DTU Wind and Energy Systems

Finite element method for wind turbine wake modelling

Author:

Julie Paddeu

DTU Wind-M-0766

July 2024

Author:
Julie Paddeu

Title:
Finite element method for wind turbine wake modelling

DTU Wind-M-0766
July 2024

ECTS: 45

Education: Master of Science

Supervisors:

Alexander Raul Meyer Forsting

DTU Wind & Energy Systems

Delphine De Tavernier
TU Delft

Norbert Warncke
Siemens Gamesa

Remarks:
This report is submitted as partial fulfillment of the requirements for graduation in the above education at the Technical University of Denmark.

Technical University of Denmark
Department of Wind and Energy Systems
Frederiksborgvej 399
4000 Roskilde
Denmark
www.wind.dtu.dk

Finite element method for wind turbine wake modelling

Master's thesis in Wind Energy

Julie Paddeu

Finite element method for wind turbine wake modelling

Master's thesis in Wind Energy

Thesis report

by

Julie Paddeu

to obtain the degrees of

Master of Science	**Master of Science**
in Wind Energy	in Aerospace Engineering
at Technical University of Denmark	at Delft University of Technology

to be defended publicly on July 30th, 2024

Thesis committee:

Chair:	Prof. Dr. S. J. Watson	TU Delft - Professor of Wind Energy Systems
Censor:	A. Schroll	SDU - Professor in Applied Mathematics
Supervisors:	Dr. N.G.W. Warncke	Siemens Gamesa - Loads Engineer
	Dr. D.A.M. De Tavernier	TU Delft - Assistant Professor
	Dr. A.R. Meyer Forsting	DTU - Senior Researcher
External examiner:	Dr. A. Palha da Silva Clérigo	Tu Delft - Associate Professor
Place:	Faculty of Aerospace Engineering, Delft	
Project Duration:	Novermber, 2023 - July, 2024	
Student number:	TU Delft	4997522
	DTU	s223023

Delft
University of
Technology

Executive Summary

As the demand for renewable energy rises, the impact of wake effects on wind farm performance and efficiency has become a primary focus for both industry professionals and academic researchers. However, to effectively mitigate these wake effects and improve wind farm efficiency, there is a critical need for an accurate wake representation. This requires well-established medium-fidelity wake models, which are currently lacking. To achieve this goal, it is essential to develop a wake model capable of accurately generating the axial wind profile within the wake. This model is crucial for precisely assessing the loads on downstream wind turbines and evaluating their energy production potential. Moreover, it provides valuable insights into optimising methods to mitigate wake effects and improve overall wind farm efficiency. For the model to be truly beneficial to the research effort and industry applications, it must demonstrate an increased accuracy with a limited increase in computational cost.

This thesis delves into medium-fidelity wake models to enhance the precision of calculations and predictions concerning wind turbine fatigue and aeroelastic loading. It specifically examines the use of a finite element method with Hermite interpolation basis functions as a numerical approach to solve the Ainslie wake model and achieve these objectives. Additionally, the study employs a downstream marching scheme to solve the partial differential equations and introduces the Newton-Raphson method to address non-linearities within the model. The potential benefit of using a finite element method lies in its potential for improved stability compared to the finite volume method, which shows satisfactory performance but limited stability, and superior performance compared to the spectral method, which has shown to exhibit poor conservative properties and instabilities for higher mode numbers. Furthermore, this thesis aims to bridge the gap between modelling and reality by implementing a pressure Poisson equation. Specifically, the focus has been on implementing the pressure component resulting from the forcing in the pressure Poisson equation.

Implementing Hermite interpolation basis functions posed several challenges. Increasing their order caused significant ill-conditioning and greater sensitivity to mesh quality, reducing their effectiveness for this application. Additionally, higher-order Hermite functions proved ineffective for solving the continuity equation due to the presence of odd basis functions. In contrast, piecewise linear basis functions have shown to be compatible with the continuity equation and were effective in solving the diffusion term. Despite this, stability issues emerged, likely due to the Ladyzhenskaya-Babuka-Brezzi condition not being met.

The implementation of forcing in the finite volume Ainslie wake model demonstrated mass and momentum conservation, and the axial velocity has been shown to align with literature. However, the dependency of the vortex strength on the domain size complicates the determination of a correct converged value for Γ. Furthermore, comparisons of the velocity field with the IEC 61400 standard implementation without the inclusion of a forcing term revealed that the forcing term implementation more accurately reflects the real flow dynamics, capturing both the blockage effect and the edge force effect.

Overall, this thesis demonstrates that the investigated method does not provide a stable solver for the Ainslie wake model. However, this does not rule out the potential of the finite element method for solving the Ainslie wake model. With adjustments to the basis functions and the application of Taylor-Hood elements, this method is still anticipated to deliver satisfactory results compared to existing methods. Furthermore, this thesis has laid the foundation for the inclusion of a Poisson solver. Specifically, it outlines the integration of pressure gradient resulting from actuator disk forcing and presents various formulations for the wind turbine forcing. However, further research is needed to address the calculation of the pressure induced by the velocity field.

Preface

This thesis signifies the end of an exceptional two-year journey, one that has taken me across borders and into the halls of two renowned universities. The path to this accomplishment has been both challenging and rewarding, filled with diverse experiences and profound learning on personal and academic levels. It has been a privilege to learn from distinguished professors and to engage with fellow students who have become lifelong friends.

Over these two years, especially while working on this thesis, I have grown into a more confident and self-assured person. A transformation which would not have been possible without the unwavering support and encouragement of some truly remarkable individuals which I would like to thank.

First and foremost, I would like to thank Norbert Warncke, my daily supervisor at Siemens Gamesa, for his unwavering support and mentorship. His expertise, readiness to answer questions, and generous sharing of time and knowledge have been invaluable in guiding me through this journey. Moreover, his profound insights have not only shaped the direction of my thesis but have also enriched its substance and depth. I am genuinely grateful to have had such an exceptional supervisor for my master's thesis, whose dedication to his students is evident in the care and effort he invests.

Next, I want to express my gratitude to my academic supervisors, Delphine De Tavernier from TU Delft and Alexander Meyer Forsting from DTU. Their confidence in my capabilities, understanding of faced the challenges, and consistent guidance throughout the thesis project have been invaluable. Moreover, their constructive feedback during our bi-weekly update meetings not only helped me stay focused but also motivated me to continually enhance my work.

Thirdly, I wish to convey my appreciation to Achim Schroll, Simon Watson, and Palha da Silva Clérigo for their enthusiasm towards this study and their willingness to contribute as members of the thesis committee.

Further, special appreciation goes to my parents, my brother, and my boyfriend for their unconditional love and support. To my parents, I am deeply thankful for the opportunities you have provided me. Your nurturing guidance has been a constant source of strength, helping me overcome challenges in my education and motivating me to pursue my goals with determination. Your guidance in resilience, honesty, compassion, and kindness has deeply impacted me, helping me to form the individual I am today. To my brother, I want to express my gratitude for always being there and making me feel completely at home during my time in Delft over the last two years. Your presence brought comfort and familiarity, making my experience even more special, especially during challenging times. To my boyfriend, thank you for keeping my spirits high during challenging moments over the past two and a half years of my studies, and for teaching me how to embrace and enjoy student life more fully.

