0^j \leq b$
end for
```

4.5 Parallel Scheme for Navier-Stokes Equations

By using the curl operator to the equations (4.2), one gets the following dimensionless form of the vorticity transport equations:

$$
\begin{aligned}
\frac{\partial \xi}{\partial t} &= -\frac{\partial (V\xi)}{\partial y} - \frac{\partial (W\xi)}{\partial z} + \frac{\partial (U\eta)}{\partial y} + \frac{\partial (U\zeta)}{\partial z} + \frac{1}{Re}(\frac{\partial^2 \xi}{\partial x^2} + \frac{\partial^2 \xi}{\partial y^2} + \frac{\partial^2 \xi}{\partial z^2}) \\
&+ \frac{\partial f_w}{\partial y} - \frac{\partial f_v}{\partial z}, \qquad (4.10) \\
\frac{\partial \eta}{\partial t} &= -\frac{\partial (U\eta)}{\partial x} - \frac{\partial (W\eta)}{\partial z} + \frac{\partial (V\xi)}{\partial x} + \frac{\partial (V\zeta)}{\partial z} + \frac{1}{Re}(\frac{\partial^2 \eta}{\partial x^2} + \frac{\partial^2 \eta}{\partial y^2} + \frac{\partial^2 \eta}{\partial z^2}) \\
&- \frac{\partial f_w}{\partial x} + \frac{\partial f_u}{\partial z}, \qquad (4.11) \\
\frac{\partial \zeta}{\partial t} &= -\frac{\partial (U\zeta)}{\partial x} - \frac{\partial (V\zeta)}{\partial y} + \frac{\partial (W\xi)}{\partial x} + \frac{\partial (W\eta)}{\partial y} + \frac{1}{Re}(\frac{\partial^2 \zeta}{\partial x^2} + \frac{\partial^2 \zeta}{\partial y^2} + \frac{\partial^2 \zeta}{\partial z^2}) \\
&+ \frac{\partial f_v}{\partial x} - \frac{\partial f_u}{\partial y}, \qquad (4.12)
\end{aligned}
$$

with the following boundary conditions:

$$
\begin{cases}
\xi(t,0,\cdot,\cdot) = \dfrac{\partial W(t,0,\cdot,\cdot)}{\partial y} - \dfrac{\partial V(t,0,\cdot,\cdot)}{\partial z}, \\
\xi(t,lx,\cdot,\cdot) = \dfrac{\partial W(t,lx,\cdot,\cdot)}{\partial y} - \dfrac{\partial V(t,lx,\cdot,\cdot)}{\partial z}, \\
\xi(t,\cdot,0,\cdot) = \dfrac{\partial W(t,\cdot,0,\cdot)}{\partial y} - \dfrac{\partial V(t,\cdot,0,\cdot)}{\partial z}, \\
\xi(t,\cdot,ly,\cdot) = \dfrac{\partial W(t,\cdot,ly,\cdot)}{\partial y} - \dfrac{\partial V(t,\cdot,ly,\cdot)}{\partial z}, \\
\xi(t,\cdot,\cdot,0) = \dfrac{\partial W(t,\cdot,\cdot,0)}{\partial y} - \dfrac{\partial V(t,\cdot,\cdot,0)}{\partial z}, \\
\xi(t,\cdot,\cdot,lz) = \dfrac{\partial W(t,\cdot,\cdot,lz)}{\partial y} - \dfrac{\partial V(t,\cdot,\cdot,lz)}{\partial z},
\end{cases} \qquad (4.13)
$$

$$\begin{cases} \eta(t,0,\cdot,\cdot) = -\dfrac{\partial W(t,0,\cdot,\cdot)}{\partial x} + \dfrac{\partial U(t,0,\cdot,\cdot)}{\partial z}, \\ \eta(t,lx,\cdot,\cdot) = -\dfrac{\partial W(t,lx,\cdot,\cdot)}{\partial x} + \dfrac{\partial U(t,lx,\cdot,\cdot)}{\partial z}, \\ \eta(t,\cdot,0,\cdot) = -\dfrac{\partial W(t,\cdot,0,\cdot)}{\partial x} + \dfrac{\partial U(t,\cdot,0,\cdot)}{\partial z}, \\ \eta(t,\cdot,ly,\cdot) = -\dfrac{\partial W(t,\cdot,ly,\cdot)}{\partial x} + \dfrac{\partial U(t,\cdot,ly,\cdot)}{\partial z}, \\ \eta(t,\cdot,\cdot,0) = -\dfrac{\partial W(t,\cdot,\cdot,0)}{\partial x} + \dfrac{\partial U(t,\cdot,\cdot,0)}{\partial z}, \\ \eta(t,\cdot,\cdot,lz) = -\dfrac{\partial W(t,\cdot,\cdot,lz)}{\partial x} + \dfrac{\partial U(t,\cdot,\cdot,lz)}{\partial z}, \end{cases} \tag{4.14}$$

$$\begin{cases} \zeta(t,0,\cdot,\cdot) = \dfrac{\partial V(t,0,\cdot,\cdot)}{\partial x} - \dfrac{\partial U(t,0,\cdot,\cdot)}{\partial y}, \\ \zeta(t,lx,\cdot,\cdot) = \dfrac{\partial V(t,lx,\cdot,\cdot)}{\partial x} - \dfrac{\partial U(t,lx,\cdot,\cdot)}{\partial y}, \\ \zeta(t,\cdot,0,\cdot) = \dfrac{\partial V(t,\cdot,0,\cdot)}{\partial x} - \dfrac{\partial U(t,\cdot,0,\cdot)}{\partial y}, \\ \zeta(t,\cdot,ly,\cdot) = \dfrac{\partial V(t,\cdot,ly,\cdot)}{\partial x} - \dfrac{\partial U(t,\cdot,ly,\cdot)}{\partial y}, \\ \zeta(t,\cdot,\cdot,0) = \dfrac{\partial V(t,\cdot,\cdot,0)}{\partial x} - \dfrac{\partial U(t,\cdot,\cdot,0)}{\partial y}, \\ \zeta(t,\cdot,\cdot,lz) = \dfrac{\partial V(t,\cdot,\cdot,lz)}{\partial x} - \dfrac{\partial U(t,\cdot,\cdot,lz)}{\partial y}. \end{cases} \tag{4.15}$$

By taking the curl of vorticity $(\xi, \eta, \zeta) = \nabla \times (U, V, W)$ and using the continuity equation (4.5), we obtain the following velocity Poisson equations:

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2} = -\frac{\partial(\zeta)}{\partial y} + \frac{\partial(\eta)}{\partial z}, \tag{4.16}$$

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = \frac{\partial(\zeta)}{\partial x} - \frac{\partial(\xi)}{\partial z}, \tag{4.17}$$

$$\frac{\partial^2 W}{\partial z^2} = -\frac{\partial^2(U)}{\partial z \partial x} - \frac{\partial^2(V)}{\partial z \partial y}, \tag{4.18}$$

with the following boundary conditions:

$$\begin{cases} U(t,0,\cdot,\cdot)=U_0, \quad V(t,0,\cdot,\cdot)=0, \quad W(t,0,\cdot,\cdot)=0, \\ U_x(t,l_x,\cdot,\cdot)=0, \quad V_x(t,l_x,\cdot,\cdot)=0, \quad W_x(t,l_x,\cdot,\cdot)=0, \\ U_y(t,\cdot,0,\cdot)=0, \quad V_y(t,\cdot,0,\cdot)=0, \quad W_y(t,\cdot,0,\cdot)=0, \\ U_y(t,\cdot,l_y,\cdot)=0, \quad V_y(t,\cdot,l_y,\cdot)=0, \quad W_y(t,\cdot,l_y,\cdot)=0, \\ U_z(t,\cdot,\cdot,0)=0, \quad V_z(t,\cdot,\cdot,0)=0, \quad W_z(t,\cdot,\cdot,0)=0, \\ U_z(t,\cdot,\cdot,l_z)=0, \quad V_z(t,\cdot,\cdot,l_z)=0, \quad W_z(t,\cdot,\cdot,l_z)=0, \end{cases} \tag{4.19}$$

and initial conditions:

$$u(0,\cdot,\cdot,\cdot)=U_0, v(t,\cdot,\cdot,\cdot)=0, w(t,\cdot,\cdot,\cdot)=0. \tag{4.20}$$

We seek for the numerical solution of $\xi(t,x,y,z)$, $\eta(t,x,y,z)$, $\zeta(t,x,y,z)$, $U(t,x,y,z)$, $V(t,x,y,z)$ and $W(t,x,y,z)$ in the computational domain $D \equiv [0,l_x]\times[0,l_y]\times[0,l_z]$ for t $\in$ $[0\ T]$ using parallel simulation. To find these numerical solutions, we first need to define the grid points in the computational domain. In this regard, we partition the computational domain using the following set of non-uniform grid points:

$$\begin{aligned} D_h = \{(x_i,y_j,z_k)\,|\, x_i &= x_{i-1}+h_x^i,\ 1\le i\le J_x, \\ y_j &= y_{j-1}+h_y^j,\ 1\le j\le J_y, \\ z_k &= z_{k-1}+h_z^k,\ 1\le k\le J_z,\} \end{aligned} \tag{4.21}$$

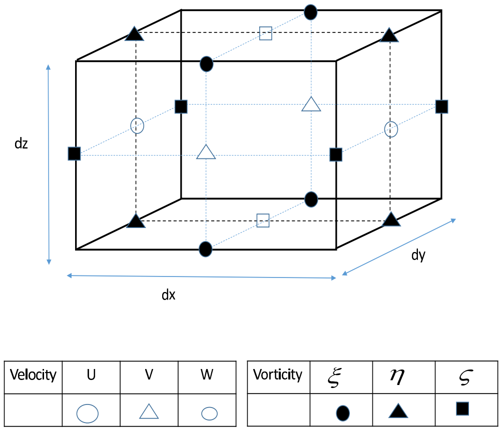


Figure 4.2: Staggered grid

where $x_0 = 0$, $y_0 = 0$, $z_0 = 0$, $x_{J_x} = l_x$, $y_{J_y} = l_y$ and $z_{J_z} = l_z$. We divide the time interval into N subinterval of equal length using the points, $t^n = n\tau$, n=0…N, where $\tau = \frac{T}{N}$. To discretize the velocity Poisson equations (4.16)-(4.18) and the vorticity transport equations (4.10)-(4.12), we use a MAC staggered grid system, as displayed in Figure 4.2, to ensure the accuracy in the prediction of velocities and vorticities. From this figure, we note that the numerical solution of U, V and W are evaluated at the grid points $(x_i, y_{j+\frac{1}{2}}, z_{k+\frac{1}{2}})$, $(x_{i+\frac{1}{2}}, y_j, z_{k+\frac{1}{2}})$ and $(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}, z_k)$, respectively. Moreover, from this figure, we also note that the numerical solution of ξ, η and ζ are computed at the grid points $(x_{i+\frac{1}{2}}, y_j, z_k)$, $(x_i, y_{j+\frac{1}{2}}, z_k)$ and $(x_i, y_j, z_{k+\frac{1}{2}})$, respectively. Before we proceed to a detailed description of the developed parallel scheme to solve the equations (4.16)-(4.18) and (4.10)-(4.12), we introduce some

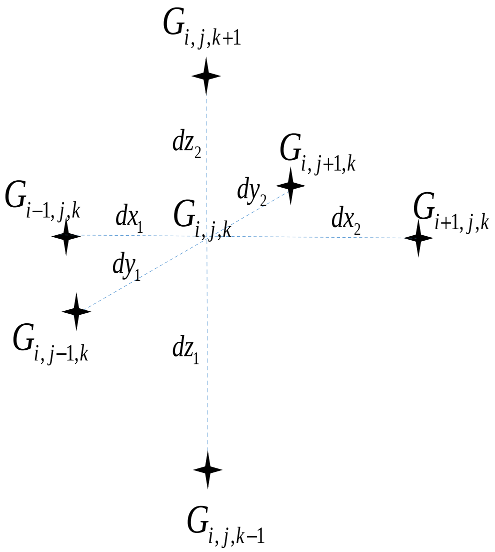


Figure 4.3: Arbitrary mesh function at an arbitrary grid point.

notations. For an arbitrary mesh function $G^n_{i,j,k} = G(t_n, x_i, y_j, z_k)$ defined at an arbitrary grid point (x_i, y_j, z_k), we introduce the following notations:

$$\partial_t G^n_{i,j,k} = \frac{G^n_{i,j,k} - G^{n-1}_{i,j,k}}{\tau}, \tag{4.22}$$

$$\delta_x G^n_{i-\frac{1}{2},j,k} = \frac{G^n_{i,j,k} - G^n_{i-1,j,k}}{dx_1}, \tag{4.23}$$

$$\delta_y G^n_{i,j-\frac{1}{2},k} = \frac{G^n_{i,j,k} - G^n_{i,j-1,k}}{dy_1}, \tag{4.24}$$

$$\delta_z G^n_{i,j,k-\frac{1}{2}} = \frac{G^n_{i,j,k} - G^n_{i,j,k-1}}{dz_1}, \tag{4.25}$$

and

$$\begin{aligned}\delta^2_x G^n_{i,j,k} = \frac{1}{dx_2(\frac{dx_1}{2} + \frac{dx_2}{2})} G^n_{i+1,j,k} &- (\frac{1}{dx_2(\frac{dx_1}{2} + \frac{dx_2}{2})} + \frac{1}{dx_1(\frac{dx_1}{2} + \frac{dx_2}{2})}) G^n_{i,j,k} \\ &+ \frac{1}{dx_1(\frac{dx_1}{2} + \frac{dx_2}{2})} G^n_{i-1,j,k},\end{aligned} \tag{4.26}$$

$$\begin{aligned}\delta_y^2 G_{i,j,k}^n = \frac{1}{dy_2(\frac{dy_1}{2}+\frac{dy_2}{2})}G_{i,j+1,k}^n &- (\frac{1}{dy_2(\frac{dy_1}{2}+\frac{dy_2}{2})}+\frac{1}{dy_1(\frac{dy_1}{2}+\frac{dy_2}{2})})G_{i,j,k}^n \\ &+ \frac{1}{dy_1(\frac{dy_1}{2}+\frac{dy_2}{2})}G_{i,j-1,k}^n, \qquad (4.27)\end{aligned}$$

$$\begin{aligned}\delta_z^2 G_{i,j,k}^n = \frac{1}{dz_2(\frac{dz_1}{2}+\frac{dz_2}{2})}G_{i,j,k+1}^n &- (\frac{1}{dz_2(\frac{dz_1}{2}+\frac{dz_2}{2})}+\frac{1}{dz_1(\frac{dz_1}{2}+\frac{dz_2}{2})})G_{i,j,k}^n \\ &+ \frac{1}{dz_1(\frac{dz_1}{2}+\frac{dz_2}{2})}G_{i,j,k-1}^n. \qquad (4.28)\end{aligned}$$

In the following subsections, we present the developed corrected-explicit-implicit domain decomposition scheme for the parallel approximation of the three-dimensional Navier-Stokes equations in a velocity-vorticity formulation. This is achieved by combining a second-order extrapolation scheme and an implicit correction technique. In the proposed scheme, at each time step, we predict the values of velocity at interface mesh points by a time second-order extrapolation scheme. Then, we approximate the interior values of velocity in each subdomain using the backward Euler method, an explicit scheme and an implicit scheme. Finally, once the subdomain solutions are available, we correct the predicted values of velocity at the interface mesh points. Using similar steps, we approximate the solutions of vorticity transport equations.

4.5.1 Parallel Approximation of U

Let $U^n_{i,j+\frac{1}{2},k+\frac{1}{2}}$ be the numerical approximation of U at the grid point $(x_i, y_{j+\frac{1}{2}}, z_{k+\frac{1}{2}})$ at time t^n, and D^u_h be the set of mesh points associated with $U^n_{i,j-\frac{1}{2},k-\frac{1}{2}}$:

$$D^u_h = \{(x_i, y_{j-\frac{1}{2}}, z_{k-\frac{1}{2}}) \mid i = 0, \ldots, J_x, \;\; j = 0, \ldots, J_y + 1, \;\; k = 0, \ldots, J_z + 1\}.$$

Decompose D^u_h into $(P^u + 1)(Q^u + 1)(S^u + 1)$ non-overlapping subdomains $D^u_{h\alpha,\beta,\gamma}$, $\alpha = 1 \ldots P^u + 1$, $\beta = 1 \ldots Q^u + 1$, $\gamma = 1 \ldots S^u + 1$:

$$D^u_{h\alpha,\beta,\gamma} = \{(x_i, y_{j-\frac{1}{2}}, z_{k-\frac{1}{2}}) \mid \; i^u_{\alpha-1} < i < i^u_\alpha, \quad j^u_{\beta-1} < j < j^u_\beta, \quad k^u_{\gamma-1} < k < k^u_\gamma\}.$$

In general, P and Q are related to the size of the problem and the number of processors in the computer platform. Associated to the subdomains, there are $(P^u + 1)(Q^u + 1)(S^u + 1)$ interfaces. Let $\Gamma^u_h = \Gamma^{u1}_h \bigcup \Gamma^{u2}_h \bigcup \Gamma^{u3}_h$ be the set of all mesh points on the interfaces where Γ^{u1}_h, Γ^{u2}_h and Γ^{u3}_h are given by:

$$\begin{aligned}
\Gamma^{u1}_h &= \{(x_{i^u_\alpha}, y_{j-\frac{1}{2}}, z_{k-\frac{1}{2}}) \mid 4 \le i^u_\alpha + 2 \le i^u_{\alpha+1}, \;\; 1 \le \alpha \le P^u, \;\; 0 < j < (J_y + 1), \\
&\qquad 0 < k < (J_z + 1)\}, \\
\Gamma^{u2}_h &= \{(x_i, y_{j^u_\beta - \frac{1}{2}}, z_{k-\frac{1}{2}}) \mid 4 \le j^u_\beta + 2 \le j^u_{\beta+1}, \;\; 1 \le \beta \le Q^u, \;\; 1 \le i \le (J_x - 1), \\
&\qquad 0 < k < (J_z + 1)\}, \\
\Gamma^{u3}_h &= \{(x_i, y_{j-\frac{1}{2}}, z_{k^u_\gamma - \frac{1}{2}}) \mid 4 \le k^u_\gamma + 2 \le k^u_{\gamma+1}, \;\; 1 \le \gamma \le S^u, \;\; 1 \le i \le (J_x - 1), \\
&\qquad 0 < j < (J_y + 1)\},
\end{aligned} \tag{4.29}$$

where $i_0^u = 0$, $i_{P+1}^u = J_x$, $j_0^u = 0$, $j_{Q+1}^u = J_y + 1$, $k_0^u = 0$ and $k_{S+1}^u = J_z + 1$. Now, we propose a corrected-explicit-implicit domain decomposition scheme over multi-block subdomains for the parallel approximation of the velocity Poisson equation (4.16) which consists of the following steps:

Step 1. Apply the following explicit scheme to predict the value of $U^n_{i,j-\frac{1}{2},k-\frac{1}{2}}$ at the interface mesh points by:

$$\tilde{U}^n_{i,j-\frac{1}{2},k-\frac{1}{2}} = 2U^{n-1}_{i,j-\frac{1}{2},k-\frac{1}{2}} - U^{n-2}_{i,j-\frac{1}{2},k-\frac{1}{2}} \qquad \text{on } \Gamma_h^u. \tag{4.30}$$

Step 2. Compute the value of $U^n_{i,j-\frac{1}{2},k-\frac{1}{2}}$ at the interior points of subdomains by:

$$\begin{cases} \dfrac{U^n_{i,j-\frac{1}{2},k-\frac{1}{2}} - U^{n-1}_{i,j-\frac{1}{2},k-\frac{1}{2}}}{\tau} = \delta_x^2 U^n_{i,j-\frac{1}{2},k-\frac{1}{2}} + \delta_y^2 U^n_{i,j-\frac{1}{2},k-\frac{1}{2}} + \delta_z^2 U^n_{i,j-\frac{1}{2},k-\frac{1}{2}} \\ \qquad + \delta_y \zeta^{n-1}_{i,j-\frac{1}{2},k-\frac{1}{2}} - \delta_z \eta^{n-1}_{i,j-\frac{1}{2},k-\frac{1}{2}} \quad \text{on } D_h^u \backslash \{\Gamma_h^u \cup \partial D_h^u\}, \\ U^n_{i,j-\frac{1}{2},k-\frac{1}{2}} = \tilde{U}^n_{i,j-\frac{1}{2},k-\frac{1}{2}} \qquad \text{on } \Gamma_h^u \ . \end{cases} \tag{4.31}$$

Step 3. Correct the predicted value of $U^n_{i,j-\frac{1}{2},k-\frac{1}{2}}$ at the interface mesh points $\Gamma_h^u \backslash \{\Gamma_h^{u1} \cap \Gamma_h^{u2},\, \Gamma_h^{u1} \cap \Gamma_h^{u3},\, \Gamma_h^{u2} \cap \Gamma_h^{u3}\}$ by:

$$\begin{cases} \dfrac{U^n_{i,j-\frac{1}{2},k-\frac{1}{2}} - U^{n-1}_{i,j-\frac{1}{2},k-\frac{1}{2}}}{\tau} = \delta_x^2 U^n_{i,j-\frac{1}{2},k-\frac{1}{2}} + \delta_y^2 U^n_{i,j-\frac{1}{2},k-\frac{1}{2}} + \delta_z^2 U^n_{i,j-\frac{1}{2},k-\frac{1}{2}} \\ \qquad + \delta_y \zeta^{n-1}_{i,j-\frac{1}{2},k-\frac{1}{2}} - \delta_z \eta^{n-1}_{i,j-\frac{1}{2},k-\frac{1}{2}} \\ \qquad\qquad \text{on } \Gamma_h^u \backslash \{\Gamma_h^{u1} \cap \Gamma_h^{u2},\, \Gamma_h^{u1} \cap \Gamma_h^{u3},\, \Gamma_h^{u2} \cap \Gamma_h^{u3}\}, \\ U^n_{i,j-\frac{1}{2},k-\frac{1}{2}} = \tilde{U}^n_{i,j-\frac{1}{2},k-\frac{1}{2}} \quad \text{on } \{\Gamma_h^{u1} \cap \Gamma_h^{u2},\, \Gamma_h^{u1} \cap \Gamma_h^{u3},\, \Gamma_h^{u2} \cap \Gamma_h^{u3}\} \ . \end{cases} \tag{4.32}$$

Step 4. Correct the predicted value of $U^n_{i,j-\frac{1}{2},k-\frac{1}{2}}$ at the interface mesh points $\{\Gamma_h^{u1} \cap \Gamma_h^{u2},\, \Gamma_h^{u1} \cap \Gamma_h^{u3},\, \Gamma_h^{u2} \cap \Gamma_h^{u3}\}$ by:

$$\begin{cases} \dfrac{U^n_{i,j-\frac{1}{2},k-\frac{1}{2}} - U^{n-1}_{i,j-\frac{1}{2},k-\frac{1}{2}}}{\tau} = \delta_x^2 U^n_{i,j-\frac{1}{2},k-\frac{1}{2}} + \delta_y^2 U^n_{i,j-\frac{1}{2},k-\frac{1}{2}} + \delta_z^2 U^n_{i,j-\frac{1}{2},k-\frac{1}{2}} \\ \qquad\qquad + \delta_y \zeta^{n-1}_{i,j-\frac{1}{2},k-\frac{1}{2}} - \delta_z \eta^{n-1}_{i,j-\frac{1}{2},k-\frac{1}{2}} \\ \qquad\qquad\qquad \text{on } \{\Gamma_h^{u1} \cap \Gamma_h^{u2},\, \Gamma_h^{u1} \cap \Gamma_h^{u3},\, \Gamma_h^{u2} \cap \Gamma_h^{u3}\}, \\ U^n_{i,j-\frac{1}{2},k-\frac{1}{2}} = \tilde{U}^n_{i,j-\frac{1}{2},k-\frac{1}{2}} \quad \text{on} \quad \{\Gamma_h^{u1} \cap \Gamma_h^{u2} \cap \Gamma_h^{u3}\}\ . \end{cases} \tag{4.33}$$

Step 5. Correct the predicted value of $U^n_{i,j-\frac{1}{2},k-\frac{1}{2}}$ at the interface mesh points $\{\Gamma_h^{u1} \cap \Gamma_h^{u2} \cap \Gamma_h^{u3}\}$ by:

$$\begin{cases} \dfrac{U^n_{i,j-\frac{1}{2},k-\frac{1}{2}} - U^{n-1}_{i,j-\frac{1}{2},k-\frac{1}{2}}}{\tau} = \delta_x^2 U^n_{i,j-\frac{1}{2},k-\frac{1}{2}} + \delta_y^2 U^n_{i,j-\frac{1}{2},k-\frac{1}{2}} + \delta_z^2 U^n_{i,j-\frac{1}{2},k-\frac{1}{2}} \\ \qquad\qquad + \delta_y \zeta^{n-1}_{i,j-\frac{1}{2},k-\frac{1}{2}} - \delta_z \eta^{n-1}_{i,j-\frac{1}{2},k-\frac{1}{2}} \quad \text{on } \{\Gamma_h^{u1} \cap \Gamma_h^{u2} \cap \Gamma_h^{u3}\}. \end{cases} \tag{4.34}$$

The boundary conditions are given in (4.19) and the first level values $U^1_{i,j-\frac{1}{2},k-\frac{1}{2}}$ are obtained without domain decomposition scheme.