Lastly, I wish to extend my profound gratitude to the exceptional group of students at the Hague office, Julia, Matti, Axel, and Leo. Your constant support has been a source of strength, guiding me through both the peaks and lows. Additionally, the moments we shared over dinners, weekends, and coffee breaks provided essential relaxation and connection for making this journey truly memorable. Additionally, to my cherished friends Suzanne, Aria, Aditya, and Te, I am deeply grateful for the laughter, comforting support, and invaluable guidance you have generously provided me with throughout this thesis period and over the past two years. I wouldn't have experienced the last two years the same way without you.

As I conclude this research journey, I have found it to be profoundly enriching and insightful. Exploring new ideas and overcoming challenges has not only expanded my knowledge but also strengthened my problem-solving abilities and resilience. These experiences have been transformative for my personal and professional development, enhancing both my skills and readiness to take on future endeavours.

Delft,
July 2024

Julie Paddeu

Contents

Nomenclature

List of Abbreviations

AD	Actuator Disk
ADM	Actuator Disk Model
AL	Actuator Line
ALM	Actuator Line Model
AS	Actuator Surface
BEM	Blade Element Momentum
CFD	Computational Fluid Dynamics
CG	Continuous Galerkin
DG	Discontinuous Galerkin
DNS	Direct Numerical Simulation
DTU	Technical University of Denmark
DWM	Dynamic Wake Meandering
FDM	Finite Difference Method
FEM	Finite Element Method
FVM	Finite Volume Method
GHG	Greenhouse Gas
IEC	International Electrotechnical Commission
LBB	Ladyzhenskaya-Babuka-Brezzi
LES	Large Eddy Simulation
LHS	Left-hand side
PDE	Partial Differential Equation
RANS	Reynolds Averaged Navier-Stokes
RHS	Right-hand side
rms	Root mean square
SGS	Sub-Grid Scale
TSL	Thin Shear Layer
TU Delft	Technical University of Delft

List of Greek Symbols

α	Unknown coefficients
δ	Vortex ring regularisation parameter
δ_{ij}	Kronecker Delta
ϵ_a	Ambient turbulence contribution
ϵ_t	Relative error in mass conservation diffusion
Γ	Strength vortex ring
Γ_b	Bound circulatoin
Γ_s	Shed vorticity
Γ_t	Trailed vorticity
$\hat{\sigma}$	Numerical flux on facets
κ	Diffusion coefficient
ν	Kinematic viscosity
ν_t	Eddy viscosity
Ω	General domain notation
Ω	Rotational speed wind turbine
ω	Rotational speed
$\partial\Omega$	Domain boundary
ϕ	Square integrable function
ϕ	Vector potential
ψ	Stream function
ψ_D	Deficit stream function
ρ	Density
σ^2	Variance
τ_{ij}	Sub-filter scale stress tensor
σ	Trial function
θ	Azimuthal dimension

ξ	Hermite interpolation basis functions

List of Latin Symbols

$\bar{K}$	Closure of a cell
$\bar{p}$	Time-averaged pressure
$\bar{S}_{ij}$	Time-averaged strain rate tensor
$\bar{u}$	Time-averaged velocity component
$\bar{u}_r$	Time-averaged radial velocity
$\bar{u}_x$	Time-averaged axial velocity
$\bar{u}_\theta$	Time-averaged rotational velocity
$\boldsymbol{w}$	Test function to function space W
Δt	Time step
$\hat{u}$	Numerical flux on facets
$\mathcal{F}$	Set of facets
$\mathcal{T}$	Set of cells
$\overline{u_i'u_j'}$	Reynolds stress tensor
∂K	Boundary simplicial element
$\tilde{p}$	Spatially filtered pressure
$\tilde{S}_{ij}$	Spatially filtered strain rate tensor
$\tilde{u}$	Spatially filtered velocity component
$\tilde{x}$	Non-dimensional downstream dimension
$\vec{f}$	Force density vector
$\vec{n}$	Normal vector
$\vec{P}$	Point at which force is experienced
$\vec{r}$	Distance between the vortex ring and the point at which $\vec{f}$ is calculated
$\vec{u}$	Velocity vector
$\vec{V}$	Point on the vortex ring
$\vec{x}$	Coordinate vector
A	General domain notation
A	Gobal Mass Matrix
a	Axial induction factor
a'	Azimuthal induction factor
a_m	Rotor-averaged axial induction factor
$a_{()}$	Axial basis function
B	Number of wind turbine blades
$b_{()}$	Radial basis function
C_T	Thrust coefficient
D	Wind turbine Diameter
E	Complete elliptical integral of the second kind
F	Common facet
F	Filter function
f	General function notation
f	Source term
f_c	Cut-off frequency
f_m	Galerkin projection function f
f_x	Axial force density
f_x	Radial force density
g	Dirichlet boundary condition
J	Jacobian Matrix
K	Complete elliptical integral of the first kind
K	Simplicial element
k	Turbulence kinetic energy
l_w	Length scale in wake shear layer
p	Pressure
p_0	Free-stream pressure
P_k	Polynomial of degree k
Q, $\mathbf{W}$	Function spaces
q, $\mathbf{w}$	Test functions
R	Wind turbine radius
r	Radial dimension
R_W	Wind turbine wake radius
R_{max}	Radial domain outer bound
S	General surface notation
s	Square of radial dimension r^2

T	Thrust force
t	Time variable
t_0	Time instant emission cascade element
U	Time-averaged axial velocity
u	General velocity component notation
u	Trial function
u'	Fluctuating velocity component
u_0	Average axial velocity at the rotor plane
u_1	Axial velocity far downstream
u_r	Radial velocity component
u'_r	Velocity fluctuations in radial direction
u_w	Velocity scale in wake shear layer
u_x	Axial velocity component
u'_x	Velocity fluctuations in axial direction
$u_{0,\theta}$	Rotational velocity at the rotor plane
$u_{1,\theta}$	Rotational velocity far downstream
U_∞	Free-stream velocity
u'_θ	Velocity fluctuations in tangential direction
V	Time-averaged radial velocity
v_c	Lateral spatially dependent large-scale turbulent velocity
v_r	Lateral large-scale turbulent velocity in the rotor plane
w_c	Vertical spatially dependent large-scale turbulent velocity
w_r	Vertical large-scale turbulent velocity in the rotor plane
x	Downstream dimension
X_{max}	Axial downstream domain outer bound
X_{min}	Axial upstream domain outer bound
y	Lateral direction
z	Vertical direction

List of Subscripts

i, j	Index Einstein notation
i, j, k	Index axial basis functions
l	Left cell boundary in marching scheme
m, n, p	Index radial basis functions
r	Right cell boundary in marching scheme

List of Figures

List of Tables

1

Introduction

A surge in global electricity demand is expected in the next decades, driven by economic growth, structural changes, and electrification [1]. However, if this increase in demand is met with reliance on traditional fossil fuel-based energy sources, it could aggravate climate change due to increased greenhouse gas (GHG) emissions. This emphasises the immediate necessity for the widespread adoption of renewable energy sources on a large scale. Europe has acknowledged this problem and has set a target to reach a renewable energy contribution of 27% by 2030 [2, 3]. Wind turbines seem to be an attractive solution to reach the set targets by Europe, as wind power is deemed to be one of the most cost-effective ways to produce renewable energy [4]. This is further confirmed by the initiatives taken by Europe to install 116 GW of new wind farms over the period from 2022-2026 [5].

Wind turbines have been used to harvest power from the wind for decades. However, they cause a reduction in flow velocity and an increase in turbulence intensity in their wakes, known as wake effects [6]. When wind turbines are subjected to these wake effects, they may lose up to 20% or 30% [7] of their power, resulting in a reduction in the efficiency of wind farms. Moreover, wake effects also increase fatigue loads, negatively affecting the lifetime of the wind turbine subjected to these effects [8]. As an increasing number of wind farms are being established, to maximise the exploitation of wind energy potential within a specific area, supporting the energy transition, the emphasis on cost-efficiency serves as a strong motivator for research in wake aerodynamics. This research is geared toward mitigating the current impediments that restrict the clustering of wind farms [9].

For a considerable period, particularly before 2010, wind turbines in wind farms operated independently. The primary focus was on optimising individual power output, without considering the influence of aerodynamic interactions on nearby turbines. However, by properly coordinating wind turbines, it is possible to achieve enhanced power output and reduced overall structural loads [10, 11]. Several wake mitigation methods are currently being developed. To evaluate these methods effectively, a wake model with low computational cost is essential, especially in industry. However, a reduction in computational cost comes at the expense of diminished model fidelity. While low-fidelity models may suffice for less intricate aspects of wind farm design, such as annual energy production, higher-fidelity models become indispensable for accurately assessing fatigue and aeroelastic loading in downstream wind turbines [10, 9].

Driven by the necessity for more sophisticated models to assess fatigue and aeroelastic loading, the primary aim of this project is to devise a medium-fidelity model distinguished by its optimal computational efficiency. This pursuit emphasises cost-effectiveness while guaranteeing that the model facilitates precise and efficient load calculations. Accordingly, this thesis explores the application of the Finite Element Method (FEM) as the numerical method for simulating wind turbine wakes, as stabilised finite element methods have demonstrated their capability to generate robust and precise numerical solutions for both compressible and incompressible Navier-Stokes equations, covering a broad spectrum of laminar and turbulent flow scenarios [12].

The refinement of load calculations not only contributes to advancing research efforts but also conserves valuable engineering hours during development. A key focus in these calculations is the acquisition of the axial velocity profile of the wake, establishing the foundational requirement for load calculations. Additionally, in pursuit of practical and realistic wind farm-level load calculations, the implementation focus is set on maintaining low computational costs [9]. Moreover, even though the model is to be used for load calculations at the wind farm level, the emphasis in this thesis is placed on modelling the wake of an individual turbine.

1.1. Research Questions

The goal of this thesis is to develop a medium-fidelity wake model designed for performing wind turbine load calculations, that achieves a balance between computational efficiency and accuracy. Furthermore, the research objective must align with the requirements derived from the needs outlined in the introduction. In particular, the wake model's output should include the essential inputs for conducting load calculations on wind turbines. Additionally, there is an expectation that the FEM demonstrates improved computational efficiency and stability compared to alternative numerical methods, such as spectral methods previously studied. Therefore, the principal research objective can be formulated as:

> *This thesis aims to establish the groundwork for a computationally efficient wake model by solving the axisymmetric and stationary Navier-Stokes equations through the implementation of a finite element method providing the necessary inflow profile to perform wind turbine load calculations.*

Derived from this research objective, the main focus of the research is centered on evaluating the suitability and effectiveness of the finite element method for axisymmetric wind turbine wake modelling. Consequently, two principal research questions can be articulated, each further segmented into sub-questions for comprehensive exploration.

1. **Can the FEM be considered a suitable numerical approach for axisymmetric wind turbine wake modelling?**
    1.1. How does the accuracy of the FEM compare to alternative numerical approaches for axisymmetric wind turbine wake simulation, in particular, compared to spectral methods?
    1.2. What are the specific challenges or limitations associated with implementing the FEM in the axisymmetric wake modelling process, and how can they be addressed?
2. **What is the level of effectiveness exhibited by the FEM in meeting the demands of industrial applications?**
    2.1. Does the computational cost of the wake model using the FEM conform to the established norms within the wind industry?
    2.2. To what extent does the wake model, utilising the FEM, enhance the accuracy in comparison to well-established engineering models currently in use?
    2.3. Is the FEM wake model sufficiently robust and stable to address the specific demands of the wind industry?

1.2. Thesis Outline

The structure of this report is outlined as follows. Chapter 2 provides an extensive review of state-of-the-art models for wind turbine wake modelling. Additionally, it discusses the different numerical methods appropriate for solving the governing equations of the described wake models. Subsequently, Chapter 3 focuses on deriving the two-dimensional Ainslie wake model using the finite element method, while Chapter 4, concerns itself with verifying the method. This is followed by Chapter 5 which discusses the extension of the Ainslie wake model with a forcing term. Lastly, Chapter 6 concludes the study and provides recommendations for future work.

2

State-of-the-art

2.1. Wind Turbine Models

To properly capture the dynamics described by the Reynolds Averaged Navier-Stokes (RANS) equation or Large Eddy Simulation (LES), it becomes imperative to incorporate an accurate representation of wind turbine blades. This is particularly essential as the behaviour of the wake is intricately linked to the forces exerted by these blades. Therefore, a comprehensive understanding of wake characteristics hinges on an effective representation of the wind turbine blades in the simulation. This is crucial for comprehending their influence on the wake. Therefore, this section explores two main approaches for modelling wind turbine blades. The initial approach, discussed in Section 2.1.1, portrays the wind turbine blades through the utilisation of body forces. In contrast, the second approach, discussed in Section 2.1.2, is a direct method that incorporates the blades by discretising the actual blade geometry on a computational mesh [13].

2.1.1. Body Force Representation

To circumvent the direct computation of blade boundary layers, thereby reducing computational costs and simplifying the mesh generation, the rotor can be represented and modelled by applying force to the flow. This force can be explicitly incorporated into the weak form momentum equation, as presented in Eq. (2.1). It is essential to note that the weak form is employed due to the force inducing a discontinuity in the pressure [13].

$$\int_{\Omega} \frac{\partial \vec{u}}{\partial t}\,\mathrm{d}\Omega + \int_{\partial\Omega} \vec{u}\,\vec{u}\cdot\vec{n}\,\mathrm{d}S = -\int_{\partial\Omega} \frac{1}{\rho} p\,\vec{n}\,\mathrm{d}S + \int_{d\Omega} \nu\nabla\vec{u}\cdot\vec{n}\,\mathrm{d}S + \int_{A\cap\Omega} \vec{f}\,\mathrm{d}A \tag{2.1}$$

2.1.1.1. Actuator Disk Model

The actuator disk model (ADM), as depicted in Fig. 2.1, offers the most simplified representation of a wind turbine, eliminating the requirement to resolve the intricate geometry of turbine blades. Instead, it models the rotor as a permeable surface normal to the free-stream velocity, which is subjected to the influence of surface forces [14, 15, 16]. To replicate the force distribution over an actual blade, these forces are distributed across annular rings, commonly subdivided into azimuthal segments to accommodate variations in azimuth. The mathematical intricacies of this method will not be delved into here, as they are deemed beyond the scope of this work.

It has been demonstrated that ADM can effectively predict the development of the far wake and power production [17]. However, it appears to overlook the vortical structures near the rotor, as it cannot account for the root and tip vortices generated by individual blades. Consequently, the ADM may not accurately represent these vortical structures near the rotor, underlining the importance of exploring alternative rotor representations for investigating this flow region [18]. Furthermore, limitations related to loading are also apparent. In scenarios characterised by high turbulence levels or partial wake operation, resolving all dynamic effects may pose challenges, making actuator line methods more appropriate.

2.1.1.2. Actuator Line

The actuator line model (ALM), as depicted in Fig. 2.1, has become a prominent tool to represent lifting surfaces, such as wind turbine blades, without having to resolve the entire flow field near the blades [19].