4.5.2 Parallel Approximation of V

Assume that $V^n_{i+\frac{1}{2},j,k+\frac{1}{2}}$ approximates the exact solution, V, at $(x_{i+\frac{1}{2}}, y_j, z_{k+\frac{1}{2}})$ at time t^n, and assume D_h^v is the set of mesh points associated with $V^n_{i-\frac{1}{2},j,k-\frac{1}{2}}$:

$$D_h^v = \{(x_{i-\frac{1}{2}}, y_j, z_{k-\frac{1}{2}}) \mid i = 0, \ldots, J_x + 1,\;\; j = 0, \ldots, J_y,\;\; k = 0, \ldots, J_z + 1\}.$$

Divide D_h^v into $(P^v+1)(Q^v+1)(S^v+1)$ non-overlapping subdomains $D^v_{h\alpha,\beta,\gamma}$, $\alpha = 1\ldots P^v+1$, $\beta = 1\ldots Q^v+1$, $\gamma = 1\ldots S^v+1$:

$$D^v_{h\alpha,\beta,\gamma} = \{(x_{i-\frac{1}{2}}, y_j, z_{k-\frac{1}{2}}) \mid \; i^v_{\alpha-1} < i < i^v_\alpha, \quad j^v_{\beta-1} < j < j^v_\beta, \quad k^v_{\gamma-1} < k < k^v_\gamma\},$$

Let $\Gamma_h^v = \Gamma_h^{v1} \bigcup \Gamma_h^{v2} \bigcup \Gamma_h^{v3}$ be the set of interface mesh points where:

$$\begin{aligned}
\Gamma_h^{v1} &= \{(x_{i^v_\alpha-\frac{1}{2}}, y_j, z_{k-\frac{1}{2}}) \mid 4 \le i^v_\alpha + 2 \le i^v_{\alpha+1}, \;\; 1 \le \alpha \le P^v, \;\; 0 < j < J_y, \\
&\qquad 0 < k < (J_z+1)\}, \\
\Gamma_h^{v2} &= \{(x_{i-\frac{1}{2}}, y_{j^v_\beta}, z_{k-\frac{1}{2}}) \mid 4 \le j^v_\beta + 2 \le j^v_{\beta+1}, \;\; 1 \le \beta \le Q^v, \;\; 0 < i < (J_x+1), \\
&\qquad 0 < k < (J_z+1)\}, \\
\Gamma_h^{v3} &= \{(x_{i-\frac{1}{2}}, y_j, z_{k^v_\gamma-\frac{1}{2}}) \mid 4 \le k^v_\gamma + 2 \le k^v_{\gamma+1}, \;\; 1 \le \gamma \le S^v, \;\; 0 < i < (J_x+1), \\
&\qquad 0 < j < J_y\}.
\end{aligned} \tag{4.35}$$

Here, $i^v_0 = 0$, $i^v_{P+1} = J_x + 1$, $j^v_0 = 0$, $j^v_{Q+1} = J_y$, $k^v_0 = 0$ and $k^v_{S+1} = J_z + 1$.

It is worth to mention that, in the parallel approximation of velocity Poisson equations, we use false transient method which is an alternative technique to solve the steady-state problems. In this approach, instead of solving the steady-state problem directly, the relevant transient problem is solved until the solution no longer varies with the time [7, 32, 56, 63]. Moreover, central finite differencing scheme is used to approximate the second-order derivatives that leads to a large linear system to be solved. Here, we use Generalized Minimum Residual technique, see Algorithm 4, but in a sparse format [75]. Now, we propose the domain decomposition scheme which is used for the parallel approximation of (4.17). It is summarized in the following:

Step 1. Predict the value of $V^n_{i-\frac{1}{2},j,k-\frac{1}{2}}$ at the interface mesh points Γ^v_h by:

$$\tilde{V}^n_{i-\frac{1}{2},j,k-\frac{1}{2}} = 2V^{n-1}_{i-\frac{1}{2},j,k-\frac{1}{2}} - V^{n-2}_{i-\frac{1}{2},j,k-\frac{1}{2}}. \tag{4.36}$$

Step 2. Find $V^n_{i-\frac{1}{2},j,k-\frac{1}{2}}$ at the interior points of subdomains by:

$$\begin{cases} \dfrac{V^n_{i-\frac{1}{2},j,k-\frac{1}{2}} - V^{n-1}_{i-\frac{1}{2},j,k-\frac{1}{2}}}{\tau} = \delta^2_x V^n_{i-\frac{1}{2},j,k-\frac{1}{2}} + \delta^2_y V^n_{i-\frac{1}{2},j,k-\frac{1}{2}} + \delta^2_z V^n_{i-\frac{1}{2},j,k-\frac{1}{2}} \\ \qquad - \delta_x \zeta^{n-1}_{i-\frac{1}{2},j,k-\frac{1}{2}} + \delta_z \xi^{n-1}_{i-\frac{1}{2},j,k-\frac{1}{2}} \quad \text{on } D^V_h \backslash \{\Gamma^v_h \cup \partial D^v_h\}, \\ V^n_{i-\frac{1}{2},j,k-\frac{1}{2}} = \tilde{V}^n_{i-\frac{1}{2},j,k-\frac{1}{2}} \quad \text{on } \Gamma^v_h \,. \end{cases} \tag{4.37}$$

Step 3. Correct the predicted value of $V^n_{i-\frac{1}{2},j,k-\frac{1}{2}}$ at the interface mesh points $\Gamma^v_h \backslash \{\Gamma^{v1}_h \cap \Gamma^{v2}_h, \, \Gamma^{v1}_h \cap \Gamma^{v3}_h, \, \Gamma^{v2}_h \cap \Gamma^{v3}_h\}$ by:

$$\begin{cases} \dfrac{V^n_{i-\frac{1}{2},j,k-\frac{1}{2}} - V^{n-1}_{i-\frac{1}{2},j,k-\frac{1}{2}}}{\tau} = \delta^2_x V^n_{i-\frac{1}{2},j,k-\frac{1}{2}} + \delta^2_y V^n_{i-\frac{1}{2},j,k-\frac{1}{2}} + \delta^2_z V^n_{i-\frac{1}{2},j,k-\frac{1}{2}} \\ \qquad - \delta_x \zeta^{n-1}_{i-\frac{1}{2},j,k-\frac{1}{2}} + \delta_z \xi^{n-1}_{i-\frac{1}{2},j,k-\frac{1}{2}} \\ \qquad\qquad \text{on } \Gamma^v_h \backslash \{\Gamma^{v1}_h \cap \Gamma^{v2}_h, \, \Gamma^{v1}_h \cap \Gamma^{v3}_h, \, \Gamma^{v2}_h \cap \Gamma^{v3}_h\}, \\ V^n_{i-\frac{1}{2},j,k-\frac{1}{2}} = \tilde{V}^n_{i-\frac{1}{2},j,k-\frac{1}{2}} \quad \text{on } \{\Gamma^{v1}_h \cap \Gamma^{v2}_h, \, \Gamma^{v1}_h \cap \Gamma^{v3}_h, \, \Gamma^{v2}_h \cap \Gamma^{v3}_h\}. \end{cases} \tag{4.38}$$

Step 4. Correct the predicted value of $V^n_{i-\frac{1}{2},j,k-\frac{1}{2}}$ at the interface mesh points $\{\Gamma^{v1}_h \cap$

Γ_h^{v2}, $\Gamma_h^{v1} \cap \Gamma_h^{v3}$, $\Gamma_h^{v2} \cap \Gamma_h^{v3}\}$ by:

$$\begin{cases} \dfrac{V^n_{i-\frac{1}{2},j,k-\frac{1}{2}} - V^{n-1}_{i-\frac{1}{2},j,k-\frac{1}{2}}}{\tau} = \delta_x^2 V^n_{i-\frac{1}{2},j,k-\frac{1}{2}} + \delta_y^2 V^n_{i-\frac{1}{2},j,k-\frac{1}{2}} + \delta_z^2 V^n_{i-\frac{1}{2},j,k-\frac{1}{2}} \\ \qquad - \delta_x \zeta^{n-1}_{i-\frac{1}{2},j,k-\frac{1}{2}} + \delta_z \xi^{n-1}_{i-\frac{1}{2},j,k-\frac{1}{2}} \\ \qquad\qquad \text{on} \quad \left\{\Gamma_h^{v1} \cap \Gamma_h^{v2},\, \Gamma_h^{v1} \cap \Gamma_h^{v3},\, \Gamma_h^{v2} \cap \Gamma_h^{v3}\right\}, \\ V^n_{i-\frac{1}{2},j,k-\frac{1}{2}} = \tilde{V}^n_{i-\frac{1}{2},j,k-\frac{1}{2}} \quad \text{on} \quad \left\{\Gamma_h^{v1} \cap \Gamma_h^{v2} \cap \Gamma_h^{v3}\right\}. \end{cases} \tag{4.39}$$

Step 5. Correct the predicted value of $V^n_{i-\frac{1}{2},j,k-\frac{1}{2}}$ at the interface mesh points $\{\Gamma_h^{v1} \cap \Gamma_h^{v2} \cap \Gamma_h^{v3}\}$ by:

$$\begin{cases} \dfrac{V^n_{i-\frac{1}{2},j,k-\frac{1}{2}} - V^{n-1}_{i-\frac{1}{2},j,k-\frac{1}{2}}}{\tau} = \delta_x^2 V^n_{i-\frac{1}{2},j,k-\frac{1}{2}} + \delta_y^2 V^n_{i-\frac{1}{2},j,k-\frac{1}{2}} + \delta_z^2 V^n_{i-\frac{1}{2},j,k-\frac{1}{2}} \\ \qquad - \delta_x \zeta^{n-1}_{i-\frac{1}{2},j,k-\frac{1}{2}} + \delta_z \xi^{n-1}_{i-\frac{1}{2},j,k-\frac{1}{2}} \quad \text{on} \ \{\Gamma_h^{v1} \cap \Gamma_h^{v2} \cap \Gamma_h^{v3}\}. \end{cases} \tag{4.40}$$

In the parallel approximation of equations (4.16), (4.17) and (4.18), we use the boundary conditions given in (4.19). Moreover, we find the solutions of U, V and W at time $t = 1$ without domain decomposition scheme.

4.5.3 Parallel Approximation of W

Let $W^n_{i+\frac{1}{2},j+\frac{1}{2},k}$ be the numerical solution of W at the grid point $(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}, z_k)$ at time t^n, and let D_h^w be the set of mesh points associated with $W^n_{i-\frac{1}{2},j-\frac{1}{2},k}$:

$$D_h^w = \{(x_{i-\frac{1}{2}}, y_{j-\frac{1}{2}}, z_k) \mid i = 0, \ldots, J_x + 1, \;\; j = 0, \ldots, J_y + 1, \;\; k = 0, \ldots, J_z\}.$$

Decompose D_h^w into $(P^w+1)(Q^w+1)(S^w+1)$ non-overlapping sub-domains where the grid points in the subdomains are given by:

$$D^w_{h\alpha,\beta,\gamma} = \{(x_{i-\frac{1}{2}}, y_{j-\frac{1}{2}}, z_k) \mid \; i^w_{\alpha-1} < i < i^w_{\alpha}, \quad j^w_{\beta-1} < j < j^w_{\beta} \quad k^w_{\gamma-1} < k < k^w_{\gamma}\}.$$

Let Γ_h^w be the set of all mesh points on the interfaces, that is, $\Gamma_h^w = \Gamma_h^{w1} \bigcup \Gamma_h^{w2} \bigcup \Gamma_h^{w3}$, where:

$$\begin{aligned}
\Gamma_h^{w1} &= \{(x_{i^w_\alpha - \frac{1}{2}}, y_{j-\frac{1}{2}}, z_k) \mid 4 \le i^w_\alpha + 2 \le i^w_{\alpha+1}, \;\; 1 \le \alpha \le P^w, \;\; 0 < j < (J_y+1), \\
&\qquad 0 < k < J_z\}, \\
\Gamma_h^{w2} &= \{(x_{i-\frac{1}{2}}, y_{j^w_\beta - \frac{1}{2}}, z_k) \mid 4 \le j^w_\beta + 2 \le j^w_{\beta+1}, \;\; 1 \le \beta \le Q^w, \;\; 0 < i < (J_x+1), \\
&\qquad 0 < k < J_z\}, \\
\Gamma_h^{w3} &= \{(x_{i-\frac{1}{2}}, y_{j-\frac{1}{2}}, z_{k^w_\gamma}) \mid 4 \le k^w_\gamma + 2 \le k^w_{\gamma+1}, \;\; 1 \le \gamma \le S^w, \;\; 0 < i < (J_x+1), \\
&\qquad 0 < j < (J_y+1)\}.
\end{aligned} \tag{4.41}$$

Here, $i_0^w = 0$, $i^w_{P+1} = J_x + 1$, $j^w_0 = 0$, $j^w_{Q+1} = J_y + 1$, $k^w_0 = 0$ and $k^w_{S+1} = J_z$. We approximate the solution of equation (4.18) in parallel by the following domain decomposition scheme:

Step 1. Predict the value of $W^n_{i-\frac{1}{2},j-\frac{1}{2},k}$ at the interface mesh points Γ_h^w by:

$$\tilde{W}^n_{i-\frac{1}{2},j-\frac{1}{2},k} = 2W^{n-1}_{i-\frac{1}{2},j-\frac{1}{2},k} - W^{n-2}_{i-\frac{1}{2},j-\frac{1}{2},k}. \tag{4.42}$$

Step 2. Find $W^n_{i-\frac{1}{2},j-\frac{1}{2},k}$ at the interior points of subdomains by:

$$\begin{cases} \dfrac{W^n_{i-\frac{1}{2},j-\frac{1}{2},k} - W^{n-1}_{i-\frac{1}{2},j-\frac{1}{2},k}}{\tau} = \delta_z^2 W^n_{i-\frac{1}{2},j-\frac{1}{2},k} + \delta_z\delta_x U^n_{i-\frac{1}{2},j-\frac{1}{2},k} \\ \qquad + \delta_z\delta_y V^n_{i-\frac{1}{2},j-\frac{1}{2},k} \quad \text{on } D_h^w \backslash \{\Gamma_h^w \cup \partial D_h^w\}, \\ W^n_{i-\frac{1}{2},j-\frac{1}{2},k} = \tilde{W}^n_{i-\frac{1}{2},j-\frac{1}{2},k} \qquad \text{on } \Gamma_h^w. \end{cases} \tag{4.43}$$

Step 3. Correct the predicted value of $W^n_{i-\frac{1}{2},j-\frac{1}{2},k}$ at the interface mesh points $\Gamma_h^w \backslash \{\Gamma_h^{w1} \cap \Gamma_h^{w2},\, \Gamma_h^{w1} \cap \Gamma_h^{w3},\, \Gamma_h^{w2} \cap \Gamma_h^{w3}\}$ by:

$$\begin{cases} \dfrac{W^n_{i-\frac{1}{2},j-\frac{1}{2},k} - W^{n-1}_{i-\frac{1}{2},j-\frac{1}{2},k}}{\tau} = \delta_z^2 W^n_{i-\frac{1}{2},j-\frac{1}{2},k} + \delta_z\delta_x U^n_{i-\frac{1}{2},j-\frac{1}{2},k} + \delta_z\delta_y V^n_{i-\frac{1}{2},j-\frac{1}{2},k} \\ \qquad \text{on } \Gamma_h^w \backslash \{\Gamma_h^{w1} \cap \Gamma_h^{w2},\, \Gamma_h^{w1} \cap \Gamma_h^{w3},\, \Gamma_h^{w2} \cap \Gamma_h^{w3}\}, \\ W^n_{i-\frac{1}{2},j-\frac{1}{2},k} = \tilde{W}^n_{i-\frac{1}{2},j-\frac{1}{2},k} \text{ on } \{\Gamma_h^{w1} \cap \Gamma_h^{w2},\, \Gamma_h^{w1} \cap \Gamma_h^{w3},\, \Gamma_h^{w2} \cap \Gamma_h^{w3}\}. \end{cases} \tag{4.44}$$

Step 4. Correct the predicted value of $W^n_{i-\frac{1}{2},j-\frac{1}{2},k}$ at the interface mesh points $\{\Gamma_h^{w1} \cap \Gamma_h^{w2},\, \Gamma_h^{w1} \cap \Gamma_h^{w3},\, \Gamma_h^{w2} \cap \Gamma_h^{w3}\}$ by:

$$\begin{cases} \dfrac{W^n_{i-\frac{1}{2},j-\frac{1}{2},k} - W^{n-1}_{i-\frac{1}{2},j-\frac{1}{2},k}}{\tau} = \delta_z^2 W^n_{i-\frac{1}{2},j-\frac{1}{2},k} + \delta_z\delta_x U^n_{i-\frac{1}{2},j-\frac{1}{2},k} + \delta_z\delta_y V^n_{i-\frac{1}{2},j-\frac{1}{2},k} \\ \qquad \text{on } \{\Gamma_h^{w1} \cap \Gamma_h^{w2},\, \Gamma_h^{w1} \cap \Gamma_h^{w3},\, \Gamma_h^{w2} \cap \Gamma_h^{w3}\}, \\ W^n_{i-\frac{1}{2},j-\frac{1}{2},k} = \tilde{W}^n_{i-\frac{1}{2},j-\frac{1}{2},k} \qquad \text{on } \{\Gamma_h^{w1} \cap \Gamma_h^{w2} \cap \Gamma_h^{w3}\}. \end{cases} \tag{4.45}$$

Step 5. Correct the predicted value of $W^n_{i-\frac{1}{2},j-\frac{1}{2},k}$ at the interface mesh points $\{\Gamma_h^{w1} \cap \Gamma_h^{w2} \cap \Gamma_h^{w3}\}$ by:

$$\begin{cases} \dfrac{W^n_{i-\frac{1}{2},j-\frac{1}{2},k} - W^{n-1}_{i-\frac{1}{2},j-\frac{1}{2},k}}{\tau} = \delta_z^2 W^n_{i-\frac{1}{2},j-\frac{1}{2},k} + \delta_z\delta_x U^n_{i-\frac{1}{2},j-\frac{1}{2},k} \\ \qquad + \delta_z\delta_y V^n_{i-\frac{1}{2},j-\frac{1}{2},k} \qquad \text{on } \{\Gamma_h^{w1} \cap \Gamma_h^{w2} \cap \Gamma_h^{w3}\}. \end{cases} \tag{4.46}$$

4.5.4 Parallel Approximation of ξ

Let $\xi^n_{i+\frac{1}{2},j,k}$ be the numerical approximation of ξ at $(x_{i+\frac{1}{2}}, y_j, z_k)$ at time t^n, and let D^ξ_h be the set of mesh points associated with $\xi^n_{i-\frac{1}{2},j,k}$:

$$D^\xi_h = \{(x_{i-\frac{1}{2}}, y_j, z_k) \mid i = 0, \ldots, J_x + 1, \;\; j = 0, \ldots, J_y, \;\; k = 0, \ldots, J_z\}.$$

Decompose D^ξ_h into $(P^\xi + 1)(Q^\xi + 1)(S^\xi + 1)$ non-overlapping sub-domains:

$$D^\xi_{h\alpha,\beta,\gamma} = \{(x_{i-\frac{1}{2}}, y_j, z_k) \mid i^\xi_{\alpha-1} < i < i^\xi_\alpha, \quad j^\xi_{\beta-1} < j < j^\xi_\beta \quad k^\xi_{\gamma-1} < k < k^\xi_\gamma\}.$$

Let $\Gamma^\xi_h = \Gamma^{\xi 1}_h \bigcup \Gamma^{\xi 2}_h \bigcup \Gamma^{\xi 3}_h$ be the set of all mesh points on the interfaces where:

$$\begin{aligned}
\Gamma^{\xi 1}_h &= \{(x_{i^\xi_\alpha - \frac{1}{2}}, y_j, z_k) \mid 4 \leq i^\xi_\alpha + 2 \leq i^\xi_{\alpha+1}, \quad 1 \leq \alpha \leq P^\xi, \quad 0 < j < J_y, \\
&\qquad 0 < k < J_z\}, \\
\Gamma^{\xi 2}_h &= \{(x_{i-\frac{1}{2}}, y_{j^\xi_\beta}, z_k) \mid 4 \leq j^\xi_\beta + 2 \leq j^\xi_{\beta+1}, \quad 1 \leq \beta \leq Q^\xi, \quad 0 < i < (J_x + 1), \\
&\qquad 0 < k < J_z\}, \\
\Gamma^{\xi 3}_h &= \{(x_{i-\frac{1}{2}}, y_j, z_{k^\xi_\gamma}) \mid 4 \leq k^\xi_\gamma + 2 \leq k^\xi_{\gamma+1}, \quad 1 \leq \gamma \leq S^\xi, \quad 0 < i < (J_x + 1), \\
&\qquad 0 < j < J_y\}.