In the ALM, the rotating blades are represented by discrete actuator points, as presented in Fig. 2.1, each with lift and drag forces determined from local flow conditions and airfoil data, resulting in an aerodynamic force vector for each actuator point. The calculated forces are then projected onto the Cartesian grid as body forces in the Navier-Stokes equations [20].

However, directly applying this force as a body force to the Navier-Stokes equations of the underlying flow solver could result in nonphysical numerical oscillations. To prevent numerical issues, the point force vectors can be transformed into grid force vectors using a smearing kernel, such as a Gaussian kernel. This method ensures a more realistic representation of aerodynamic effects, preventing unwanted oscillations in the simulation [20].

In comparison to computational approaches, the ALM distinguishes itself through its simpler mesh requirements, which are notably less complex than those needed for blade-resolved simulations. Additionally, the ALM is characterised by a reduced computational cost [21]. Moreover, when compared to the ADM, the ALM offers a more accurate representation of physics in the near wake, as it can capture root and tip vortices, which the ADM cannot. The ALM is also capable of depicting helical and asymmetric structures across the wake, leading to an asymmetric roll-up of Kelvin-Helmholtz-like instabilities in the wake's outer shear layer. Hence, when focusing on the near wake, the preference leans toward the ALM over the ADM. However, when examining the far wake, the ADM yields satisfactory results while offering computational cost savings [18]. Furthermore, the ALM holds the advantage that, with the application of a correction for force smearing, a tip correction is no longer necessary. In contrast, the ADM still necessitates a tip correction, posing challenges in its implementation within a CFD setting.

2.1.1.3. Actuator Surface

Shen et al. [22] extended the actuator line method to an actuator surface (AS) method, as illustrated in Fig. 2.1. In this modified approach, the wind turbine blade is portrayed as a planar surface defined by both the blade length and chord [13]. This planar surface is then superimposed onto the mesh, encompassing an area equivalent to the blade element over a single time step. As a result, the force distribution closely resembles that achieved in a simulation with fully resolved blades, enhancing the management of near-wake vortices [23]. While this method has been demonstrated to represent the flow field more realistically than the ALM approach, it currently lacks the capability to model the wake of an airfoil due to the absence of accounting for shear forces on the blade. Additionally, this method demands more precise airfoil data; instead of requiring information solely about the lift and drag coefficients, it necessitates knowledge of the pressure and skin-friction distribution on the airfoil surface [13].

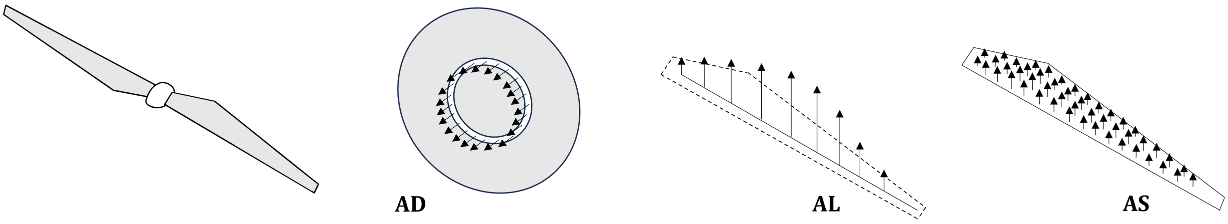


Figure 2.1: Illustration of actuator disk (AD), actuator line (AL), and actuator surface (AS) rotor representations [13].

2.1.1.4. Conclusion

A hierarchy of actuator models has emerged, spanning from actuator disks to more refined actuator line and surface models. Striving for greater accuracy entails higher computational costs and the need for more detailed airfoil data. It would appear that the unsteady nature of actuator line and surface methods makes them more suitable for LES simulations, while the steady characteristics of actuator disk methods tend to confine their application to RANS simulations. However, computational limitations primarily confine the application of actuator line methods in LES simulations to studies focused on single wakes. Consequently, in wind farm simulations, actuator disks are predominantly chosen for LES, indicating that they can also handle unsteady conditions [13]. Therefore, the choice of rotor representation is determined not by the steady or unsteady nature of the simulation, but rather by computational costs and the specific aspect of the wake being studied.

2.1.2. Direct Modelling

Wind turbine blades can be resolved using direct modelling based on Computational Fluid Dynamics (CFD) simulations. While this approach provides the most precise depiction of the blades and the boundary layer, it is also associated with significant computational expenses and challenges in creating a high-quality mesh.

2.2. Wake Models

Wind turbine wakes play a crucial role when it comes to maximising the power output of wind farms, as they reduce the power production and increase the loading experienced by downstream wind turbines. Therefore, to predict wind farm performance, understand the loads wind turbines are subjected to and evaluate several wake mitigation methods, there is a continuous need for wake models [24]. Three main wake analysis methods can be distinguished.

First, wake effects can be analysed using analytical wake models. The main advantage of analytical models is that the obtained solutions are exact and provide fundamental insight into the available resources. This, however, comes with the drawback that these analytical models are obtained under strong assumptions and are therefore only applicable to specific problems [25, 26]. Nevertheless, the simplicity and low computational cost of these models make them attractive for wind farm layout optimisation and wind farm control, especially in the context of very large problems, such as wind farm clusters with several hundred turbines. Nowadays, there is also a growing interest in running time series simulations, further emphasising the practical appeal of these models for optimisation techniques that necessitate the simulation of thousands of cases [26].

Secondly, wake effects can also be analysed using computational fluid dynamics (CFD), which is directly based on the Navier-Stokes equations. Various numerical models exist, which, in terms of their (dis)advantages, are complementary. More specifically, increasing the fidelity of the numerical analysis resolves smaller turbulent scales, but comes with the drawback of significantly higher computational costs. Advantages of CFD methods include easy simulation of specific physical conditions, extraction of physical information in the whole flow field, and simulation of problems with complex geometries [25, 26]. However, these advantages come with the drawbacks of numerical errors, discretisation errors, and convergence problems. Another drawback of CFD models emerges due to the multitude of unknown inflow parameters, necessitating tuning. These parameters significantly impact the solution outcome, leading to a notable level of uncertainty. In contrast, analytical models only have a few tuning parameters, making these models less sensitive to the input.

Lastly, wake effects can be researched through field or wind tunnel experiments. Field tests offer a more comprehensive understanding of wake effects in real-world conditions, but they also come with their share of challenges, including potential measurement errors, uncontrollable external influences, and the need for substantial investments in terms of time and resources. Wind tunnel experiments on the other hand might result in a relative reduction of time and resource investment, and make it easier to control external parameters. However, wind tunnel experiments come with the extra challenge of scaling the wind turbine or wind farm. Additionally, both field and wind tunnel experiments are necessary to provide useful data for the validation of numerical models [25, 26].

Before introducing various wake models, it should be noted that within wake modelling, it is common to divide the wake development into two distinct regions: the near-wake and the far-wake. Generally, the near-wake allows for the differentiation of properties from the upstream rotor. In contrast, the far wake is predominantly influenced by the power properties of the upstream wind turbine [27, 9].

2.2.1. Momentum Method

Axial momentum theory, in which the rotor is modelled as an actuator disk, offers a simple method to evaluate the velocity and pressure in the wake. It assumes inviscid, incompressible, axisymmetric, and stationary flow, and neglects the effects of external forces. Moreover, an ideal disk is assumed, meaning there are no friction forces, and there is no rotational velocity component in the wake [16, 14, 28].

With the applied assumptions, the momentum equation and Bernoulli's principle can be applied to the actuator disk, resulting in the streamlines, axial velocity, and pressure distributions presented in Fig. 2.2. This representation shows that when the flow is approaching the rotor, the axial velocity decreases gradually from the free-stream velocity U_∞ to an average axial velocity of u_0 at the rotor plane and further to an average axial velocity of u_1 far downstream in the wake, which is depicted by the axial induction factor a, as shown in Eq. (2.2). Consequently, the pressure increases following Bernoulli, except at the rotor plane where there is a pressure loss. This is due to energy being extracted from the flow by the applied force field at the rotor plane. Furthermore, the streamlines show an expansion moving downstream, resulting from the conservation of continuity [16].

$$u_0 = (1-a)U_\infty \qquad \text{and} \qquad u_1 = (1-2a)U_\infty \tag{2.2}$$

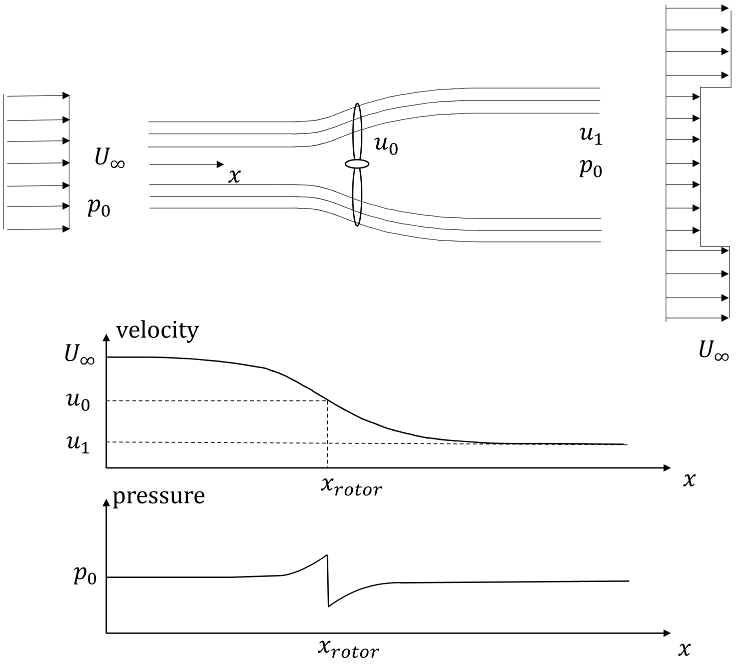


Figure 2.2: Representation of streamlines past the rotor disk and axial velocity and pressure up and downstream of the rotor [16].

Expanding the axial momentum theory to account for wake rotation entails introducing an expression for angular momentum balance. This involves equating changes in angular momentum to the torque exerted by the rotor on the air, introducing an azimuthal induction factor a'. The rotational velocity at the rotor plane and in the far wake are defined in Eq. (2.3) [29, 28, 16].

$$u_{0,\theta} = (1-a')\omega r \qquad \text{and} \qquad u_{1,\theta} = (1-2a')\omega r \tag{2.3}$$

In case of a non-uniform inflow, the momentum and energy relation can be applied on an annular level. This facilitates the alignment of momentum analysis results with the blade element properties and geometry specific to the annulus [28]. However, this approach assumes no radial dependency, indicating no interaction between the annuli. Additionally, the force exerted by the blades on the flow remains constant within each annular element, resembling a rotor with an infinite number of blades. These assumptions however introduce errors [16]. To rectify errors resulting from these assumptions, the momentum method requires some corrections to be applied.

The first correction is called Prandtl's tip loss factor. This correction is essential to account for the assumption of an infinite number of blades, as the wake vortex system for a turbine with an infinite number of blades deviates from that of a turbine with a limited number of blades [16]. However, this tip loss model assumes that the tip loss only affects the induction factor, not the mass flux [30, 9]. Wilson and Lissaman [31] introduced a refined tip loss model, suggesting that the mass flow through the rotor disc could be corrected similarly to the induced velocity in the wake. However, this tip loss model did not satisfy the orthogonality of the induced velocity to the relative velocity at the blade element [30]. Shen et al. [30]

later proposed an improved tip loss model, including a correction for the mass flux. Comparisons between experimental data and various correction models have shown that the proposed tip loss model by Shen et al. [30] better predicts the aerodynamic force distribution in the vicinity of the tip.

Secondly, the momentum method breaks down for high rotor loading. Glauert [29] proposed a correction to the thrust coefficient to account for the fact that when the axial induction factor becomes larger than about 0.4, the basic momentum theory is no longer valid [29, 16, 32, 28].

Lastly, the momentum method assumes that the radial flow component is negligibly small compared to the axial and azimuthal flow components. However, since the wake expansion is dependent on the radial velocity component, this introduces another source of error [33]. Micallef et al. [33] proposed a three-dimensional potential flow model to account for these effects.

2.2.2. Engineering Wake Models

Numerous engineering wake models have been developed over time, as indicated in studies such as [34, 26, 27]. These models are frequently fine-tuned based on both experimental data and numerical simulations, playing a crucial role in efficiently calculating wake deficit for wind farm optimisations. Unlike the intricate and time-consuming computations associated with vortex-based and numerical wake models, engineering wake models offer a more favourable alternative. Researchers have actively worked on advancing these analytical wake models, aiming to deliver swift and accurate assessments of wake deficit, thereby enhancing their practicality in studies related to wind farm optimisation [35]. Before delving into the review of some prominent engineering wake models, it is crucial to recognise that, at a fundamental level, all these models share a common structure. They typically exhibit radial and streamwise dependencies, relying on an expansion factor and the shape of the radial profile.

The Jensen wake model, based on studies by Jensen et al. [36] and Katic et al. [37] offers one of the simplest representations of the wake based on global conservation of momentum and is widely adopted in commercial applications for wind farm optimisation. However, this wake model assumes a top-hat wake velocity profile, which is simply related to the downstream distance, such that the wake velocity distribution in the wake influence plane is constant at a certain distance. Due to this assumption, the model is limited to making predictions solely about the far wake velocity deficit. Nevertheless, the primary objective of this model is not to achieve an accurate velocity profile but rather to estimate the energy content perceived by downstream wind turbines, as highlighted in [34, 24]. It was demonstrated that the Jensen wake model can reasonably predict the maximum velocity deficit in certain regions further downstream. Still, it tends to underestimate the maximum velocity deficit in other regions when compared to LES and experimental data [38]. The Frandsen model, additionally to the Jensen model applied balance of momentum, as presented in [39], shares a similar assumption regarding the wake velocity profile, leading to comparable issues [35]. Top-hat wake models, such as the Jensen and Frandsen wake models, as expected, generally underestimate the velocity deficit at the centre of the wake and overestimate it near the edge of the wake [38].

Subsequent wind tunnel tests unveiled a self-similar Gaussian wake loss distribution, inspiring the development of an enhanced wake model. Bastankhah and Porté-Agel introduced a novel wake deficit model in their work [38], employing a Gaussian velocity profile within the wake and applying mass and momentum conservation. This proposed wake model has demonstrated consistency and acceptable accuracy in power estimation. In contrast, top-hat models like the Jensen and Fransen wake models exhibit lower accuracy and heightened sensitivity to the relative position of turbines concerning the wind direction [38]. Nevertheless, this model still exhibits concerns regarding its robustness and universality [40].