\end{aligned} \tag{4.47}$$

Here, $i^\xi_0 = 0$, $i^\xi_{P+1} = J_x + 1$, $j^\xi_0 = 0$, $j^\xi_{Q+1} = J_y$, $k^\xi_0 = 0$ and $k^\xi_{S+1} = J_z$. The vorticity equation (4.11) is solved in parallel by the following steps:

Step 1. Predict the value of $\xi^n_{i-\frac{1}{2},j,k}$ at the interface mesh points Γ^ξ_h by:

$$\tilde{\xi}^n_{i-\frac{1}{2},j,k} = 2\xi^{n-1}_{i-\frac{1}{2},j,k} - \xi^{n-2}_{i-\frac{1}{2},j,k}. \tag{4.48}$$

Step 2. Find $\xi^n_{i-\frac{1}{2},j,k}$ at the interior points of subdomains by:

$$\begin{cases} \dfrac{\xi^n_{i-\frac{1}{2},j,k} - \xi^{n-1}_{i-\frac{1}{2},j,k}}{\tau} = \dfrac{1}{Re}(\delta_x^2 \xi^n_{i-\frac{1}{2},j,k} + \delta_y^2 \xi^n_{i-\frac{1}{2},j,k} + \delta_z^2 \xi^n_{i-\frac{1}{2},j,k}) \\ \qquad + \delta_y (U\eta)^{n-1}_{i-\frac{1}{2},j,k} - \delta_y (V\xi)^{n-1}_{i-\frac{1}{2},j,k} - \delta_z (W\xi)^{n-1}_{i-\frac{1}{2},j,k} \\ \qquad + \delta_z (U\zeta)^{n-1}_{i-\frac{1}{2},j,k} + \delta_y {f_w}^n_{i-\frac{1}{2},j,k} - \delta_z {f_v}^n_{i-\frac{1}{2},j,k} \\ \qquad\qquad \text{on } \ D^\xi_h \backslash \{\Gamma^\xi_h \cup \partial D^\xi_h\}, \\ \xi^n_{i-\frac{1}{2},j-\frac{1}{2},k} = \tilde{\xi}^n_{i-\frac{1}{2},j-\frac{1}{2},k} \qquad \text{on } \ \Gamma^\xi_h \ . \end{cases} \tag{4.49}$$

Step 3. Correct the predicted value of $\xi^n_{i-\frac{1}{2},j,k}$ at the interface mesh points $\Gamma^\xi_h \backslash \{\Gamma^{\xi 1}_h \cap \Gamma^{\xi 2}_h,\ \Gamma^{\xi 1}_h \cap \Gamma^{\xi 3}_h,\ \Gamma^{\xi 2}_h \cap \Gamma^{\xi 3}_h\}$ by:

$$\begin{cases} \dfrac{\xi^n_{i-\frac{1}{2},j,k} - \xi^{n-1}_{i-\frac{1}{2},j,k}}{\tau} = \dfrac{1}{Re}(\delta_x^2 \xi^n_{i-\frac{1}{2},j,k} + \delta_y^2 \xi^n_{i-\frac{1}{2},j,k} + \delta_z^2 \xi^n_{i-\frac{1}{2},j,k}) \\ \qquad + \delta_y (U\eta)^{n-1}_{i-\frac{1}{2},j,k} - \delta_y (V\xi)^{n-1}_{i-\frac{1}{2},j,k} - \delta_z (W\xi)^{n-1}_{i-\frac{1}{2},j,k} \\ \qquad + \delta_z (U\zeta)^{n-1}_{i-\frac{1}{2},j,k} + \delta_y {f_w}^n_{i-\frac{1}{2},j,k} - \delta_z {f_v}^n_{i-\frac{1}{2},j,k} \\ \qquad\qquad \text{on } \ \Gamma^\xi_h \backslash \{\Gamma^{\xi 1}_h \cap \Gamma^{\xi 2}_h,\ \Gamma^{\xi 1}_h \cap \Gamma^{\xi 3}_h,\ \Gamma^{\xi 2}_h \cap \Gamma^{\xi 3}_h\}, \\ \xi^n_{i-\frac{1}{2},j-\frac{1}{2},k} = \tilde{\xi}^n_{i-\frac{1}{2},j-\frac{1}{2},k} \qquad \text{on } \ \{\Gamma^{\xi 1}_h \cap \Gamma^{\xi 2}_h,\ \Gamma^{\xi 1}_h \cap \Gamma^{\xi 3}_h,\ \Gamma^{\xi 2}_h \cap \Gamma^{\xi 3}_h\}. \end{cases} \tag{4.50}$$

Step 4. Correct the predicted value of $\xi^n_{i-\frac{1}{2},j,k}$ at the interface mesh points $\{\Gamma^{\xi 1}_h \cap$

$\Gamma_h^{\xi 2}$, $\Gamma_h^{\xi 1} \cap \Gamma_h^{\xi 3}$, $\Gamma_h^{\xi 2} \cap \Gamma_h^{\xi 3}\}$ by:

$$\left\{\begin{aligned}
\frac{\xi^{n}_{i-\frac{1}{2},j,k} - \xi^{n-1}_{i-\frac{1}{2},j,k}}{\tau} &= \frac{1}{Re}(\delta_x^2 \xi^{n}_{i-\frac{1}{2},j,k} + \delta_y^2 \xi^{n}_{i-\frac{1}{2},j,k} + \delta_z^2 \xi^{n}_{i-\frac{1}{2},j,k}) \\
&\quad + \delta_y (U\eta)^{n-1}_{i-\frac{1}{2},j,k} - \delta_y (V\xi)^{n-1}_{i-\frac{1}{2},j,k} - \delta_z (W\xi)^{n-1}_{i-\frac{1}{2},j,k} \\
&\quad + \delta_z (U\zeta)^{n-1}_{i-\frac{1}{2},j,k} + \delta_y {f_w}^{n}_{i-\frac{1}{2},j,k} - \delta_z {f_v}^{n}_{i-\frac{1}{2},j,k} \\
&\qquad\qquad \text{on } \{\Gamma_h^{\xi 1} \cap \Gamma_h^{\xi 2},\, \Gamma_h^{\xi 1} \cap \Gamma_h^{\xi 3},\, \Gamma_h^{xi2} \cap \Gamma_h^{\xi 3}\} \\
\xi^{n}_{i-\frac{1}{2},j-\frac{1}{2},k} = \tilde{\xi}^{n}_{i-\frac{1}{2},j-\frac{1}{2},k} &\qquad \text{on } \{\Gamma_h^{\xi 1} \cap \Gamma_h^{\xi 2} \cap \Gamma_h^{\xi 3}\}.
\end{aligned}\right. \tag{4.51}$$

Step 5. Correct the predicted value of $\xi^{n}_{i-\frac{1}{2},j,k}$ at the interface mesh points $\{\Gamma_h^{\xi 1} \cap \Gamma_h^{\xi 2} \cap \Gamma_h^{\xi 3}\}$ by:

$$\left\{\begin{aligned}
\frac{\xi^{n}_{i-\frac{1}{2},j,k} - \xi^{n-1}_{i-\frac{1}{2},j,k}}{\tau} &= \frac{1}{Re}(\delta_x^2 \xi^{n}_{i-\frac{1}{2},j,k} + \delta_y^2 \xi^{n}_{i-\frac{1}{2},j,k} + \delta_z^2 \xi^{n}_{i-\frac{1}{2},j,k}) \\
&\quad + \delta_y (U\eta)^{n-1}_{i-\frac{1}{2},j,k} - \delta_y (V\xi)^{n-1}_{i-\frac{1}{2},j,k} - \delta_z (W\xi)^{n-1}_{i-\frac{1}{2},j,k} \\
&\quad + \delta_z (U\zeta)^{n-1}_{i-\frac{1}{2},j,k} + \delta_y {f_w}^{n}_{i-\frac{1}{2},j,k} - \delta_z {f_v}^{n}_{i-\frac{1}{2},j,k} \\
&\qquad\qquad \text{on } \{\Gamma_h^{\xi 1} \cap \Gamma_h^{\xi 2} \cap \Gamma_h^{\xi 3}\}.
\end{aligned}\right. \tag{4.52}$$

The boundary conditions used in the parallel approximation of equations (4.10), (4.11) and (4.12) are given in the equations (4.13), (4.14) and (4.13), respectively. Moreover, the first level values are obtained by the definition, by taking curl of the velocity vectors obtained at time level 1.

4.5.5 Parallel Approximation of η

Let $\eta^n_{i,j+\frac{1}{2},k}$ be the numerical approximation of the exact solution at $(x_i, y_{j+\frac{1}{2}}, z_k)$ at time t^n and let D^η_h be the set of mesh points associated with $\eta^n_{i,j-\frac{1}{2},k}$:

$$D^\eta_h = \{(x_i, y_{j-\frac{1}{2}}, z_k) \mid i = 0, \ldots, J_x, \;\; j = 0, \ldots, J_y + 1, \;\; k = 0, \ldots, J_z\}$$

Decompose D^η_h into $(P^\eta + 1)(Q^\eta + 1)(S^\eta + 1)$ non-overlapping sub-domains:

$$D^\eta_{h\alpha,\beta,\gamma} = \{(x_i, y_{j-\frac{1}{2}}, z_k) \mid \; i^\eta_{\alpha-1} < i < i^\eta_\alpha, \quad j^\eta_{\beta-1} < j < j^\eta_\beta \quad k^\eta_{\gamma-1} < k < k^\eta_\gamma\}.$$

Let Γ^η_h be the set of all mesh points on the interfaces, that is, $\Gamma^\eta_h = \Gamma^{\eta 1}_h \bigcup \Gamma^{\eta 2}_h \bigcup \Gamma^{\eta 3}_h$, where:

$$\begin{aligned}
\Gamma^{\eta 1}_h &= \{(x_{i^\eta_\alpha}, y_{j-\frac{1}{2}}, z_k) \mid 4 \leq i^\eta_\alpha + 2 \leq i^\eta_{\alpha+1}, \;\; 1 \leq \alpha \leq P^\eta, \;\; 0 < j < (J_y + 1) \\
&\qquad 0 < k < J_z\}, \\
\Gamma^{\eta 2}_h &= \{(x_i, y_{j^\eta_\beta - \frac{1}{2}}, z_k) \mid 4 \leq j^\eta_\beta + 2 \leq j^\eta_{\beta+1}, \;\; 1 \leq \beta \leq Q^\eta, \;\; 0 < i < J_x, \\
&\qquad 0 < k < J_z\}, \\
\Gamma^{\eta 3}_h &= \{(x_i, y_{j-\frac{1}{2}}, z_{k^\eta_\gamma}) \mid 4 \leq k^\eta_\gamma + 2 \leq k^\eta_{\gamma+1}, \;\; 1 \leq \gamma \leq S^\eta, \;\; 0 < i < J_x, \\
&\qquad 0 < j < (J_y + 1)\},
\end{aligned} \tag{4.53}$$

Here, $i^\eta_0 = 0$, $i^\eta_{P+1} = J_x$, $j^\eta_0 = 0$, $j^\eta_{Q+1} = J_y + 1$, $k^\eta_0 = 0$ and $k^\eta_{S+1} = J_z$. Below is the summary of parallel scheme for solving equation (4.10):

Step 1. Predict the value $\eta^n_{i,j-\frac{1}{2},k}$ at the interface mesh points Γ^η_h by:

$$\tilde{\eta}^n_{i,j-\frac{1}{2},k} = 2\eta^{n-1}_{i,j-\frac{1}{2},k} - \eta^{n-2}_{i,j-\frac{1}{2},k}. \tag{4.54}$$

Step 2. Find $\eta^n_{i,j-\frac{1}{2},k}$ at the interior points of subdomains by:

$$\begin{cases} \dfrac{\eta^n_{i,j-\frac{1}{2},k} - \eta^{n-1}_{i,j-\frac{1}{2},k}}{\tau} = \dfrac{1}{Re}(\delta_x^2\eta^n_{i,j-\frac{1}{2},k} + \delta_y^2\eta^n_{i,j-\frac{1}{2},k} + \delta_z^2\eta^n_{i,j-\frac{1}{2},k}) \\ \qquad - \delta_x(U\eta)^{n-1}_{i,j-\frac{1}{2},k} - \delta_z(W\eta)^{n-1}_{i,j-\frac{1}{2},k} + \delta_x(V\xi)^{n-1}_{i,j-\frac{1}{2},k} \\ \qquad + \delta_z(V\zeta)^{n-1}_{i,j-\frac{1}{2},k} - \delta_x {f_w}^n_{i,j-\frac{1}{2},k} + \delta_z {f_u}^n_{i,j-\frac{1}{2},k} \\ \qquad\qquad \text{on } D^\eta_h \backslash \{\Gamma^\eta_h \cup \partial D^\eta_h\}, \\ \eta^n_{i,j-\frac{1}{2},k} = \tilde{\eta}^n_{i,j-\frac{1}{2},k} \qquad \text{on } \Gamma^\eta_h . \end{cases} \tag{4.55}$$

Step 3. Correct the predicted value of $\eta^n_{i,j-\frac{1}{2},k}$ at the interface mesh points $\Gamma^\eta_h \backslash \{\Gamma^{\eta1}_h \cap \Gamma^{\eta2}_h,\, \Gamma^{\eta1}_h \cap \Gamma^{\eta3}_h,\, \Gamma^{\eta2}_h \cap \Gamma^{\eta3}_h\}$ by:

$$\begin{cases} \dfrac{\eta^n_{i,j-\frac{1}{2},k} - \eta^{n-1}_{i,j-\frac{1}{2},k}}{\tau} = \dfrac{1}{Re}(\delta_x^2\eta^n_{i,j-\frac{1}{2},k} + \delta_y^2\eta^n_{i,j-\frac{1}{2},k} + \delta_z^2\eta^n_{i,j-\frac{1}{2},k}) \\ \qquad - \delta_x(U\eta)^{n-1}_{i,j-\frac{1}{2},k} - \delta_z(W\eta)^{n-1}_{i,j-\frac{1}{2},k} + \delta_x(V\xi)^{n-1}_{i,j-\frac{1}{2},k} \\ \qquad + \delta_z(V\zeta)^{n-1}_{i,j-\frac{1}{2},k} - \delta_x {f_w}^n_{i,j-\frac{1}{2},k} + \delta_z {f_u}^n_{i,j-\frac{1}{2},k} \\ \qquad\qquad \text{on } \Gamma^\eta_h \backslash \{\Gamma^{\eta1}_h \cap \Gamma^{\eta2}_h,\, \Gamma^{\eta1}_h \cap \Gamma^{\eta3}_h,\, \Gamma^{\eta2}_h \cap \Gamma^{\eta3}_h\}, \\ \eta^n_{i,j-\frac{1}{2},k} = \tilde{\eta}^n_{i,j-\frac{1}{2},k} \qquad \text{on } \{\Gamma^{\eta1}_h \cap \Gamma^{\eta2}_h,\, \Gamma^{\eta1}_h \cap \Gamma^{\eta3}_h,\, \Gamma^{\eta2}_h \cap \Gamma^{\eta3}_h\}. \end{cases} \tag{4.56}$$

Step 4. Correct the predicted value of $\eta^n_{i,j-\frac{1}{2},k}$ at the interface mesh points $\{\Gamma^{\eta 1}_h \cap \Gamma^{\eta 2}_h,\, \Gamma^{\eta 1}_h \cap \Gamma^{\eta 3}_h,\, \Gamma^{\eta 2}_h \cap \Gamma^{\eta 3}_h\}$ by:

$$\begin{cases} \dfrac{\eta^n_{i,j-\frac{1}{2},k} - \eta^{n-1}_{i,j-\frac{1}{2},k}}{\tau} = \dfrac{1}{Re}(\delta_x^2 \eta^n_{i,j-\frac{1}{2},k} + \delta_y^2 \eta^n_{i,j-\frac{1}{2},k} + \delta_z^2 \eta^n_{i,j-\frac{1}{2},k}) \\ \qquad - \delta_x(U\eta)^{n-1}_{i,j-\frac{1}{2},k} - \delta_z(W\eta)^{n-1}_{i,j-\frac{1}{2},k} + \delta_x(V\xi)^{n-1}_{i,j-\frac{1}{2},k} \\ \qquad + \delta_z(V\zeta)^{n-1}_{i,j-\frac{1}{2},k} - \delta_x {f_w}^n_{i,j-\frac{1}{2},k} + \delta_z {f_u}^n_{i,j-\frac{1}{2},k} \\ \qquad\qquad \text{on } \{\Gamma^{\eta 1}_h \cap \Gamma^{\eta 2}_h,\, \Gamma^{\eta 1}_h \cap \Gamma^{\eta 3}_h,\, \Gamma^{\eta 2}_h \cap \Gamma^{\eta 3}_h\}, \\ \eta^n_{i,j-\frac{1}{2},k} = \tilde{\eta}^n_{i,j-\frac{1}{2},k} \qquad \text{on } \{\Gamma^{\eta 1}_h \cap \Gamma^{\eta 2}_h \cap \Gamma^{\eta 3}_h\}. \end{cases} \tag{4.57}$$

Step 5. Correct the predicted value of $\eta^n_{i,j-\frac{1}{2},k}$ at the interface mesh points $\{\Gamma^{\eta 1}_h \cap \Gamma^{\eta 2}_h \cap \Gamma^{\eta 3}_h\}$ by:

$$\begin{cases} \dfrac{\eta^n_{i,j-\frac{1}{2},k} - \eta^{n-1}_{i,j-\frac{1}{2},k}}{\tau} = \dfrac{1}{Re}(\delta_x^2 \eta^n_{i,j-\frac{1}{2},k} + \delta_y^2 \eta^n_{i,j-\frac{1}{2},k} + \delta_z^2 \eta^n_{i,j-\frac{1}{2},k}) \\ \qquad - \delta_x(U\eta)^{n-1}_{i,j-\frac{1}{2},k} - \delta_z(W\eta)^{n-1}_{i,j-\frac{1}{2},k} + \delta_x(V\xi)^{n-1}_{i,j-\frac{1}{2},k} \\ \qquad + \delta_z(V\zeta)^{n-1}_{i,j-\frac{1}{2},k} - \delta_x {f_w}^n_{i,j-\frac{1}{2},k} + \delta_z {f_u}^n_{i,j-\frac{1}{2},k} \\ \qquad\qquad \text{on } \{\Gamma^{\eta 1}_h \cap \Gamma^{\eta 2}_h \cap \Gamma^{\eta 3}_h\}. \end{cases} \tag{4.58}$$

4.5.6 Parallel Approximation of ζ

Let $\zeta^n_{i,j,k+\frac{1}{2}}$ be the numerical approximation of ζ at $(x_i, y_j, z_{k+\frac{1}{2}})$ at time t^n, and let D^ζ_h be the set of mesh points associated with $\zeta^n_{i,j,k-\frac{1}{2}}$:

$$D^\zeta_h = \{(x_i, y_j, z_{k-\frac{1}{2}}) \mid i = 0, \ldots, J_x,\;\; j = 0, \ldots, J_y,\;\; k = 0, \ldots, J_z + 1\}.$$

Decompose D_h^ζ into $(P^\zeta+1)(Q^\zeta+1)(S^\zeta+1)$ non-overlapping subdomains:

$$D^\zeta_{h\alpha,\beta,\gamma} = \{(x_i, y_j, z_{k-\frac{1}{2}}) \mid i^\zeta_{\alpha-1} < i < i^\zeta_\alpha, \quad j^\zeta_{\beta-1} < j < j^\zeta_\beta \quad k^\zeta_{\gamma-1} < k < k^\zeta_\gamma\}.$$

Let Γ_h^ζ be the set of all mesh points on the interfaces, that is, $\Gamma_h^\zeta = \Gamma_h^{\zeta 1} \bigcup \Gamma_h^{\zeta 2} \bigcup \Gamma_h^{\zeta 3}$, where:

$$\begin{aligned}
\Gamma_h^{\zeta 1} &= \{(x_{i^\zeta_\alpha}, y_j, z_{k-\frac{1}{2}}) \mid 4 \le i^\zeta_\alpha + 2 \le i^\zeta_{\alpha+1}, \quad 1 \le \alpha \le P^\zeta, \quad 0 < j < J_y \\
&\qquad 0 < k < (J_z+1)\}, \\
\Gamma_h^{\zeta 2} &= \{(x_i, y_{j^\zeta_\beta}, z_{k-\frac{1}{2}}) \mid 4 \le j^\zeta_\beta + 2 \le j^\zeta_{\beta+1}, \quad 1 \le \beta \le Q^\zeta, \quad 0 < i < J_x, \\
&\qquad 0 < k < (J_z+1)\}, \\