An improvement to the Bastankhah and Porté-Agel [38] Gaussian wake model was proposed by Ishihara and Qian [40] by taking into account the inflow turbulence intensity and thrust coefficient. Utilising a substantial quantity of large-eddy (LES) simulations and incorporating data from prior simulations, they calibrated the parameters in the existing Gaussian model and rectified the maximum wake velocity deficit. Consequently, they proposed a Gaussian model for both the near- and far-wake regions. Numerical simulations closely align with wind tunnel tests, revealing similar profiles for mean velocity and turbulence intensity behind both the model and utility-scale wind turbines. This suggests that the thrust coefficient and ambient turbulence intensity are key parameters influencing wake flow, regardless of the specific wind turbine type [40]. However, the Ishihara and Qian wake model results are still not ideal, which can result in inaccurate results in wind farm design [35].

Each engineering model outlined here provides insights into the wake deficit of an individual wind turbine. Through the superposition of these individual turbine wake deficits, the cumulative impact of multiple wakes can be examined. Widely adopted in the wind energy community for predicting power production in wind farms, this approach allows for a comprehensive analysis of the collective influence on entire wind farms. Such a thorough assessment is crucial for optimising wind farm layouts, mitigating interference effects, and enhancing overall energy extraction efficiency.

However, a significant challenge with the majority of existing wake superposition methods is their failure to conserve streamwise momentum [41]. Hence, Zong et al. introduced an innovative wake superposition method, as detailed in [41], capable of conserving the total momentum deficit in the streamwise direction. Thorough comparisons with existing methods demonstrate that this novel approach outperforms others, providing accurate predictions of power production and centerline wake velocity deficit, with a typical error of less than 5% (excluding the near-wake region). Furthermore, the momentum-conserving wake superposition method is expanded to incorporate transverse velocities induced by yawed wind turbines, effectively reproducing the secondary wake steering effect crucial for power optimisation in active wake control.

Furthermore, the general approach adopted by the IEC 61400-1 standard [42] will be explored, which relies on the inflow wind speed. In instances where the inflow wind speed falls below the rated wind speed, the resulting wake contribution is determined by selecting the minimum from individual wake contributions, as outlined in Eq. (2.4). On the contrary, when the inflow wind speed surpasses the rated wind speed, the wake is regarded as the sum of all local wake contributions, as defined in Eq. (2.5). This nuanced approach mirrors the standard's strategy in effectively managing wind turbine wakes across diverse wind speed conditions.

$$\vec{u}(\vec{x},t) = \min_i \left(\vec{u}_i(\vec{x},t)\right) \tag{2.4}$$

$$\vec{u}(\vec{x},t) = \sum_i \vec{u}_i(\vec{x},t) \tag{2.5}$$

2.2.3. Vortex Wake Models

Vortex wake models belong to a category of method where the rotor blades, along with the trailing and shed vortices in the wake, are represented using lifting lines or surfaces. Compared to engineering wake models or the momentum method, vortex wake models have the advantage that they are able to describe multiple aerodynamic phenomena as they include some physical aerodynamic elements, which are discarded in momentum-based methods. This however comes at the expense of a higher computational cost.

Various vorticity elements around the rotor can be distinguished. The vorticity trailing the turbine is referred to as wake vorticity, interacting with turbulence and shear vorticity in the atmosphere. Close to the turbine, iso-vorticity contours encircle each blade, indicating bound vorticity. At the blade tip, vorticity follows a helical path in the wake, known as tip vorticity (or tip-vortex), and similar helical patterns are observed at the root, termed root vorticity (or root-vortex). Both tip and root vorticity contribute to the overall wake vorticity. Vorticity is typically concentrated and more pronounced at these locations. The remaining wake vorticity is emitted along the blades, also tracing a helical pattern behind the rotor [43].

To simplify matters, the assumption is made that the emitted vorticity is confined to a narrow layer originating from the body's trailing edge, simplified as a vorticity sheet. This simplification is considered appropriate for high-Reynolds attached flows, and thus applicable for wind turbines in general, and therefore applicable for wind turbine wakes in general. The vorticity generated in the wake due to temporal changes in the bound circulation is known as shed vorticity, Γ_s. The conservation of circulation dictates that the circulation Γ_s around this shed vorticity satisfies [43, 44, 16, 6]:

$$\Gamma_s(r) = \frac{\partial \Gamma_b(r)}{\partial t} \mathrm{d}t \tag{2.6}$$

The spanwise variation in bound vorticity generates vorticity that is released into the wake, known as trailed vorticity. This is by Helmholtz's theorem. The circulation encircling the trailed vorticity, represented as Γ_t, equals the gradient of the bound circulation [43, 44, 16, 6]:

$$\Gamma_t(r) = -\frac{\partial \Gamma_b(r)}{\partial r}\mathrm{d}r \tag{2.7}$$

Knowing the circulation, the global flow field is determined from the Biot-Savart law, where the vortex filaments in the wake are convected by the superposition of the undisturbed flow and the induced velocity field. Assuming that the flow in the region outside the trailing and shed vortices is curl-free, the overall flow field can be described by evaluating the Biot-Savart law [44, 16, 6].

In vortex models, the structure of the wake can be predetermined or computed as part of the overall solution procedure. In a prescribed vortex technique, vortical element positions are determined based on measurements or semi-empirical rules, enabling fast computation but limiting its applicability to well-understood steady flow situations. For unsteady flow and intricate wake structures, a free wake analysis is necessary. The free wake method allows vortex elements to freely convect and deform under the influence of the velocity field, offering the advantage of handling diverse flow scenarios, including yawed wake structures and dynamic inflow. However, this method is computationally more demanding than the prescribed wake method, since the Biot-Savart law needs evaluation at each time step. Additionally, free-wake vortex methods may face stability challenges due to the inherent singularity in induced velocities as vortex elements approach each other. This can be partially mitigated by introducing a vortex core model with a cut-off parameter representing the inner viscous part of the vortex filament. A variant of free vortex wake methods is Voutsinas' method [45], where wake modelling is managed through the use of vortex particles [6].

2.2.4. Numerical Wake Models

Numerical wake models can be classified based on their level of fidelity. As fidelity increases, more physical details are captured, but this comes at the expense of higher computational costs.

2.2.4.1. Governing Equations

Numerical wake models utilise the incompressible Navier-Stokes equations, as detailed in Equation 2.8. The assumption of incompressibility is applicable when the local Mach number is below 0.3, corresponding to a flow velocity of around 100 m/s [46]. According to Snel [47], this assumption is thoroughly justified. However, Pijl [48] points out the possibility of compressibility effects at the blade tips. This is because, at a tip speed of 75 m/s, the maximum speed on the suction side could exceed 100 m/s. Despite this concern, the incompressibility assumption remains valid due to significantly lower velocities in the wake, as emphasised by Sanderse [46].

$$\begin{aligned} \frac{\partial u_i}{\partial x_i} &= 0 \\ \frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} &= -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} \end{aligned} \tag{2.8}$$

Through spatial and temporal discretisation, the numerical implementation of the Navier-Stokes equations becomes possible. However, the vast range of spatial and temporal scales influencing the wake of a wind turbine makes a direct numerical analysis impractical due to the associated computational costs. Therefore, assumptions and/or turbulence models need to be implemented to address and overcome this challenge [25].

2.2.4.2. Large Eddy Simulation

Considering computational costs, Large Eddy Simulation (LES) stands out as the model with the highest fidelity that remains practical in terms of computational time for a substantial number of flow cases. Despite the potential for Direct Numerical Simulations (DNS) to provide more precise results, the associated computational expenses make it impractical for many scenarios. Consequently, DNS will not be discussed as it does not align with the requirement for low computational costs.