\Gamma_h^{\zeta 3} &= \{(x_i, y_j, z_{k^\zeta_\gamma-\frac{1}{2}}) \mid 4 \le k^\zeta_\gamma + 2 \le k^\zeta_{\gamma+1}, \quad 1 \le \gamma \le S^\zeta, \quad 0 < i < J_x, \\
&\qquad 0 < j < J_y\},
\end{aligned} \tag{4.59}$$

Here, $i^\zeta_0 = 0$, $i^\zeta_{P+1} = J_x$, $j^\zeta_0 = 0$, $j^\zeta_{Q+1} = J_y$, $k^\zeta_0 = 0$ and $k^\zeta_{S+1} = J_z + 1$. We present the developed parallel scheme for solving the equation (4.12) in the following:

Step 1. Predict the value of $\zeta^n_{i,j,k-\frac{1}{2}}$ at the interface mesh points Γ^ζ_h by:

$$\tilde{\zeta}^n_{i,j,k-\frac{1}{2}} = 2\zeta^{n-1}_{i,j,k-\frac{1}{2}} - \zeta^{n-2}_{i,j,k-\frac{1}{2}}. \tag{4.60}$$

Step 2. Find $\zeta^n_{i,j,k-\frac{1}{2}}$ at the interior points of subdomains by:

$$\left\{\begin{aligned} \frac{\zeta^n_{i,j,k-\frac{1}{2}} - \zeta^{n-1}_{i,j,k-\frac{1}{2}}}{\tau} &= \frac{1}{Re}(\delta_x^2\zeta^n_{i,j,k-\frac{1}{2}} + \delta_y^2\zeta^n_{i,j,k-\frac{1}{2}} + \delta_z^2\zeta^n_{i,j,k-\frac{1}{2}}) \\ &\quad - \delta_x(U\zeta)^{n-1}_{i,j,k-\frac{1}{2}} - \delta_y(V\zeta)^{n-1}_{i,j,k-\frac{1}{2}} + \delta_x(W\xi)^{n-1}_{i,j,k-\frac{1}{2}} \\ &\quad + \delta_y(W\eta)^{n-1}_{i,j,k-\frac{1}{2}} + \delta_x {f_v}^n_{i,j,k-\frac{1}{2}} - \delta_y {f_u}^n_{i,j,k-\frac{1}{2}} \\ &\qquad\qquad \text{on } D^\zeta_h \backslash \{\Gamma^\zeta_h \cup \partial D^\zeta_h\}, \\ \zeta^n_{i,j,k-\frac{1}{2}} &= \tilde{\zeta}^n_{i,j,k-\frac{1}{2}} \qquad \text{on } \Gamma^\zeta_h. \end{aligned}\right. \tag{4.61}$$

Step 3. Correct the predicted value of $\zeta^n_{i,j,k-\frac{1}{2}}$ at the interface mesh points $\Gamma^\zeta_h \backslash \{\Gamma^{\zeta 1}_h \cap \Gamma^{\zeta 2}_h, \Gamma^{\zeta 1}_h \cap \Gamma^{\zeta 3}_h, \Gamma^{\zeta 2}_h \cap \Gamma^{\zeta 3}_h\}$ by:

$$\left\{\begin{aligned} \frac{\zeta^n_{i,j,k-\frac{1}{2}} - \zeta^{n-1}_{i,j,k-\frac{1}{2}}}{\tau} &= \frac{1}{Re}(\delta_x^2\zeta^n_{i,j,k-\frac{1}{2}} + \delta_y^2\zeta^n_{i,j,k-\frac{1}{2}} + \delta_z^2\zeta^n_{i,j,k-\frac{1}{2}}) \\ &\quad - \delta_x(U\zeta)^{n-1}_{i,j,k-\frac{1}{2}} - \delta_y(V\zeta)^{n-1}_{i,j,k-\frac{1}{2}} + \delta_x(W\xi)^{n-1}_{i,j,k-\frac{1}{2}} \\ &\quad + \delta_y(W\eta)^{n-1}_{i,j,k-\frac{1}{2}} + \delta_x {f_v}^n_{i,j,k-\frac{1}{2}} - \delta_y {f_u}^n_{i,j,k-\frac{1}{2}} \\ &\qquad\qquad \text{on } \Gamma^\zeta_h \backslash \{\Gamma^{\zeta 1}_h \cap \Gamma^{\zeta 2}_h, \Gamma^{\zeta 1}_h \cap \Gamma^{\zeta 3}_h, \Gamma^{\zeta 2}_h \cap \Gamma^{\zeta 3}_h\}, \\ \zeta^n_{i,j,k-\frac{1}{2}} &= \tilde{\zeta}^n_{i,j,k-\frac{1}{2}} \qquad \text{on } \{\Gamma^{\zeta 1}_h \cap \Gamma^{\zeta 2}_h, \Gamma^{\zeta 1}_h \cap \Gamma^{\zeta 3}_h, \Gamma^{\zeta 2}_h \cap \Gamma^{\zeta 3}_h\}. \end{aligned}\right. \tag{4.62}$$

Step 4. Correct the predicted value of $\zeta^n_{i,j,k-\frac{1}{2}}$ at the interface mesh points $\{\Gamma_h^{\zeta 1} \cap \Gamma_h^{\zeta 2}, \Gamma_h^{\zeta 1} \cap \Gamma_h^{\zeta 3}, \Gamma_h^{\zeta 2} \cap \Gamma_h^{\zeta 3}\}$ by:

$$\begin{cases} \dfrac{\zeta^n_{i,j,k-\frac{1}{2}} - \zeta^{n-1}_{i,j,k-\frac{1}{2}}}{\tau} = \dfrac{1}{Re}(\delta_x^2 \zeta^n_{i,j,k-\frac{1}{2}} + \delta_y^2 \zeta^n_{i,j,k-\frac{1}{2}} + \delta_z^2 \zeta^n_{i,j,k-\frac{1}{2}}) \\ \qquad - \delta_x (U\zeta)^{n-1}_{i,j,k-\frac{1}{2}} - \delta_y (V\zeta)^{n-1}_{i,j,k-\frac{1}{2}} + \delta_x (W\xi)^{n-1}_{i,j,k-\frac{1}{2}} \\ \qquad + \delta_y (W\eta)^{n-1}_{i,j,k-\frac{1}{2}} + \delta_x f_{v\,i,j,k-\frac{1}{2}}^{\,n} - \delta_y f_{u\,i,j,k-\frac{1}{2}}^{\,n} \\ \qquad\qquad \text{on } \{\Gamma_h^{\zeta 1} \cap \Gamma_h^{\zeta 2}, \Gamma_h^{\zeta 1} \cap \Gamma_h^{\zeta 3}, \Gamma_h^{\zeta 2} \cap \Gamma_h^{\zeta 3}\}, \\ \zeta^n_{i,j,k-\frac{1}{2}} = \tilde{\zeta}^n_{i,j,k-\frac{1}{2}} \qquad \text{on } \{\Gamma_h^{\zeta 1} \cap \Gamma_h^{\zeta 2} \cap \Gamma_h^{\zeta 3}\}. \end{cases} \tag{4.63}$$

Step 5. Correct the predicted value of $\zeta^n_{i,j,k-\frac{1}{2}}$ at the interface mesh points $\{\Gamma_h^{\zeta 1} \cap \Gamma_h^{\zeta 2} \cap \Gamma_h^{\zeta 3}\}$ by

$$\begin{cases} \dfrac{\zeta^n_{i,j,k-\frac{1}{2}} - \zeta^{n-1}_{i,j,k-\frac{1}{2}}}{\tau} = \dfrac{1}{Re}(\delta_x^2 \zeta^n_{i,j,k-\frac{1}{2}} + \delta_y^2 \zeta^n_{i,j,k-\frac{1}{2}} + \delta_z^2 \zeta^n_{i,j,k-\frac{1}{2}}) \\ \qquad - \delta_x (U\zeta)^{n-1}_{i,j,k-\frac{1}{2}} - \delta_y (V\zeta)^{n-1}_{i,j,k-\frac{1}{2}} + \delta_x (W\xi)^{n-1}_{i,j,k-\frac{1}{2}} \\ \qquad + \delta_y (W\eta)^{n-1}_{i,j,k-\frac{1}{2}} + \delta_x f_{v\,i,j,k-\frac{1}{2}}^{\,n} - \delta_y f_{u\,i,j,k-\frac{1}{2}}^{\,n} \\ \qquad\qquad \text{on } \{\Gamma_h^{\zeta 1} \cap \Gamma_h^{\zeta 2} \cap \Gamma_h^{\zeta 3}\}. \end{cases} \tag{4.64}$$

4.6 Determination of Body Forces

To determine the body forces acting on the rotor blades, a blade-element approach combined with two-dimensional airfoil characteristics is used. The full description is given in Chapter 3 in Section 3.6.

4.7 Joint Optimization Using Stochastic Programming

To solve the scenario-based approximation model (4.9) which is an approximation to the stochastic optimization model (4.1), we use the pattern search algorithm [67]. This method starts with an arbitrary initial point, i.e., a set of operating configuration, $\beta_i, \Omega_i, i = 1, \cdots, N$, where β_i and Ω_i satisfy the constraints in (4.1). Then, it chooses a certain set of search directions at each iterate, and evaluates the expected total power which is the objective function at a given step length along each of these directions. If an operating point with a significantly higher expected total power is found, it is adopted as the new operating point and the step length is increased, otherwise the step length is decreased; this process is repeated until convergence. For the model (4.9), let x be the vector of all decision variables, and D_{feasible} be the feasible region:

$$x \equiv [\beta_1, \cdots, \beta_N, \Omega_1, \cdots, \Omega_N]^T, \tag{4.65}$$

$$D_{\text{feasible}} = \{x \mid \beta_{1\min} \leq x_1 \leq \beta_{1\max}, \ldots, \Omega_{N\min} \leq x_{2N} \leq \Omega_{N\max}\},$$

and f be the objective function:

$$f \equiv \sum_{i=1}^{N} \frac{\sum_{j=1}^{M}[P_i(U_0^j)]}{M}, \tag{4.66}$$

then the stochastic joint optimization algorithm based on the pattern search method in its matrix notation is presented in Algorithm 7. All the parameters in this algorithm such as the convergence tolerance, the contraction parameter, the aggressive parameter, the sufficient increase function and the direction set are the same as the ones used in Algorithm 5.

4.8 Numerical Simulation

In this section, we present numerical results. First, we perform two numerical experiments to validate the developed corrected-explicit-implicit domain decomposition scheme for the parallel approximation of the three-dimensional Navier-Stokes equations in a velocity-vorticity formulation (4.30)-(4.64). Then, we estimate the power generation of a commercial wind turbine called WindSpot, and investigate its wake characteristic using the developed parallel scheme and the actuator line method. Finally, we present numerical results to test the efficiency and accuracy of the proposed numerical Algorithm 7 which is used to optimize the total power while considering the wake effect as well as the randomness of the incoming wind speed. Withing this algorithm, we use the developed parallel algorithm and the actuator line method to evaluate the objective function at a given operating points.

4.8.1 Validation of Parallel Simulation of Navier-Stokes Equations

We carry out two numerical experiments to validate the developed parallel scheme (4.30)-(4.64). In the first numerical experiment, we choose a problem that has an exact analytical solution and use the exact solution as a benchmark. In the sec-

Algorithm 7 Stochastic Joint Optimization

```
initialize the parameters $\gamma_{\text{tol}}$, $\theta$, $\eta$, $\mu(\cdot)$, guess $x_0$ and set $\gamma_0 \geq \gamma_{tol}$.
Generate M scenarios, $U_0^j, j = 1, 2, 3, \ldots, M$ by Algorithm 6.
evaluate $f(x_0)$ by Algorithm (4.5) and (3.87).
for $k = 1, 2, 3, \ldots$ do
    if $\gamma_k \leq \gamma_{\text{tol}}$ then
        return.
    end if
    for $p_k \in d$ do
        if $(x_k + \gamma_k p_k) \notin D_{\text{feasible}}$ then
            $x_k + \gamma_k p_k = x_b$.
        end if
        evaluate $f(x_k + \gamma_k p_k)$ by Algorithm 4.5, Algorithm 6 and (3.87);
        if $f(x_k + \gamma_k p_k) > f(x_k) + \mu(\gamma_k)$ then
            $x_{k+1} \leftarrow x_k + \gamma_k p_k$, $\gamma_{k+1} \leftarrow \gamma_k \eta$ and break.
        else
            $x_{k+1} \leftarrow x_k$ and $\gamma_{k+1} \leftarrow \gamma_k \theta$.
        end if
    end for
end for
```

ond numerical experiment, we choose a lid-driven cavity problem as a benchmark problem. This problem is always used in the literature [35,41,44,87,94] to test the accuracy and efficiency of a new scheme developed for the numerical approximation of Navier-Stokes equations.

Example 1: Consider the following three-dimensional incompressible Navier-Stokes equations:

$$\left\{\begin{aligned} \frac{\partial U}{\partial t} + UU_x + VU_y + WU_z &= -\frac{1}{\rho}p_x + \nu(U_{xx} + U_{yy} + U_{zz}) + f_u, \\ \frac{\partial U}{\partial t} + UV_x + VV_y + WV_z &= -\frac{1}{\rho}p_y + \nu(V_{xx} + V_{yy} + V_{zz}) + f_v, \\ \frac{\partial W}{\partial t} + UW_x + VW_y + WW_z &= -\frac{1}{\rho}p_z + \nu(W_{xx} + W_{yy} + W_{zz}) + f_w, \\ U_x + V_y + W_z &= 0, \end{aligned}\right. \tag{4.67}$$

where the source terms are zero. It has an exact solution given by:

$$\left\{\begin{aligned} U(t,x,y,z) &= -a[\exp(ax)\sin(ay+dz) + \exp(az)\sin(ax+dy)]\exp(-d^2\nu t), \\ V(t,x,y,z) &= -a[\exp(ay)\sin(az+dx) + \exp(ax)\sin(ay+dz)]\exp(-d^2\nu t), \\ W(t,x,y,z) &= -a[\exp(az)\sin(ax+dy) + \exp(ay)\sin(az+dx)]\exp(-d^2\nu t), \end{aligned}\right. \tag{4.68}$$

where $a = 1$, $d = 1$ and $\nu = 1$. Now, we apply the developed parallel scheme (4.30)-(4.64) to approximate the solutions of three-dimensional incompressible Navier-Stokes equations (4.67) in the computational domain $D = [0,1] \times [0,1] \times [0,1]$. In our simulation, the boundary conditions and the initial solutions are obtained using the exact solutions (4.68). Moreover, in our simulation, we use non-uniform grid points in x, y and z-direction where 17 gird points are used in each direction with the smallest spatial step size 0.05 and the maximum spatial step size 0.075. The numerical results as well as the exact solutions at the final time T=0.1 are given in Figures 4.4 and 4.5. In Figure 4.4 the approximated solutions obtained with no domain decomposition, whereas in Figure 4.5 the approximated solutions obtained with domain decomposition (the computational domain is decomposed into 2 × 2 × 2 multi-blocks). From these figures, it is evident that the proposed parallel scheme approximates the solutions of three-dimensional Navier-Stokes equations (4.67) with a good accuracy.

Example 2 (Lid-driven cavity problem): Consider the three-dimensional incompressible Navier-Stokes equations given in (4.67) with the following boundary conditions:

$$\begin{cases} U(t,0,\cdot,\cdot)=0, \;\; V(t,0,\cdot,\cdot)=0, \;\; W(t,0,\cdot,\cdot)=0, \\ U(t,1,\cdot,\cdot)=0, \;\; V(t,1,\cdot,\cdot)=0, \;\; W(t,1,\cdot,\cdot)=0, \\ U(t,\cdot,0,\cdot)=0, \;\; V(t,\cdot,0,\cdot)=0, \;\; W(t,\cdot,0,\cdot)=0, \\ U(t,\cdot,1,\cdot)=1, \;\; V(t,\cdot,1,\cdot)=0, \;\; W(t,\cdot,1,\cdot)=0, \\ U(t,\cdot,\cdot,0)=0, \;\; V(t,\cdot,\cdot,0)=0, \;\; W(t,\cdot,\cdot,0)=0, \\ U(t,\cdot,\cdot,1)=0, \;\; V(t,\cdot,\cdot,1)=0, \;\; W(t,\cdot,\cdot,1)=0. \end{cases} \tag{4.69}$$

We apply the developed parallel scheme (4.30)-(4.64) to simulate the solutions of

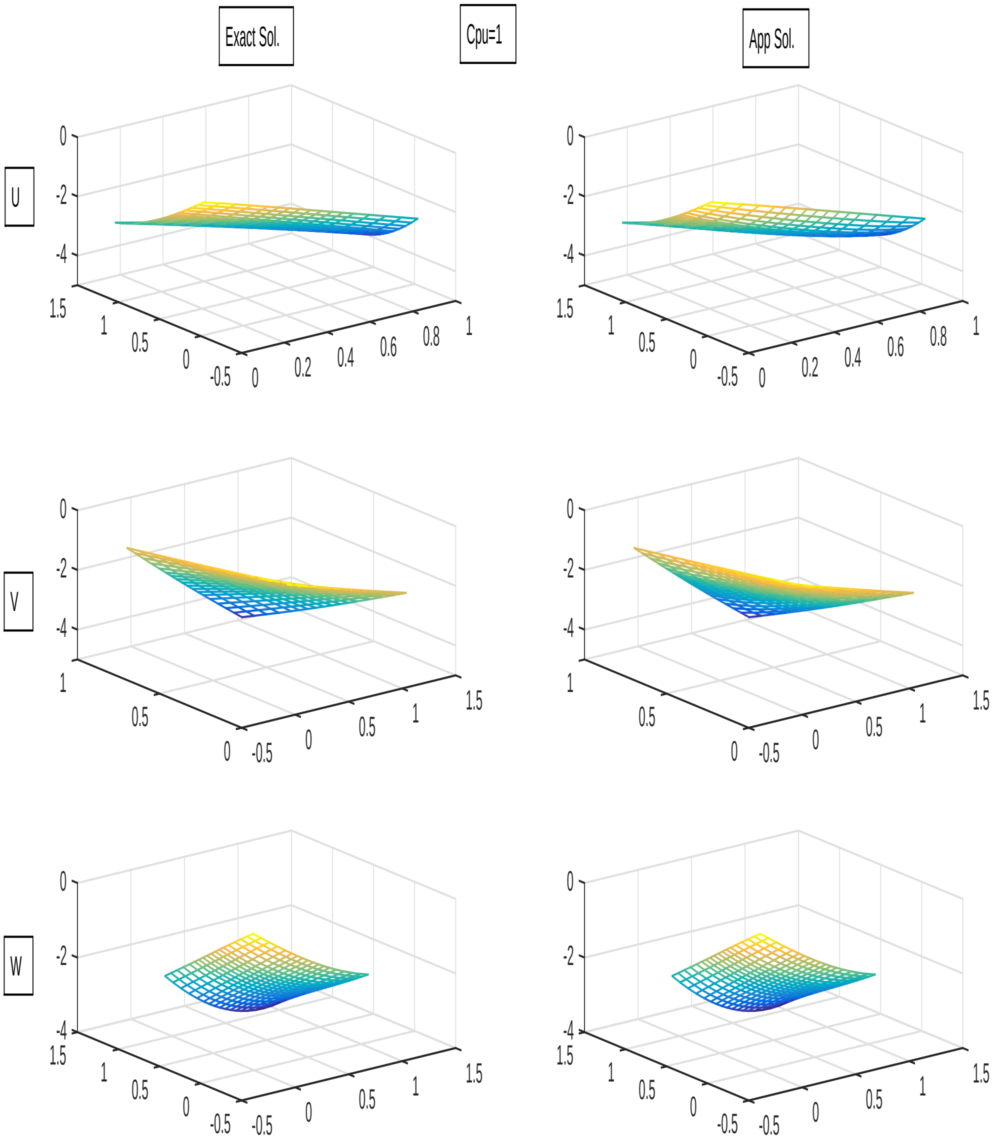


Figure 4.4: Comparison of estimated solutions and exact solutions.