In LES, the computation focuses on directly calculating the large-scale motions (large eddies), while only modelling the small-scale motions (sub-grid scale (SGS)). This leads to a substantial reduction in computational costs compared to DNS. Furthermore, LES surpasses the accuracy of the RANS approach because it directly captures the large eddies, which contain the majority of turbulent energy and contribute significantly to momentum transfer and turbulent mixing. In contrast, the RANS approach relies on modelling these eddies and can therefore only capture a single, pre-determined scale. Additionally, the smaller scales in LES tend to exhibit more isotropic and homogeneous characteristics, making the modelling of SGS motions more manageable than attempting to model all scales within a single framework, as done in the RANS approach [49].

In LES, spatial filtering is used to decompose the velocity field into large-scale motions and small-scale motions. Applying a spatial low-pass filter to the Navier-Stokes equations results in Eq. (2.9). The SGS stress tensor, which appears from applying the spatial filter, represents the effect of the small (unresolved) scales on the large scales and requires a separate SGS model. Various SGS models exist, which are discussed by Meneveau and Katz [50].

$$\underbrace{\frac{\partial \tilde{u}_i}{\partial t}}_{\text{Acceleration term}} + \underbrace{\frac{\partial \tilde{u}_i \tilde{u}_j}{\partial x_j}}_{\text{Advection term}} = -\underbrace{\frac{1}{\rho}\frac{\partial \tilde{p}}{\partial x_i}}_{\text{Pressure term}} + \underbrace{2\nu\frac{\partial \tilde{S}_{ij}}{\partial x_j}}_{\text{Viscous term}} - \underbrace{\frac{1}{\rho}\frac{\partial \tau_{ij}}{\partial x_j}}_{\text{Sub-filter scale stress tensor}} \tag{2.9}$$

2.2.4.3. Dynamic Wake Meandering Model

The dynamic wake meandering (DWM) model developed by Larsen et al. [51] at the Technical University of Denmark (DTU) is based on the fundamental concept that the wake behind a wind turbine in the atmospheric boundary layer behaves similarly to a passive tracer, influenced by the large-scale turbulence structures in the atmospheric boundary layer. This concept underlying the DWM model has been demonstrated to closely align with full-scale experimental observations [51].

The DWM model's structure can be broken down into three essential cornerstones, as depicted in Fig. 2.3 [52]. This involves modelling the quasi-steady wake deficit and its development downstream, the downstream wake-meandering processes, and the turbulence effects induced by the turbine rotor. The quasi-steady wake deficit accounts for the wake deficit formulated in the moving (meandering) frame of reference and considers the wake expansion in response to downstream transportation time, influenced by turbulence diffusion and the rotor pressure field. The wake meandering model outlines the stochastic downstream displacement of wakes emitted from upstream, guided by large-scale turbulence structures. It is assumed that the presence of a wind turbine has minimal impact on these structures. Lastly, the turbulence effects initiated by the upstream turbine rotor involve small-scale turbulence with eddy sizes up to about one rotor diameter. These effects include contributions from conventionally generated turbulence due to wake shear, as well as from the vorticity bound to the blades, primarily composed of tip and root vortices [51].

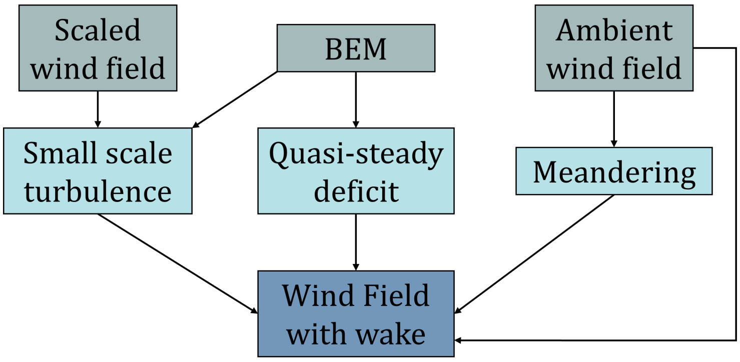


Figure 2.3: Overview of the key components constituting the DWM model [53].

Modelling Wake Deficit

In the initial stages of implementing the DWM model, deficits were calculated using a CFD-based actuator disc model coupled with an aeroelastic rotor model [51]. Yet, for the complete integration of deficit prediction within an aeroelastic code framework, a redefined approach has been undertaken for quasi-steady wake deficit prediction. This revised formulation operates under the assumption that the velocity deficit can be determined by considering inductions in both the rotor plane and the far wake. It employs Blade Element Momentum (BEM) theory in conjunction with the boundary layer approximation of the Navier-Stokes equations to model the downstream progression of the deficit [52]. This model is strongly inspired by the work of J. F. Ainslie [54], who has previously formulated a comparable wake model.

The wake model comprises three fundamental steps. Firstly, the initial wake deficit is established by computing induced velocities in the rotor plane. Subsequently, the expansion of this initial wake deficit in the nearby wake region is determined, which is predominantly influenced by pressure recovery in that particular area. Lastly, the evolution, including expansion and attenuation, of the velocity deficit as a function of the downstream distance from the turbine generating the wake is measured. This analysis considers turbulent mixing influenced by both ambient turbulence and the turbulence generated by the wake shear field. The model relies on a thin shear layer (TSL) approximation of the Navier-Stokes equations in their rotationally symmetric form [52].

Wake Meandering

The foundation of the wake meandering model rests on the core assumption that the transportation of the wake within the atmospheric boundary layer can be effectively represented by treating the wakes as passive tracers, influenced by the large-scale turbulence structures in both lateral and vertical directions. The process of modelling the meandering process requires accurately characterising the stochastic transport medium and defining a suitable cut-off frequency to identify large-scale turbulence structures. Typically, meandering is determined using the Mann synthetic turbulence model to generate a low-pass filtered ambient wind field. This approach aligns with conventional practices for quantifying turbulence loads. The filter cut-off frequency, denoted as f_c, is defined by mean free wind speed, U_∞, and the rotor diameter, D, as outlined in Eq. (2.10) [51, 55].

$$f_c = \frac{U_\infty}{2D} \tag{2.10}$$

The DWM model represents the wake as a sequence of deficits emitted at consecutive time instants, following a passive tracer analogy. By incorporating Taylor's hypothesis, the downstream advection of this sequence of wake deficits is presumed to be governed by the mean wind speed of the ambient wind field. This assumption enables the straightforward separation of the wake along the wind deficit profile and its expansion from the transportation process [51, 55].