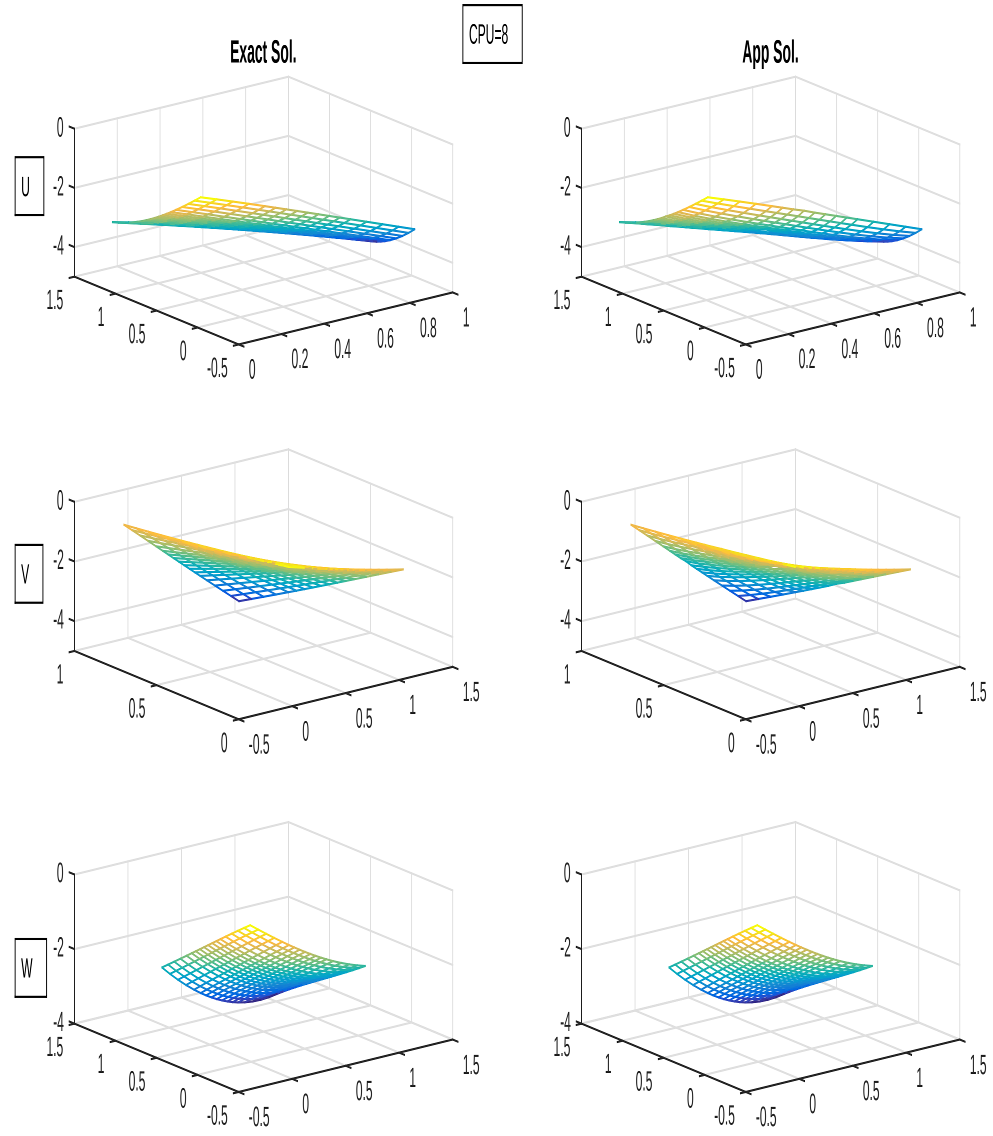


Figure 4.5: Comparison of estimated solutions and exact solutions.

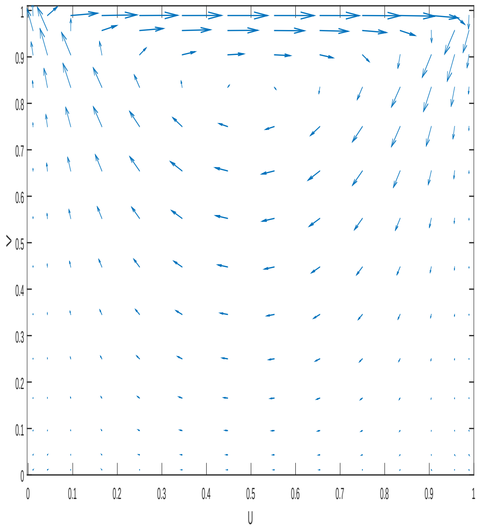


Figure 4.6: Velocity profile for the lid driven cavity flow at z=0.5 plane.

these Navier-Stokes equations (4.67) with the boundary conditions given in (4.69). We set the initial values for all the variables at the interior points to be zero, and run the simulation until the steady-state solutions are reached. In our simulation, the viscosity coefficient, the time step and the computational domain are taken as $\nu = 1$, $dt = 1 \times 10^{-5}$ and $D = [0,1] \times [0,1] \times [0,1]$, respectively. Moreover, in our simulation, we decompose the domain into $2 \times 2 \times 1$ subdomains and use non-uniform grid points in x, y and z-direction where 15 gird points are used in each direction. The grid point distribution in three spatial directions is taken the same and chosen as:

$$x_i = \frac{\cos(\frac{\pi}{2N}) - \cos((2i-1)\frac{\pi}{2N})}{\cos(\frac{\pi}{2N}) - \cos((2N-1)\frac{\pi}{2N})}, \quad i = 1, 2, \ldots, N.$$

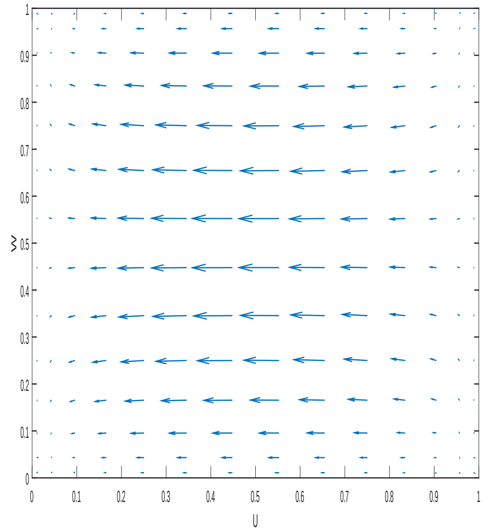


Figure 4.7: Velocity profile for the lid driven cavity flow at y=0.5 plane.

The numerical results are displayed in Figures 4.6, 4.7 and 4.8. These figures show the velocity profile at three typical planes. These flow patterns agree well with the results obtained in the references [41, 44].

4.8.2 Model Validation by Experimental Data

WindSpot is a 3.5 kW three-bladed wind turbine with a rotor diameter of 4.05 meters. It can be fitted with different set of blades, essentially at zero twist angle with a active pitch control system. The chord length of this turbine is 0.254 meters at the hub, and it decreases linearly to 0.156 meters at the blade tip. Moreover, it is operating at a fixed rotational speed of 12 $\frac{rad}{s}$ and a fixed pitch angle of 10.5

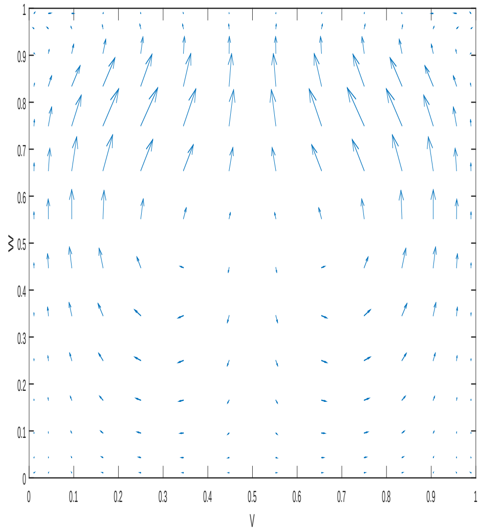


Figure 4.8: Velocity profile for the lid driven cavity flow at x=0.5 plane.

degrees [52].

In our simulation, the computational domain is taken as $30m \times 14m \times 14m$, and the grid points, see Figure 4.10, are concentrated near the blade tips and stretched in the x, y and z-direction. The resulting grid consists of 46 grid points in the axial direction, 50 points in the y-direction and 57 points in the z-direction. In the axial direction the grid spacing ranges from $dx = 0.02$ at the rotor plane to about $dx = 1.9476$ in the far wake and in the y-direction the spacing takes values from $dy = 0.02$ near the tip to about $dy = 1.2150$ at the lateral boundary. Moreover, in the z-direction the spacing takes values from $dz = 0.0346$ near the tip to about $dz = 1.4863$ at the lateral boundary. To ensure that the flow is fully developed in most of the wake, we carry out the computations until t=10 with a time step of $dt = 1 \times 10^{-3}$ which corresponds to 10000 time steps.

4.8.2.1 Power generation of WindSpot

We apply the developed corrected-explicit-implicit domain decomposition scheme (4.30)-(4.64) combined with the actuator line technique to estimate the power generation of WindSpot for different incoming wind speed. The numerical results are displayed in Figure 4.9. This figure shows the computed and the experimental power coefficient of WindSpot for different incoming wind speed. From this figure, it is evident that the computed and measured values are in excellent agreement for wind speeds up to about 10 $\frac{m}{s}$.

4.8.2.2 Wake Structures

In this subsection, we test the efficiency and accuracy of the developed parallel algorithm (4.30)-(4.64) in simulating a wind turbine wake. We apply the developed parallel scheme (4.30)-(4.64) combined with the actuator line technique to simulate the wake of WindSpot which is operating at a fixed rotational speed of 12 $\frac{rad}{s}$ and a fixed pitch angle of 10.5 degrees. The numerical results are displayed in Figures 4.11, 4.12,4.13 and 4.14. These figures show contours of magnitude of vorticity at y-z planes in ultimate wakes when the incoming wind speed is 10 $\frac{m}{s}$. From these

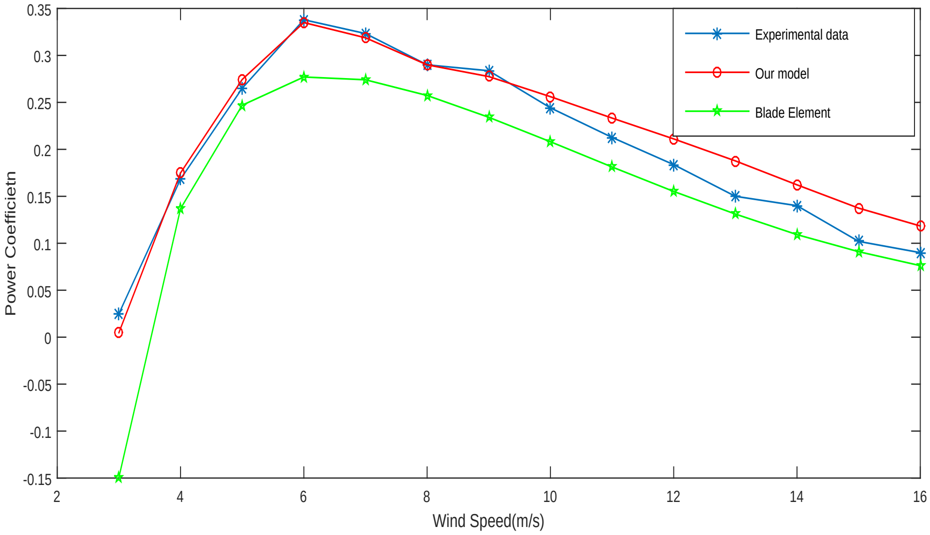


Figure 4.9: Comparison of measured and computed power.

figures, it is evident the developed parallel scheme combined with the actuator line method simulates the development of the wake vortices with high accuracy. Figure 4.14 shows the diffusion of the vortex about 3 to 4 rotor diameters behind the wind turbine. However, from experiments, it is known that the diffusion of the vortex happens at distance far behind the wind turbine [91]. This early diffusion of the vortex in our simulation is due to small Reynolds number and coarse grid used at far wake. The contours of magnitude of vorticity for different wind speeds are also investigated. Figure 4.15 depicts the contours of magnitude of vorticity at the plane of rotor for different incoming wind speeds. These results confirm that the developed parallel scheme combined with the actuator line method simulates the development of the wake vortices for different incoming wind speed with high accuracy.

Figure 4.16 shows the distribution of the axial interference factor (3.9), in the rotor plane when the incoming wind speed is 10 $\frac{m}{s}$. The three blades are seen as lines with a high density of contour lines. This is due to the large changes in induced velocity that takes place across the blades. The values of axial factor on the blades range from -0.0158 to 0.3933, with peak values appearing near the mid-section of the blades with a positive value on one side of the blade and a negative value on the other side of the blade. The tip vortices appear as localized regions where the value of axial interference factor is negative. In this region the minimum value of axial interference factor is about -.0119. This value of axial interference factor, -.0119, corresponded to an axial velocity that is 1.19% higher than the incoming wind speed.

The distribution of axial factor in z-direction at a constant radius is also investigated. Figure 4.17 shows the distribution of axial factor from one side of the blade to the other side in z-direction. From this figure, it is evident that the distribution is dominated by minimum and maximum values. The development of the axial factor distribution in the wake is depicted in Figure 4.18. These distributions are plotted along the blade from the hub to the tip at different distances behind the wind turbine in the wake region. It appears that the distribution dies out at

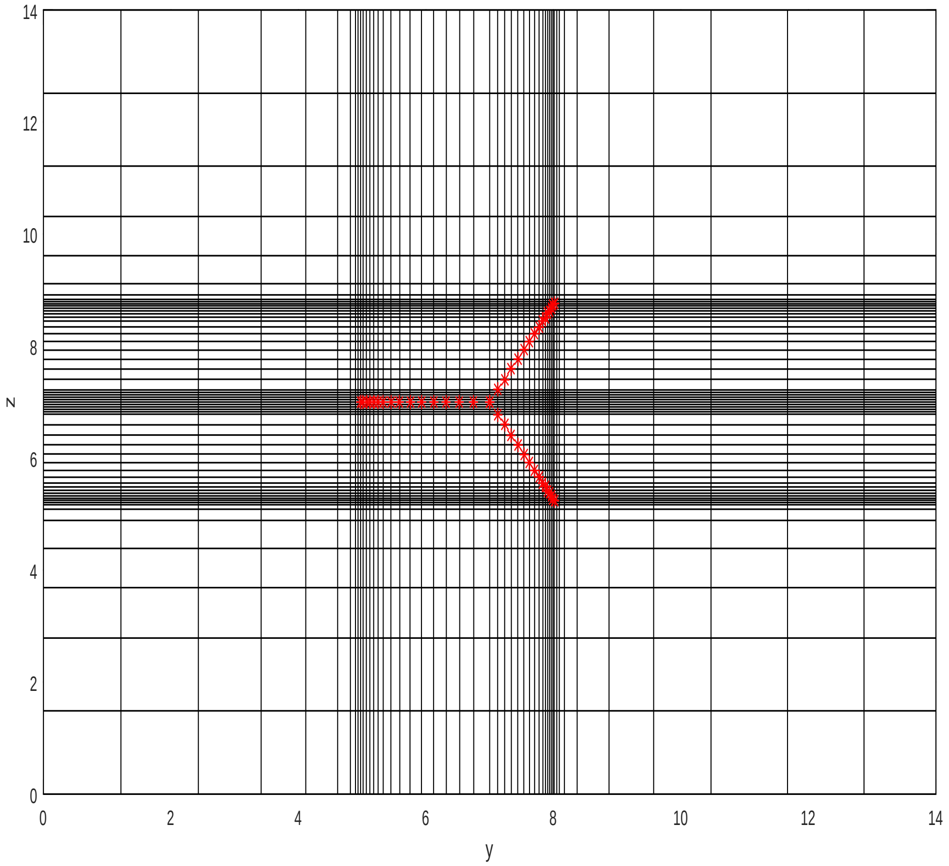


Figure 4.10: Mesh grid in the plane of the wind turbine.

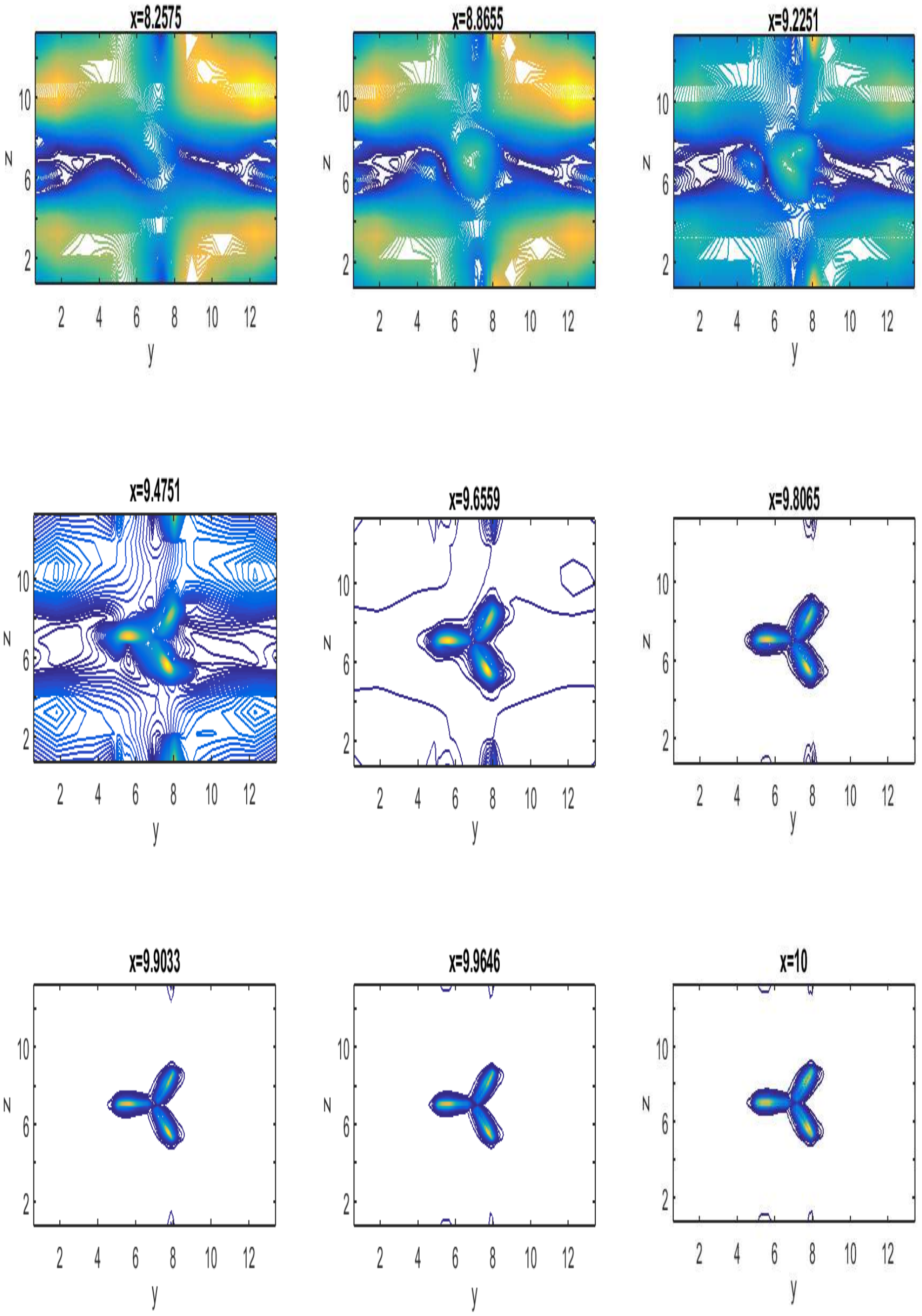


Figure 4.11: Computed magnitude of vorticity at y-z planes for $U_0 = 10\frac{m}{s}$.

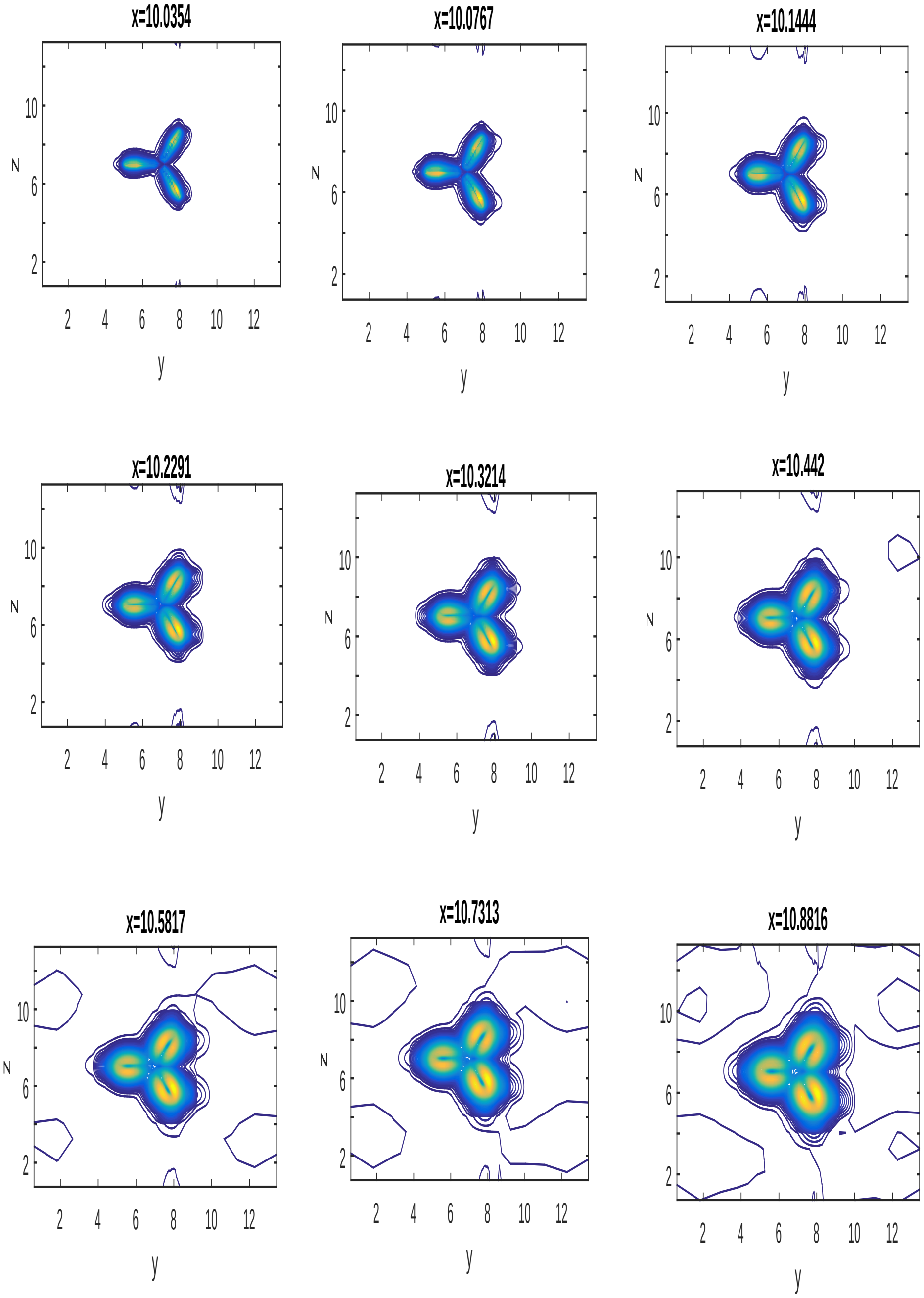


Figure 4.12: Computed magnitude of vorticity behind the turbine.

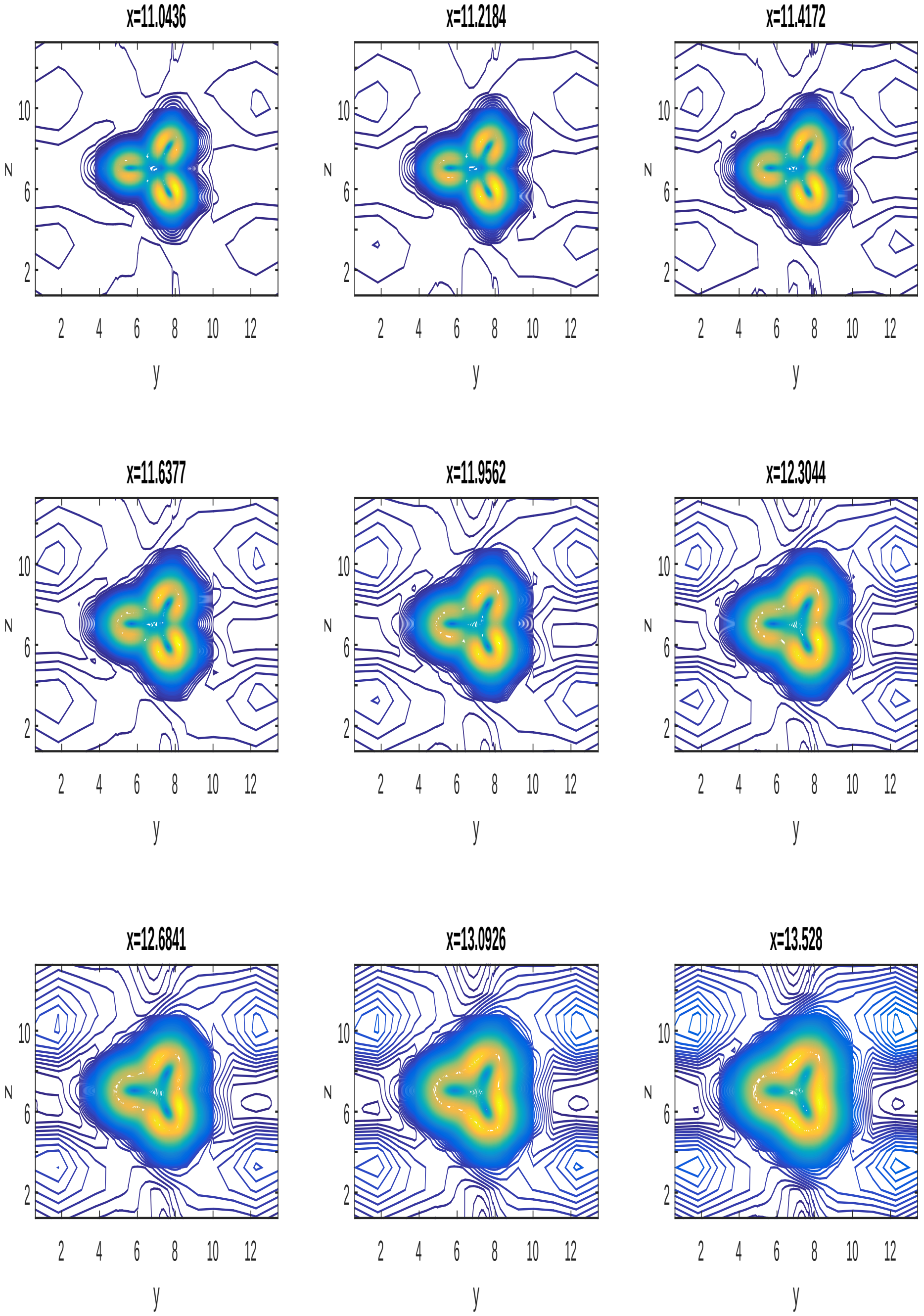


Figure 4.13: Computed magnitude of vorticity at y-z planes in near wake.

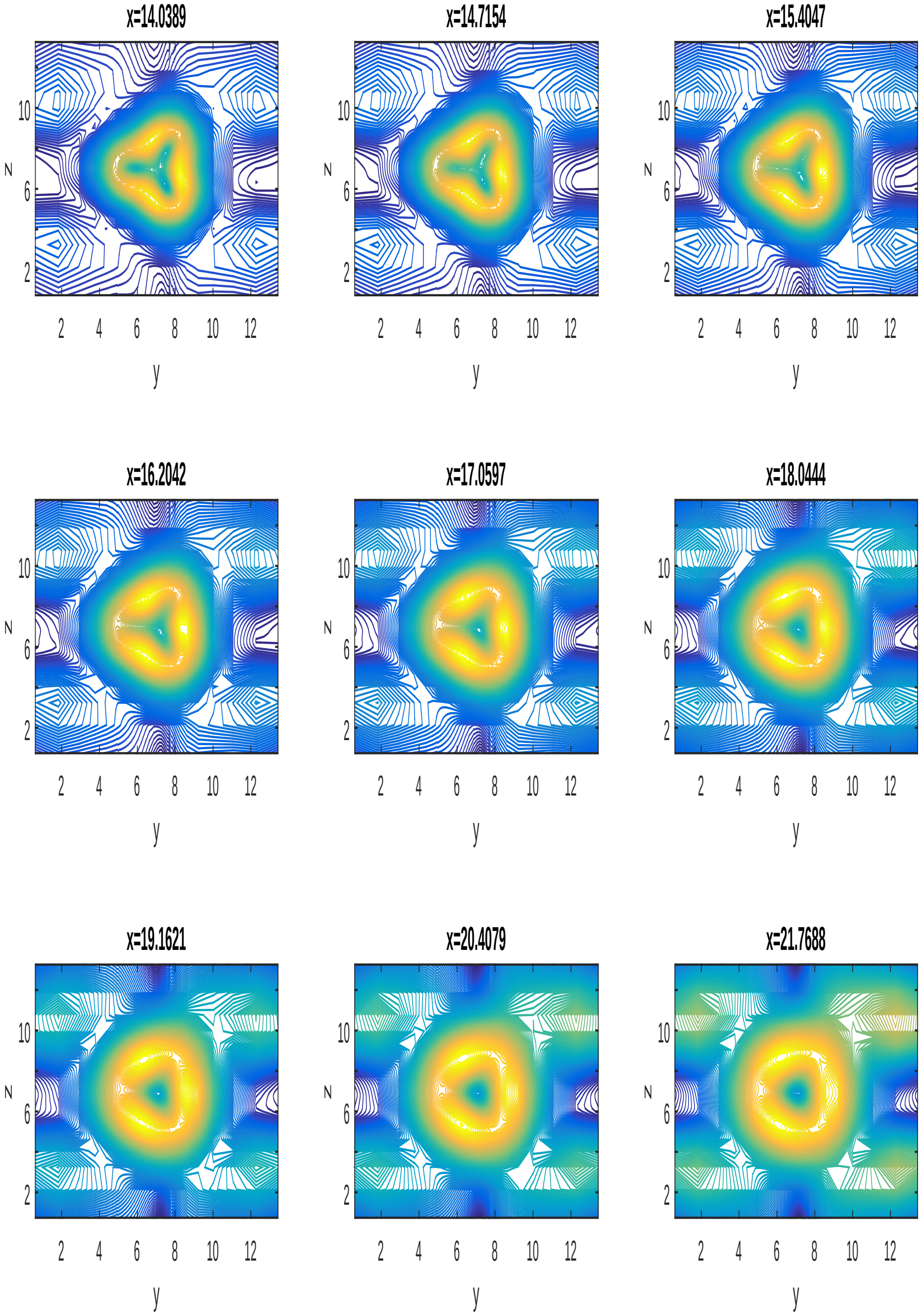


Figure 4.14: Computed magnitude of vorticity at y-z planes in far wake.

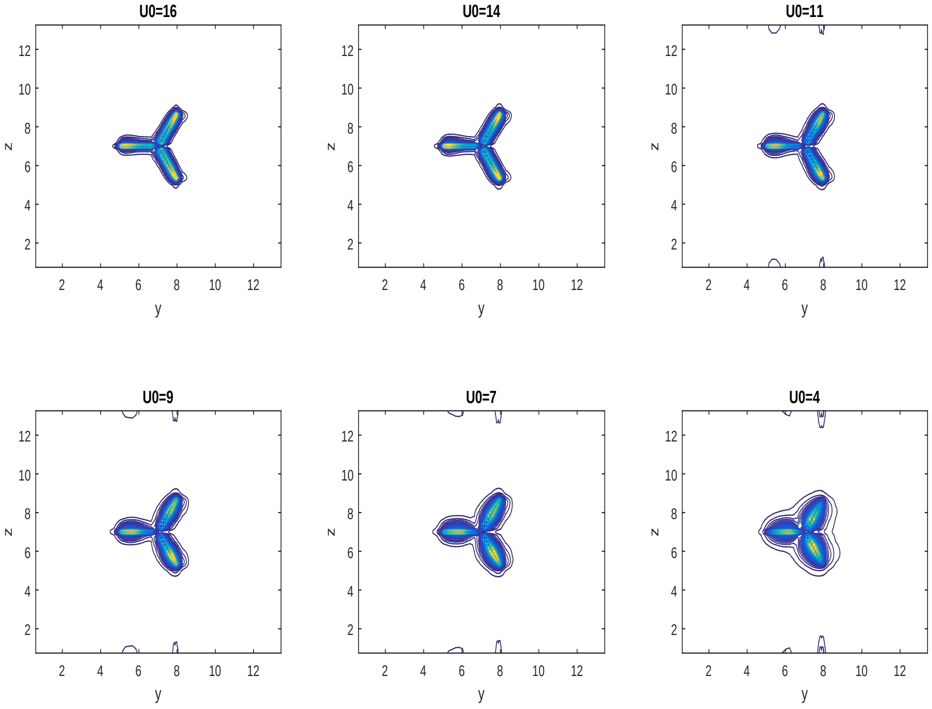


Figure 4.15: Computed magnitude of vorticity for different wind speed.

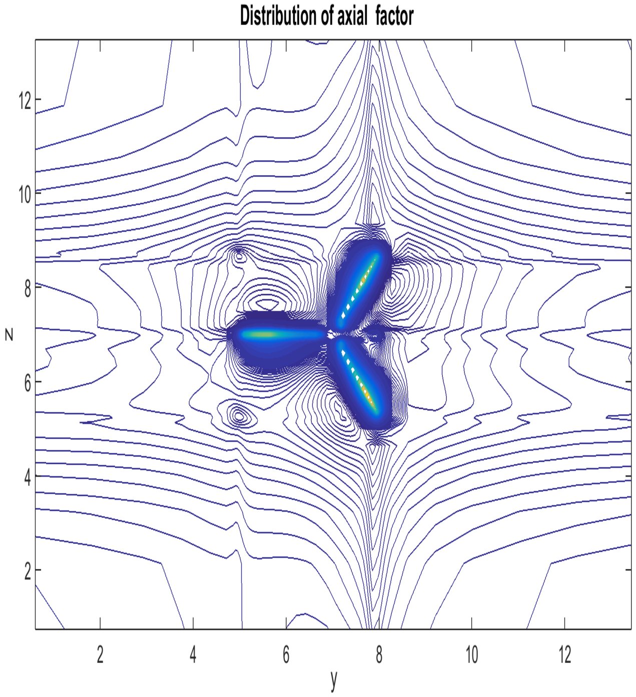


Figure 4.16: Distribution of axial factor in the rotor plane for $U_0 = 10$.

the far wake which is due to the fact that at the far wake the wind speed will recover to the free stream value. Figures 4.19, 4.20 and 4.21 show the distribution of averaged axial velocity in y-z plane in x-direction. From these figures, as the wind approaches the rotor the axial velocity keeps dropping and then immediately behind the wind turbine, it keeps increasing until it recovers in the far wake. It is noticed that even in the far wake the wind speed has not fully recovered and this is one proof that joint optimization of multiple wind turbines is necessary for improving the performance of wind turbines in the wind farm.

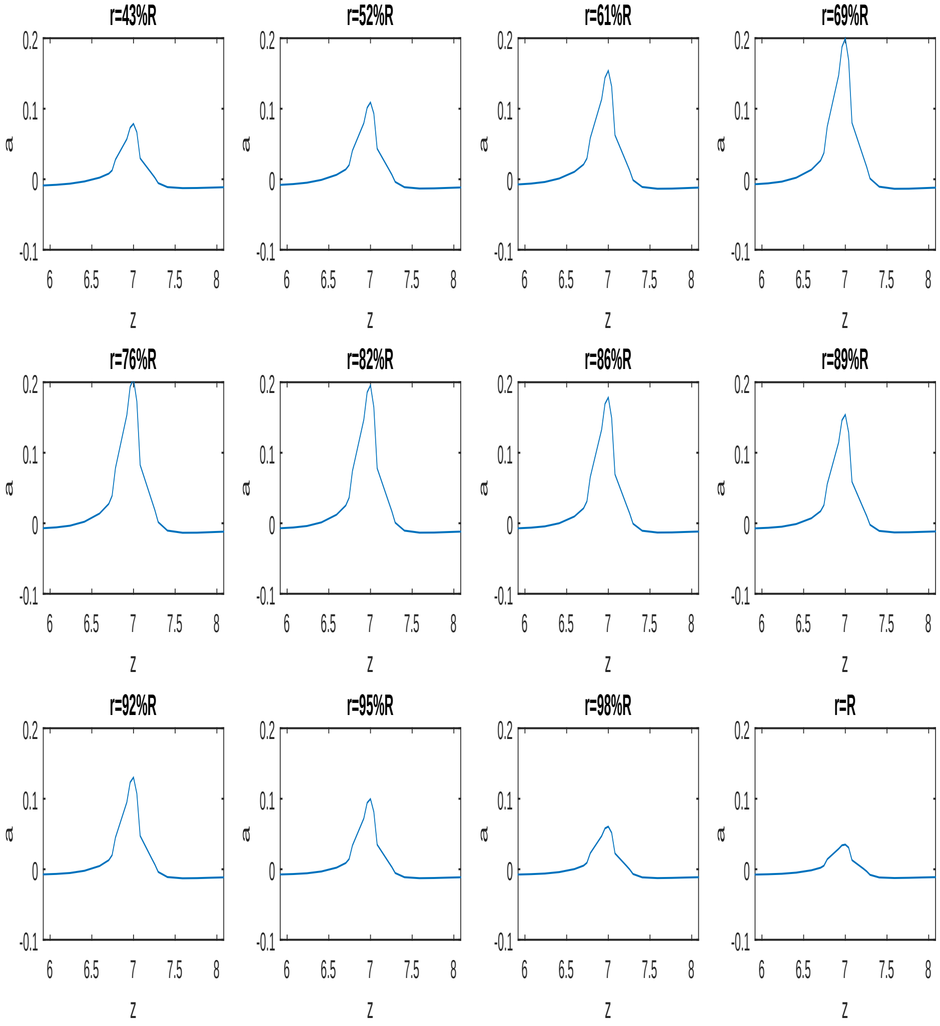


Figure 4.17: Distribution of axial factor in z-direction.

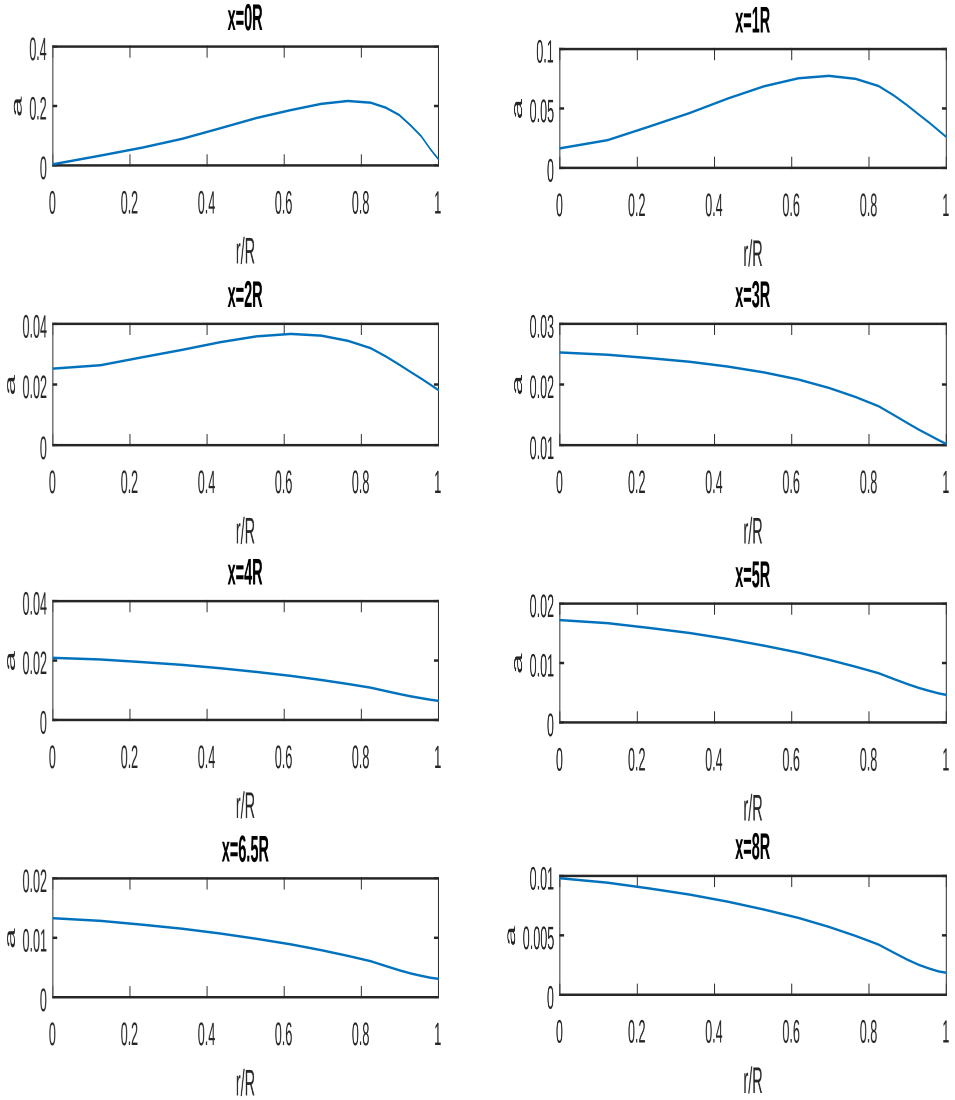


Figure 4.18: Distribution of axial factor along the blade (wake region).

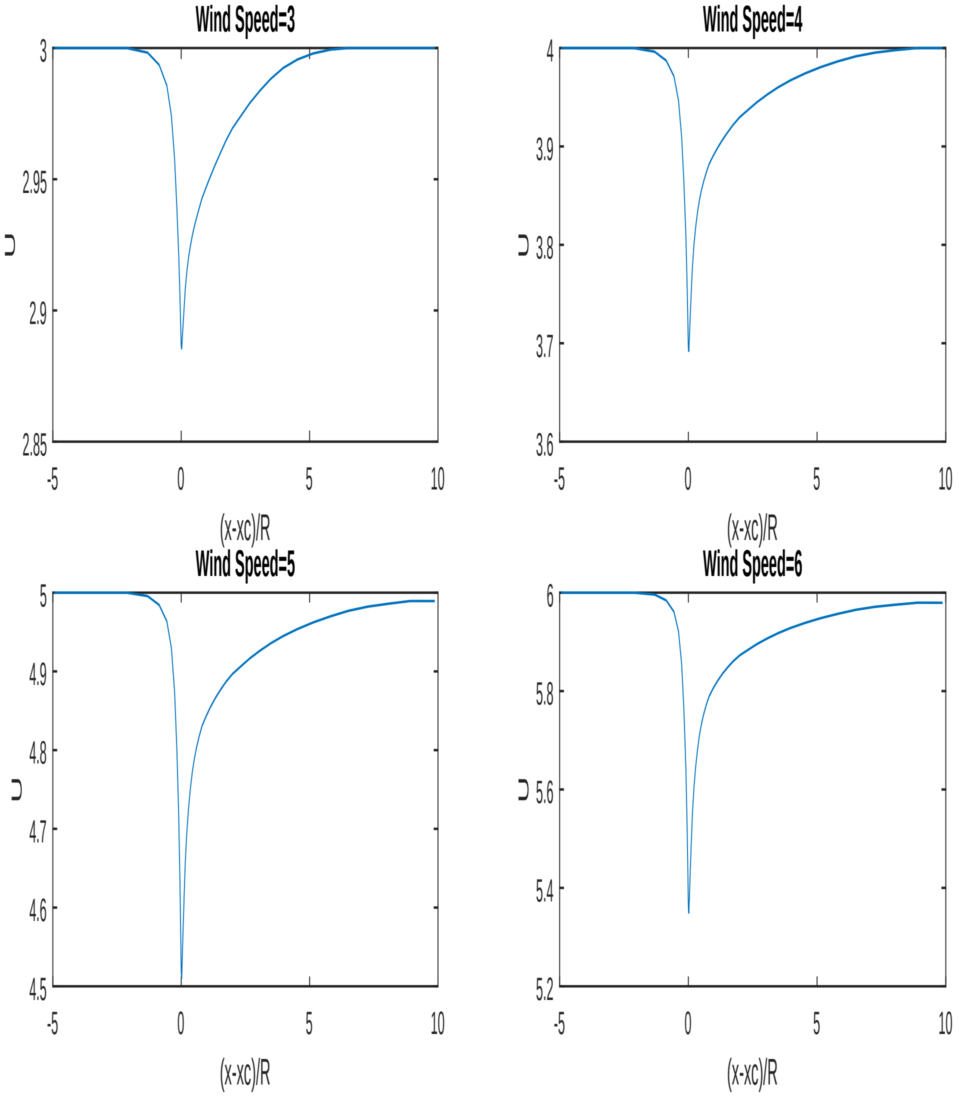


Figure 4.19: Velocity profile in x-direction for wind speeds $U_0 = 3, \ldots, 6$.

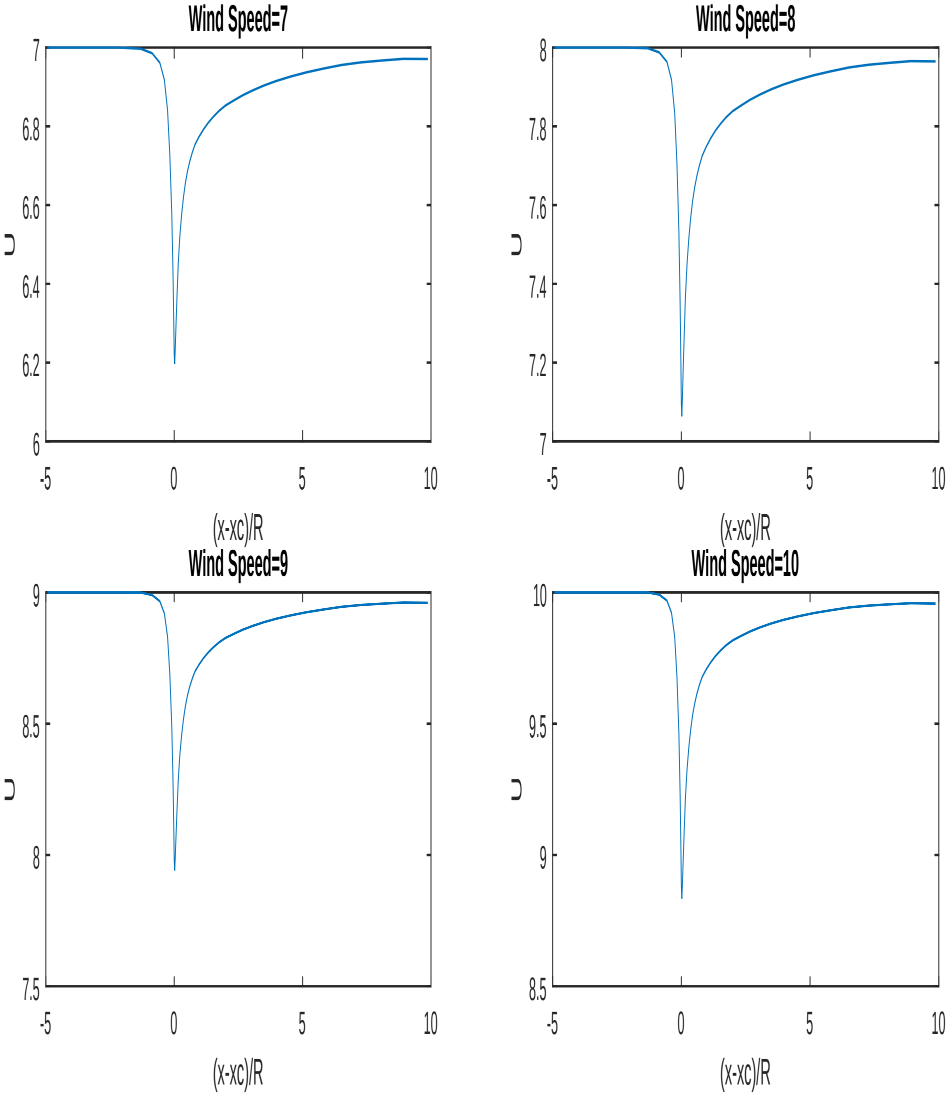


Figure 4.20: Velocity profile in x-direction for wind speeds $U_0 = 7, \ldots, 10$.

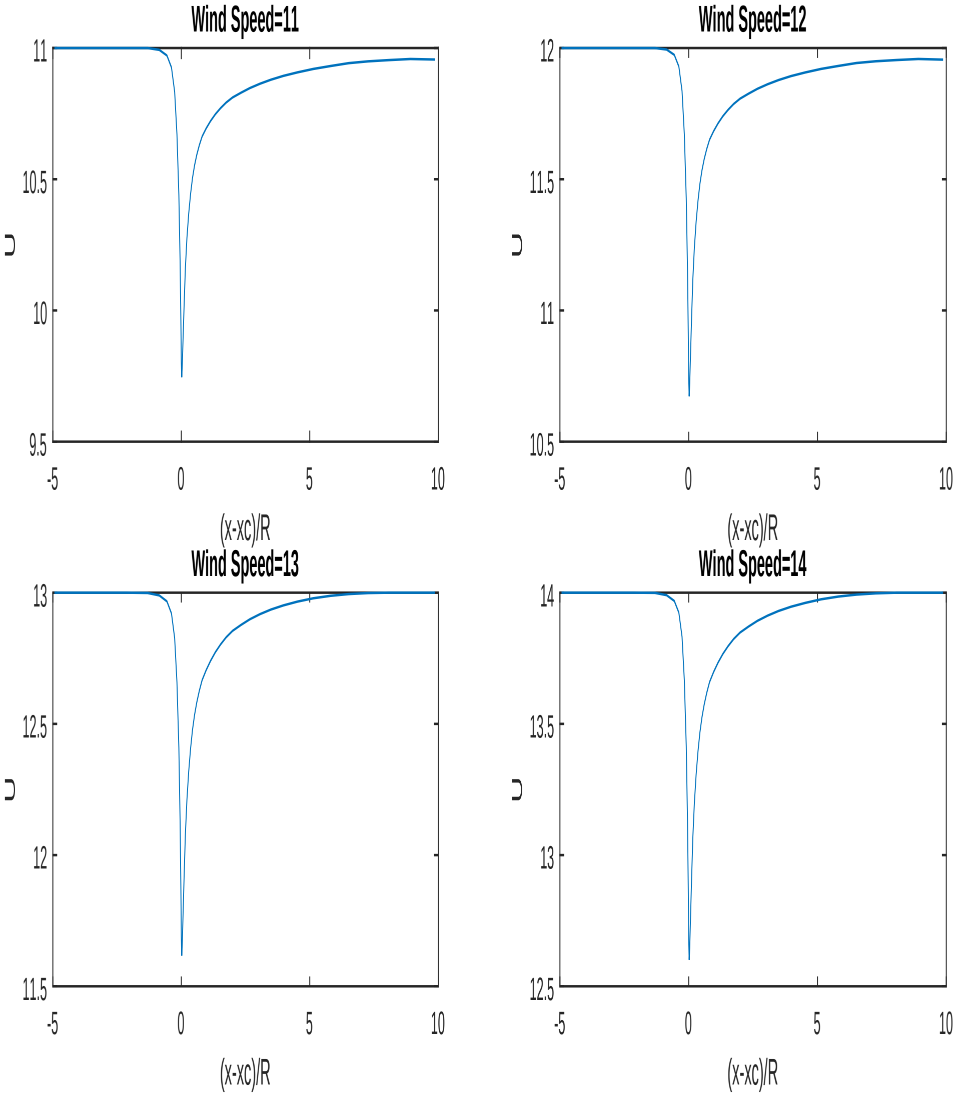


Figure 4.21: Velocity profile in x-direction for wind speeds $U_0 = 11, \ldots, 14$.

4.8.3 Numerical Results for NACA 23012

In this section, we present two numerical case studies to test the efficiency and accuracy of the proposed numerical algorithm 7 as well as the developed parallel algorithm (4.30)-(4.64). In these numerical case studies, we use a three-bladed wind turbine with a rotor diameter of 10 meters where the blade sections consist of NACA 23012 series airfoils. The chord length and the manufactured twist angle of this turbine are obtained using the formulas given in (3.48) and (3.44).

To capture the gradients of the flow field, grid points are concentrated near the blade tips and stretched in the x, y and z-direction. In the case of one turbine, the resulting grid consists of 86 grid points in the axial direction, 90 points in the y-direction and 102 points in the z-direction. In the case of two turbines, the resulting grid consists of 106 grid points in the axial direction, 90 points in the y-direction and 102 points in the z-direction. In the axial direction the grid spacing ranges from $dx = 0.02$ at the rotor plane to about $dx = 1.9476$ in the far wake, and in the y-direction the grid spacing takes values from $dy = 0.02$ near the tip to about $dy = 1.2150$ at the lateral boundary. Moreover, in the z-direction, the grid spacing takes values from $dz = 0.0346$ near the tip to about $dz = 1.4863$ at the lateral boundary. The computations are carried out on a $100m \times 60m \times 60m$ computational domain and at an effective Reynolds number of $Re = U_0 * R/\nu = 5000$. Moreover, in all the experiments, the computational domain is decomposed into $2 \times 2 \times 2$ subdomains, and the doubly truncated Weibull distribution is used to model the random behavior of the incoming wind speed. Here, we use the doubly truncated Weibull distribution with the shape parameter $k_c = 3$, the scale parameter $s_c = 9$, the lower limit $a = 6$ and the upper limit $b = 15$. To ensure that the flow is fully developed in most of the wake, we run the simulation to about t=20 with a time step of dt=2×10^{-3} which corresponds to 10000 time steps.

4.8.3.1 One Turbine

We consider the case that there is only one turbine in the wind farm and use two approaches to find the optimal operating points of this free-standing wind turbine. The first approach is to solve a deterministic model, and the second approach is to solve a stochastic model. The former approach replaces the random incoming wind speed by the expected value of the doubly truncated Weibull distribution which is $9.020\frac{m}{s}$ (4.8). Since the incoming wind speed is $9.020\frac{m}{s}$, the optimal rotational velocity and the optimal pitch angle of the wind turbine are $\beta_d^* = 1.1215°$ and $\Omega_d^* = 1.6840$ rad/s, respectively, see Table 3.3. At this optimal operating points, we calculate the expected power output of the turbine as follows. First, we use Algorithm 6 to generate 5000 scenarios. Each scenario represents the incoming wind speed, and it is generated using doubly truncated Weibull distribution. Moreover, each scenario is rounded to the nearest integer less than or equal that scenario. Then, for each scenario, we use the developed parallel algorithm (4.30)-(4.64) combined with the actuator line technique to compute the power generated by the turbine. Having calculated the power generated by the turbine for each scenario, we evaluate the expected power output of the turbine. The results are tabulated in Table 4.1. From this table, we observe that if we operate the turbine at deterministic optimal points, then the expected power output, the angle of attack, the thrust and the torque are 21.0339kw power, 15.0587 degrees, 4.8115 kilonewton and 0.7536 kilonewton, respectively.

Now, we apply the latter approach which explicitly includes the randomness of the incoming wind speed to find the optimal operating points of the free-standing wind turbine. In the latter approach, we solve the scenario-based approximation model (4.9) by the developed Algorithm 7 to find the optimal operating points of the wind turbine. Algorithm 7 starts with a feasible initial operating point and, in step 3, it generates 5000 scenarios by Algorithm 6. Likewise for the deterministic approach, the scenarios are generated using doubly truncated Weibull distribution, and each scenario is rounded to the nearest integer less than or equal that scenario. For each scenario, Algorithm 7 uses the developed parallel algorithm (4.30)-(4.64) combined with the actuator line technique to compute the power generated by the turbine. Having calculated the power generated by the turbine for each scenario,

Table 4.1: Deterministic optimal operating point of NACA 23012.

β	Ω	α	T	Ψ	P
			$U_0 = 9.020\frac{m}{s}$		
1.1213	1.6837	15.0587	4.8115	0.7536	21.0339

we compute the expected power output of the turbine which is the objective function of the scenario-based approximation model (4.9). Algorithm 7 repeats these process until convergence. The optimal pitch angle and the rotational speed obtained by Algorithm 7 are $\beta_s^* = 1.5201°$ and $\Omega_s^* = 2.2048$ rad/s, respectively. Now, we compare these optimal operating points with the optimal operating points obtained using the deterministic approach in a sense that which one leads to a higher expected power output. In this regard, we compute the expected power output of the turbine operating at the deterministic optimal points as well as the expected power output of the turbine operating at the stochastic optimal points. The results are tabulated in the Table 4.2. From this table, the expected power output of the turbine operating at the stochastic optimal points is 22.3990, and the expected power output of the turbine operating at the deterministic optimal points is 21.0339. Therefore, by optimizing the turbines' operation while considering the randomness of the incoming wind speed, we can gain an additional 6.46% in the expected power since $22.3990 = (1 + 6.46\%)21.0339$.

Table 4.2: Stochastic optimal operating point of NACA 23012.

	β	Ω	α	T	Ψ	P
stochastic	1.5201	2.2048	10.3990	6.5451	0.7056	22.3990
deterministic	1.1215	1.6840	15.0587	4.8115	0.7536	21.0339
					Gain:	6.46%

Table 4.3: Joint optimal operating points of two NACA 23012.

	Turbine	β	Ω	α	T	Ψ	P	P_{Total}	Gain
Joint Stochastic	1st,	0.331	1.722	16.636	5.657	0.734	20.183		
	2nd,	0.827	1.611	15.648	4.979	0.733	19.623	39.806	—
Non-Joint Deterministic	1st,	1.121	1.684	13.042	4.357	0.600	17.771		
	2nd,	1.121	1.684	12.884	4.333	0.587	17.494	35.265	12.88%
Joint Deterministic	1st,	1.164	1.591	15.143	4.377	0.705	19.014		
	2nd,	0.747	1.308	25.818	4.424	0.700	17.511	36.525	8.98%
Non-Joint Stochastic	1st,	1.520	2.204	8.760	5.958	0.561	19.086		
	2nd,	1.520	2.204	8.654	5.924	0.549	18.654	37.740	5.47%

4.8.3.2 Two Turbines

We consider the case that there are two turbines, four rotor diameters apart, in the wind farm and use four approaches to find the optimal operating points of the wind turbines. The first approach is to solve a deterministic model without considering the wake effect, the second approach is to solve a deterministic model while considering the wake effect, the third approach is to solve a stochastic model without considering the wake effect, and the last approach is to solve a stochastic model while considering the wake effect. The first two approaches replace the random incoming wind speed by the expected value of the doubly truncated Weibull distribution which is $9.020\frac{m}{s}$ (4.8). Since the incoming wind speed is $9.020\frac{m}{s}$, the first approach finds $\beta^*_{1njd} = \beta^*_{2njd} = 1.1215°$ and $\Omega^*_{1njd} = \Omega^*_{2njd} = 1.6840$ rad/s for the optimal operating points of wind turbines, see Table 3.3. Moreover, since the incoming wind speed is $9.020\frac{m}{s}$, applying the second approach, we find that the optimal operating points of the upstream turbine are $\beta^*_{1jd} = 1.1646°$ and $\Omega^*_{1jd} = 1.5915$, and the optimal operating points of the downstream turbine are $\beta^*_{2jd} = 0.747°$ and $\Omega^*_{2jd} = 1.3086$, see Table 3.5. The third approach takes into account the random behavior of the wind speed on the boundary of the wind farm. However, it ignores the wake interaction between the upstream and downstream turbine. Using the third approach, we find $\beta^*_{1njs} = \beta^*_{2njs} = 1.5201°$ and $\Omega^*_{1njs} = \Omega^*_{2njs} = 2.2048$ rad/s for the optimal operating points of the wind turbines, see Table 4.2. Finally, we use the last approach which considers the randomness of the incoming wind speed as well as the wake interaction between the upstream and downstream turbine to find the optimal operating points of wind turbines. In this approach, we solve the scenario-based approximation model (4.9) by the developed Algorithm 7 to find the optimal operating points of the wind turbines. Algorithm 7 starts with a feasible initial operating point and, at each iterate, it computes the expected total power output in the same way as when we compute the expected power out of the free-standing turbine in the stochastic approach. Using the last approach, we find that the optimal operating points of the upstream turbine are $\beta^*_{1js} = 0.3310°$ and $\Omega^*_{1js} = 1.7222$ rad/s, and the optimal operating points of the downstream turbine are $\beta^*_{2js} = 0.8275°$ and $\Omega^*_{2js} = 1.6114$ rad/s. Now, we compare the optimal operating points obtained by these four approaches in a sense that which one leads to a higher expected total power output. In this regard, we compute the expected total power output at the optimal operating points associated with each approach. The results are tabulated in Table 4.3. From this table, the expected total power output at optimal operating points associated with the first, second, third and fourth approach are 35.2656kw , 36.5255kw , 37.7408kw and 39.8067kw, respectively. Therefore, using the last approach, the gain of 12.88%, 8.98% and 5.47% in the expected total power output are obtained with respect to the other three approaches. In conclusion, we can improve the performance of the wind farm by considering the randomness of the incoming wind speed as well as the wake interaction among the turbines.

4.9 Conclusion

We studied how to optimize wind turbines' power production in a wind farm where the wind speed on the boundary of the wind farm is random. We modeled the three-dimensional flow field in the wind farm by combining the actuator line model and the solutions of Navier-Stokes equations while taking into account the randomness of incoming wind speed. Furthermore, we developed a parallel scheme to solve three-dimensional Navier-Stokes equations in velocity-vorticity formulation which was ultimately used in the simulation of the stochastic optimization model. The parallel scheme and the model were further validated by a benchmark used in the literature and experimental data. It was shown that by taking into account the randomness of incoming wind speed and optimizing the total power, we can improve the performance of wind turbines in a wind farm. We found that by optimizing the turbines' operation and taking into account the randomness of wind speed, we can gain an additional 9%, in total power.

Bibliography

[1] J. D. Anderson. *Computational Fluid Dynamics - The Basics with Applications*. McGraw-Hill, 1995.

[2] W. E. Arnoldi. The principle of minimized iteration in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, 9:17–29, 1951.

[3] O. Axelsson. Conjugate gradient type methods for unsymmetric and inconsistent systems of linear equations. *Linear Algebra and its Applications*, 29:1–16, 1980.

[4] M. Azaiez and A. Quarteroni. A spectral Stokes solver in domain decomposition methods. *Journal of Contemporary Mathematical*, pages 151–156, 1994.

[5] K. Aziz and A. Settari. *Petroleum Reservoir Simulation*. Applied Science Publishers Ltd., London, 1979.

[6] R. J. Barthelmie, S. T. Frandsen, K. Hansen, J. G. Schepers, K. Rados, W. Schlez, A. Neubert, L. E. Jensen, and S. Neckelmann. Modelling the impact of wakes on power output at nysted and horns Rev. *European Wind Energy Conference*, 2009.

[7] G. Batchelor. *Introduction to Fluid Dynamics*. Cambridge University Press, London, 1967.

[8] J. Bear. *Hydraulics of Groundwater*. McGraw Hill, New York, 1978.

[9] K. Black. Polynomial collocation using a domain decomposition solution to parabolic PDEs via the penalty method and explicit/implicit time marching. *Journal of Scientific Computing*, 7:313–338, 1992.

[10] S. P. Breton, F. N. Coton, and G. Moe. A study on rotational effects and different stall delay models using a prescribed wake vortex scheme and NREL phase VI experiment data. *Wind Energy*, 11(5):459–482, 2008.

[11] R. Chandra. *Conjugate gradient methods for partial differential equations*. PhD thesis, Computer Science Department and Yale University, New Haven, CT, 1978.

[12] J. C. Chapman, J. A. C. Orme, and M. R. Willis. A robust blade element momentum theory model for tidal stream turbines including tip and hub loss corrections. *Journal of Marine Engineering and Technology*, 10:25–35, 2011.

[13] H. J. H. Clercx. A spectral solver for the Navier-Stokes equations in the velocity-vorticity formulation for flows with two nonperiodic directions. *Journal of Computational Physics*, 137:186–211, 1997.

[14] J. Conway. Analytical solutions for the actuator disk with variable radial distribution of load. *Journal of Fluid Mechanics*, 297:327–355, 1995.

[15] J. Conway. Exact actuator disk solution for non-uniform heavy loading and slipstream contraction. *Journal of Fluid Mechanics*, 365:235–267, 1998.

[16] A. Crespoc, L. J. Vermeera, and J. N. Sorensen. Wind turbine wake aerodynamics. *Progress in Aerospace Sciences*, 39:467–510, 2003.

[17] C. N. Dawson, Q. Du, and T. F. Dupont. A finite difference domain decomposition algorithm for numerical solution of the heat equation. *Mathematics of Computation*, 57:63–71, 1991.

[18] O. de Vries. *Fluid dynamic aspects of wind energy conversion.* Advisory Group For Aerospace Research and Development Neuilly-Seine (France), 1979.

[19] C. Du and D. Liang. An efficient S-DDM iterative approach for compressible contamination fluid flows in porous media. *Journal of Computational Physics*, 229:4501–4521, 2010.

[20] Q. Du, M. Mu, and Z. N. Wu. Efficient parallel algorithms for parabolic problems. *SIAM Journal on Numerical Analysis*, 39:1469–1487, 2001.

[21] S. C. Eisenstat, H. C. Elman, and M. H. Schultz. Variational iterative methods for nonsymmetric systems of linear equations. *SIAM Journal on Numerical Analysis*, 20:345–357, 1983.

[22] H. C. Elman. *Iterative methods for large sparse nonsymmetric systems of linear equations*. PhD thesis, Computer Science Department and Yale Univesity, New Haven, CT, 1982.

[23] A. M. Elshabka and T. J. Chung. Numerical solution of three dimensional stream function vector components of vorticity transport equations. *Computer Methods in Applied Mechanics and Engineering*, 170:131– 153, 1999.

[24] I. Fejtek and L. Roberts. Navier-Stokes computation of wing/rotor interaction for a tilt rotor in hover. *The American Institute of Aeronautics and Astronautics*, 30(11):2595–2603, 1992.

[25] G. Fernandez and M. Hafez. Stable finite-element solution of the incompressible Navier-Stokes equations using linear interpolations for velocity and pressure. *Computer Methods in Applied Mechanics and Engineering*, 191:545–559, 2001.

[26] J. H. Ferziger and M. Peric. *Computational Methods for Fluid Dynamics*. Springer, 1996.

[27] M. Fortin and R. Aboulaich. Schwarz's decomposition method for incompressible flow problems. *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 333–349, 1987.

[28] P. Giannattasio and M. Napolitano. Optimal vorticity conditions for the node-centered finite-difference discretization of the second-order vorticity-velocity equations. *Journal of Computational Physics*, 127:208–217, 1996.

[29] J. H. Giese. Stream functions for three-dimensional flows. *Journal of Mathematical Physics*, 30(1):31–35, 1951.

[30] R. Glowinski and O. Pironneau. Finite element methods for Navier-Stokes equations. *Annual Review of Fluid Mechanics*, 24:167–204, 1992.

[31] M. D. Greenberg. Nonlinear actuator disc theory. *Z. Flugwiss.*, 203:90–98, 1972.

[32] G. Guj and F. Stella. Numerical solutions of the high-Re recirculating flows in vorticity-velocity form. *International Journal for Numerical Methods in Fluids*, 8:405– 416, 1988.

[33] G. Guj and F. Stella. A vorticity-velocity method for the numerical solution of 3D incompressible flows. *Journal of Computational Physics*, 106:286–298, 1993.

[34] A. L. Hageman and D. M. Young. *Applied Iterative Methods*. Academic Press, New York, 1981.

[35] C. J. Ho and F. H. Lin. Numerical simulation of three-dimensional incompressible flow by a new formulation. *International Journal for Numerical Methods in Fluids*, 23:1073–1084, 1996.

[36] T. v. Holten. The computation of aerodynamic loads on helicopter blades in forward flight using the method of the acceleration potential. *Report VTH-189, Department of Aerospace Engineering, Delft University of Technology, The Netherlands*, 1975.

[37] C. Huang, F. Li, T. Ding, and Z. Jin. Second-order cone programming-based optimal control strategy for wind energy conversion systems over complete operating regions. *IEEE Transactions on Sustainable Energy*, 6(1):263–271, 2015.

[38] G. Ingram. Wind turbine blade analysis using the blade element momentum method. *In Note on the BEM method; Durham University: Durham, NC, USA*, 2005.

[39] F. Iov, H. A. D., P. Sorensen, and F. Blaadjerg. Centralized power control of wind farm with doubly fed induction generators. *Renewable energy*, 31:935–941, 2006.

[40] K. C. Jea and D. M. Young. Generalized conjugate gradient acceleration of nonsymmetrizable iterative methods. *Linear Algebra and its Applications*, 34:159–194, 1980.

[41] B. N. Jiang, T. L. Lin, and L. A. Povinelli. Large-scale computation of incompressible viscous flow by least-squares finite element method. *Computer Methods in Applied Mechanics and Engineering*, 114:213–231, 1994.

[42] Z. Jin, F. Li, X. Ma, and S. Djouadi. Semi-definite programming for power output control in wind energy conversion system. *IEEE Transactions on Sustainable Energy*, 5(2):466–575, 2014.

[43] K. E. Johnson and N. Thomas. Wind farm control: Addressing the aerodynamic interaction among wind turbines. *American Control Conference*, pages 2104–2109, 2009.

[44] H. C. Ku, R. S. Hirsh, and T. D. Taylor. A pseudospectral method for solution of the three-dimensional incompressible Navier-Stokes equations. *Journal of Computational Physics*, 70:439–462, 1987.

[45] Y. Kuznetsov. New algorithms for approximate realization of implicit difference scheme. *Soviet Journal of Numerical Analysis and Mathematical Modelling*, 3:99–114, 1988.

[46] Y. M. Laevsky and S. V. Gololobov. Explicit-implicit domain decomposition methods for the solution of parabolic equations. *Siberian Mathematical Journal*, 36:506–516, 1995.

[47] A. Lapin and J. Pieska. On the parallel domain decomposition algorithms for time-dependent problems. *Lobachevskii Journal of Mathematics*, 10:27–44, 2002.

[48] R. D. Lazarov, I. D. Mishev, and P. S. Vassilevski. Finite volume methods for convection-diffusion problems. *SIAM Journal on Numerical Analysis*, 33:31–55, 1996.

[49] R. D. Lazarov, I. D. Mishev, and P. S. Vassilevski. Finite volume methods for convection-diffusion problems. *SIAM Journal on Numerical Analysis*, 33(1):31–55, 1996.

[50] D. Liang and W. Zhao. An optimal weighted upwinding covolume method on non-standard grids for convection diffusion problems in 2D. *International Journal for Numerical Methods in Engineering*, 67:553–577, 2006.

[51] J. G. Liu. Finite difference methods for 3D viscous incompressible flows in the vorticity-vector-potential formulation on nonstaggered grids. *Journal of Computational Physics*, 138:57–82, 1997.

[52] S. Liu and I. Janajreh. Development and application of an improved blade element momentum method model on horizontal axis wind turbines. *International Journal of Energy and Environmental Engineering*, 3:30, 2012.

[53] H. A. Madsen. A CFD analysis for the actuator disc flow compared with momentum theory results. *Proceedings of the 10th IEA Symposium on the Aerodynamics of Wind Turbines, Pedersen, B. Maribo Pedersen, Department of Fluid Mechanics, Technical University of Denmark*, pages 109–124, 1996.

[54] H. A. Madsen, R. Mikkelsen, S. Oye, C. Bak, and J. Johansen. A detailed investigation of the blade element momentum (BEM) model based on analytical and numerical results and proposal for modifications of the BEM model. *Journal of Physics: Conference Series, 75:012016.*

[55] R. Malki, A. J. Williams, T. N. Croft, M. Togneri, and I. Masters. A coupled blade element momentum-computational fluid dynamics model for evaluating

tidal stream turbine performance. *Applied Mathematical Modelling*, 37:3006–3020, 2013.

[56] G. D. Mallinson and G. de Vahl Davis. The method of the false transient for the solution of coupled elliptic equation. *Journal of Computational Physics*, 12:435–461, 1973.

[57] J. F. Manwell, J. G. McGowan, and A. L. Rogers. *Wind energy explained: theory, design and application.* A John Wiley and Sons Ltd, 2009.

[58] J. R. Marden, S. Ruben, and L. Pao. A model-free approach to wind farm control using game theoretic methods. *IEEE Transactions on Control Systems Technology*, pages 1207–1214, 2013.

[59] R. H. Miller. The aerodynamic and dynamic analysis of horizontal axis wind turbines. *Journal of Wind Engineering and Industrial Aerodynamics*, 1:329–340, 1983.

[60] P. J. Moriarty and A. C. Hansen. Aerodyn theory manual. *Technical report NREL/TP-500-36881*, 2005.

[61] K. W. Morton. *Numerical solution of Convection-diffusion Problems.* Chapman and Hall, London, 1996.

[62] J. Murua, R. Palacios, and J. Graham. Applications of the unsteady vortex-lattice method in aircraft aeroelasticity and flight dynamics. *Progress in Aerospace Sciences*, 55:46–72, 2012.

[63] M. Napolitano and G. Pascazio. A numerical method for the vorticity-velocity Navier-Stokes equations in two and three dimensions. *Computer and Fluids*, 19:489–495, 1991.

[64] K. A. Narayan, C. Schleeberger, and K. L. Bristow. Modelling seawater intrusion in the burdekin delta irrigation area. *Agricultural Water Management*, 89(3):217–28, 2007.

[65] H. E. Neustadter and D. A. Spera. Method for evaluating wind turbine wake effects on wind farm performance. *Journal of Solar Energy Engineering*, 107:240–243, 1985.

[66] A. Ning. A simple solution method for the blade element momentum equations with guaranteed convergence. *Wind Energy*, 7(9):1327–1345, 2014.

[67] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. New York: Springer-Verlag, 1999.

[68] C. C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, 12:617–624, 1975.

[69] J. Y. Parlarge. Water transport in soils. *Annual Review of Fluid Mechanics*, 2:77–102, 1980.

[70] B. N. Parlett. A new look at the lanczos algorithm for solving symmetric systems of linear equations. *Linear Algebra and its Applications*, 29:323–346, 1980.

[71] R. Peyret and T. D. Taylor. *Computational Methods for Fluid Flow*. Springer Series in Computational Physics, 1983.

[72] R. G. Rajagopalan and S. R. Mathur. Three dimensional analysis of a rotor in forward flight. *Journal of the American Helicopter Society*, 38(3), 1993.

[73] P. E. Rethore, P. V. D. Laan, and F. Zahle. Verification and validation of an actuator disc model. *Wind Energy*, 17(6):919–937, 2014.

[74] Y. Saad. Variations on arnoldi method for computing eigenelements of large unsymmetric matrices. *Linear Algebra and its Applications*, 34:269–295, 1980.

[75] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, Boston, 1996.

[76] Y. Saad and M. H. Schultz. A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7:856–869, 1986.

[77] B. Sanderse. Aerodynamics of wind turbine wakes literature review. *Energy Research Centre of the Netherlands*, 16, 2009.

[78] B. Sanderse, S. P. van der Pijl, and B. Koren. Review of computational fluid dynamics for wind turbine wake aerodynamics. *Wind Energy*, 14(7):799–819, 2011.

[79] J. V. Seguro and T. W. Lambert. Modern estimation of the parameters of the weibull wind speed distribution for wind energy analysis. *Journal of Wind Engineering and Industrial Aerodynamics*, 85:15–84, 2000.

[80] W. Z. Shen, J. N. Sorensen, and R. Mikkelsen. Tip loss correction for actuator/Navier–Stokes computations. *Journal of Solar Energy Engineering*, 127(2):209–213, 2005.

[81] W. Z. Shen, J. N. Sorensen, and X. Munduate. Analysis of wake states by a full-field actuator disc model. *Wind Energy*, 1:73–88, 1998.

[82] W. Z. S. Shen, W. J. Zhu, and J. N. Sorensen. Actuator line/Navier-Stokes computations for the MEXICO rotor: comparison with detailed measurements. *Wind Energy*, 15(5):811–825, 2012.

[83] X. Shen, X. Zhu, and Z. Du. Wind turbine aerodynamics and loads control in wind shear flow. *Energy*, 36(3):1424–34, 2011.

[84] Z. Q. Sheng, G. W. Yuan, and X. D. Hang. Unconditional stability of parallel difference schemes with second order accuracy for parabolic equation. *Applied Mathematics and Computation*, 184:1015–1031, 2007.

[85] H. S. Shi and H. L. Liao. Unconditional stability of corrected explicit-implicit domain decomposition algorithms for parallel approximation of heat equations. *SIAM Journal on Numerical Analysis*, 44:1584–1611, 2006.

[86] M. Shives and C. Crawford. Mesh and load distribution requirements for actuator line CFD simulations. *Wind Energy*, 16(8):1183–1196, 2013.

[87] F. Shoichi, T. Masahisa, and F. Yasuji. Extension to three dimensional problems of the upwind finite element scheme based on the choice of up and downwind points. *Computer Methods in Applied Mechanics and Engineering*, 112:109–131, 1994.

[88] F. J. Simoes and J. M. R. Graham. Prediction of loading on a horizontal axis wind turbine using a free vortex wake model. *13th BWEA Wind Energy Conference, Quarton, Mechanical Engineering Publications Limited, London*, pages 247–254, 1991.

[89] J. N. Sorensen and C. W. Kock. A model for unsteady rotor aerodynamics. *Journal of Wind Engineering and Industrial Aerodynamics*, 58:259–275, 1995.

[90] J. N. Sorensen and A. Myken. Unsteady actuator disc model for horizontal axis wind turbines. *Journal of Wind Engineering and Industrial Aerodynamics*, 39:139–149, 1992.

[91] J. N. Sorensen and W. Z. Shen. Numerical modeling of wind turbine wakes. *Journal of Fluids Engineering*, 124(2):393–9, 2002.

[92] J. N. Sorensen, W. Z. Shen, and X. Munduate. Analysis of wake states by a full-field actuator disc model. *Wind Energy*, 1:73–88, 1998.

[93] J. N. Sorensen and G. A. M. van Kuik. General momentum theory for wind turbines at low tip speed ratios. *Wind Energy*, 14(7):821–839, 2011.

[94] L. Q. Tang, T. Cheng, T. Tate, and H. Tsang. Transient solutions for three-dimensional lid-driven cavity flows by a least-squares finite element method. *International Journal for Numerical Methods in Fluids*, 21:413–432, 1995.

[95] F. Thomasset. Implementation of finite element methods for the Navier-Stokes equations. *Springer: Berlin*, 1981.

[96] P. Torres, J. W. van Wingerden, and M. Verhaegen. Modeling of the flow in wind farms for total power optimization. *IEEE International Conference on Control and Automation*, pages 18–21, 2011.

[97] N. Troldborg, J. N. Sorensen, and R. Mikkelsen. Actuator line simulation of wake of wind turbine operating in turbulent inflow. *Journal of Physics: Conference Series*, 75:012063, 2007.

[98] N. Troldborg, J. N. Sorensen, and R. Mikkelsen. Numerical simulations of wake characteristics of a wind turbine in uniform inflow. *Wind Energy*, 13(1):86–99, 2010.

[99] J. Trujillo and G. Karniadakis. A penalty method for the vorticity-velocity formulation. *Journal of Computational Physics*, 149:32–581, 1999.

[100] G. J. W. van Bussel. The aerodynamics of horizontal axis wind turbine rotors explored with asymptotic expansion methods. *Doctoral dissertation, Technische Universiteit Delft, The Netherlands*, 1995.

[101] H. Versteeg and W. Malalasekera. *An introduction to computational fluid dynamics: the finite volume method.* Pearson Prentice Hall, 2007.

[102] S. G. Voutsinas, M. A. Beleiss, and K. G. Rados. Investigation of the yawed operation of wind turbines by means of a vortex particle method,. *AGARD Conference Proceedings*, 552:11.1–11.11, 1995.

[103] C. Wang and J. Liu. Analysis of finite difference schemes for unsteady Navier-Stokes equations in vorticity formulation. *Journal of Numerical Mathematics*, 91(2):543–576, 2002.

[104] E. Weinan and J. Liu. Essentially compact schemes for unsteady viscous incompressible flows. *Journal of Computational Physics*, 126:122–138, 1996.

[105] K. L. Wong and A. J. Baker. A 3D incompressible Navier-Stokes velocity-vorticity weak form finite element algorithm. *International Journal for Numerical Methods in Fluids*, 38:99–123, 2002.

[106] T. Y. Wu. Flow through a heavily loaded actuator disc. *Ship Technology Research*, 9:134–138, 1962.

[107] X. H. Wu, J. Z. Wu, and J. M. Wu. Effective vorticity-velocity formulations for three-dimensional incompressible viscous flows. *Journal of Computational Physics*, 122:68–82, 1995.

[108] D. L. Young, J. T. Chang, and T. I. Eldho. A coupled BEM and arbitrary lagrangian-eulerian FEM model for the solution of two-dimensional laminar flows in external flow fields. *International Journal for Numerical Methods in Engineering*, 51:1053 – 1077, 2001.

[109] D. L. Young, Y. H. Liu, and T. I. Eldho. A combined BEM-FEM model for the velocity-vorticity formulation of the Navier-Stokes equations in three dimensions. *Engineering Analysis with Boundary Elements*, 24:307–316, 2000.

[110] D. L. Young, S. K. Yang, and T. I. Eldho. Solution of the Navier-Stokes equations in velocity-vorticity form using a eulerian-lagrangian boundary element method. *International Journal for Numerical Methods in Fluids*, 34:627–650, 2000.

[111] G. W. Yuan, X. D. Hang, and Z. Q. Sheng. Parallel difference schemes with interface extrapolation terms for quasi-linear parabolic systems. *Science in China Series A: Mathematics*, 50(2):253–275, 2007.

[112] G. W. Yuan and X. D. Sheng. The unconditional stability of parallel difference schemes with second order convergence for nonlinear parabolic system. *Journal of Partial Differential Equations*, 20:45–64, 2007.

[113] L. Zhu, G. Yuan, and Q. Du. Explicit implicit predictor corrector domain decomposition method for time dependent multi-dimensional convection diffusion equations. *Numerical Mathematics: Theory, Methods and Applications*, 2(3):301–325, 2009.

[114] Y. Zhuang and X. Sun. Stabilized explicit-implicit domain decomposition methods for the numerical solution of parabolic equations. *SIAM Journal on Scientific Computing*, 24:335–358, 2002.