In the lateral and vertical directions, each wake cascade element is displaced based on large-scale turbulence velocities at its specific position and time. This implicitly neglects any potential mutual interaction between wake momentum and displacement in these directions, justified by the minimal impact of lateral and vertical wake momentum compared to along-wind momentum. Mathematically, the wake deficit dynamics in the lateral (y) and vertical (z) directions are described by a system of differential equations [51]. Mathematically, the wake deficit dynamics in the lateral direction (y) and the vertical direction (z) are assumed to be governed by the system of differential equations presented in Eq. (2.11) [51]. Here, v_c and w_c represent the spatially dependent large-scale turbulent velocities, and t_0 denotes the time instant at which the particular cascade element is emitted [51].

$$\frac{\mathrm{d}y(t,t_0)}{\mathrm{d}t} = v_c(y,w,t,t_0) \qquad \frac{\mathrm{d}z(t)}{\mathrm{d}t,t_0} = w_c(y,z,t,t_0) \tag{2.11}$$

Given the predominant influence of large-scale turbulence structures on the meandering process, it is reasonable to approximate lateral and vertical turbulence as spatially homogeneous for each cascade element, especially for small to moderate downstream distances. This implies that the displacement of the wake centre is primarily influenced by the large-scale lateral and vertical velocities over the rotor disc at the release time. In this scenario, the defined system of differential in Eq. (2.11) can be simplified as presented in Eq. (2.12). Here, v_r and w_r represent the large-scale turbulent velocities at the rotor plane [51].

$$\frac{\mathrm{d}y(t,t_0)}{\mathrm{d}t} = v_r(t_0) \qquad \frac{\mathrm{d}z(t)}{\mathrm{d}t} = w_r(t_0) \tag{2.12}$$

Added Wake Turbulence

The turbulence arising from wind turbines in their wake is a complex phenomenon, resulting from a combination of contributions. These include conventional mechanically generated turbulence caused by wake shear and blade-bound vorticity, primarily originating from tip and root vortices [51, 52, 55]. In the beginning, these vortices showcase organised and coherent flow structures. Nevertheless, as time progresses, they experience a gradual breakdown and start to manifest characteristics akin to conventional turbulence. However, these attributes are modified compared to atmospheric turbulence [52].

At a designated downstream position, the modelling of wake turbulence is conducted by employing a homogeneous Mann turbulence field with cross-sectional coverage extending over one rotor diameter. The turbulence box exhibits a fine resolution, consisting of 128x128 points in planes perpendicular to the mean flow direction, enabling the effective resolution of small turbulence eddies. Its extension along the mean flow direction spans 4800 m with a resolution of 1024 points. The added turbulence is anchored in the meandering frame, synchronising the centre position of the turbulence box with that of the corresponding wake deficits. Despite deviating from the second-order statistics of the turbulence field, the inhomogeneity of the added turbulence is estimated through scaling of the created homogeneous Mann field. Assuming rotational symmetry of the added wake turbulence intensity, the scaling coefficient is solely dependent on the radial rotor coordinate, the wake deficit depth and the wake deficit radial gradient [52, 55].

2.2.4.4. Reynolds Averaged Navier-Stokes Equation

The Navier-Stokes equation is commonly assessed in a time-averaged manner, leading to the formulation of the Reynolds-Averaged Navier-Stokes (RANS) equation. The derivation of the RANS equation involves applying the Reynolds decomposition concept, which involves separating the velocity into its time-averaged and fluctuating components, as presented in Eq. (2.13). Incorporating the Reynolds decomposition into the incompressible Navier-Stokes equation and taking the time-average results in Eq. (2.14).

$$u(\vec{x},t) = \overline{u}(\vec{x}) + u'(\vec{x},t) \tag{2.13}$$

$$\underbrace{\overline{u}_j \frac{\partial \overline{u}_i}{\partial x_j}}_{\text{Advection term}} = - \underbrace{\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_i}}_{\text{Pressure term}} + \underbrace{2\nu \frac{\partial \overline{S}_{ij}}{\partial x_j}}_{\text{Viscous term}} - \underbrace{\frac{\partial \overline{u_i' u_j'}}{\partial x_i}}_{\text{Reynolds stress}} \tag{2.14}$$

The Reynolds stress tensor $\overline{u_i' u_j'}$, which represents the momentum transfer caused by turbulent fluctuations, leads to an indeterminate system by introducing additional unknown variables. This problem, commonly known as the closure problem, requires additional modelling to close the RANS equation, and has resulted in the creation of various turbulence models.

Linear eddy viscosity models belong to the category of turbulence models, representing the Reynolds stress as the product of eddy viscosity and mean strain rate (expressed in Equation Eq. (2.15)), known as the Boussinesq hypothesis. However, the eddy viscosity also requires to be modelled. This can be done using algebraic models, or turbulence-energy-equation models. The latter are commonly classified by the number of equations employed to close the set of equations.

$$-\overline{u_i' u_j'} = \nu_t \left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i} \right) - \frac{2}{3} k \delta_{ij} \tag{2.15}$$

RANS has proven to offer a more precise portrayal of the time-averaged velocity field in the wake than lower fidelity models, balancing this accuracy with a reduction in computational costs compared to higher fidelity models. However, its drawback lies in the incapacity to capture dynamic phenomena due to time-averaging. More specifically, with RANS only isotropic turbulence can be captured, however, atmospheric turbulence is anisotropic. Furthermore, turbulence models often rely on parameters that necessitate tuning based on higher fidelity simulations or experiments. Consequently, there is an opportunity to enhance efficiency by further simplifying the RANS equations, thereby minimising the required level of modelling.

2.2.4.5. Ainslie Eddy Viscosity Wake Model

The Ainslie wake model, representing a field model, will be explored. Other notable field models have been developed by Taylor [56], Liu [57], Crespo [58], and Magnusson [59]. All these models address a simplified version of the Reynolds-averaged Navier-Stokes flow equations to incorporate the conservation of mass, momentum, and energy within the flow. Additionally, they employ an eddy viscosity turbulence model to account for the turbulent mixing contributions from both the shear layer and ambient turbulence [60].

More specifically, the Ainslie wake model originates from simplifying the incompressible RANS equation in cylindrical coordinates (see Eqs. (2.16) to (2.18)), employing the thin shear layer approximation and incorporating the eddy-viscosity closure term for Reynolds stress [54, 61, 9, 60].

The starting point for deriving the Ainslie wake model are the steady incompressible RANS equations in cylindrical coordinates, which are expressed in Eqs. (2.16) to (2.18). This formulation omits the influence of gravity and external forces. In this context, $\bar{u}_x$, $\bar{u}_\theta$, and $\bar{u}_r$ denote the time-averaged velocity components in the streamwise (x), azimuthal (θ), and radial (r) directions, respectively. The variable $\bar{p}$ represents the time-averaged pressure. Additionally, turbulent fluctuations with a zero mean for the velocity components are represented by u'_x, u'_θ, and u'_r [62].

Axial momentum equation

$$\begin{aligned}&\bar{u}_x\frac{\partial \bar{u}_x}{\partial x}+\bar{u}_r\frac{\partial \bar{u}_x}{\partial r}+\frac{\bar{u}_\theta}{r}\frac{\partial \bar{u}_x}{\partial \theta}=-\frac{1}{\rho}\frac{\partial \bar{p}}{\partial x}\\&\quad+\nu\left[\frac{\partial^2\bar{u}_x}{\partial x^2}+\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \bar{u}_x}{\partial r}\right)+\frac{1}{r^2}\frac{\partial^2\bar{u}_x}{\partial \theta^2}\right]-\frac{\partial \overline{u'_x u'_x}}{\partial x}-\frac{1}{r}\frac{\partial(r\overline{u'_x u'_r})}{\partial r}-\frac{1}{r}\frac{\partial \overline{u'_x u'_\theta}}{\partial \theta}\end{aligned}\tag{2.16}$$

Azimuthal momentum equation

$$\begin{aligned}&\bar{u}_x\frac{\partial \bar{u}_\theta}{\partial x}+\bar{u}_r\frac{\partial \bar{u}_\theta}{\partial r}+\frac{\bar{u}_\theta}{r}\frac{\partial \bar{u}_\theta}{\partial \theta}+\frac{\bar{u}_\theta\bar{u}_r}{r}=-\frac{1}{r}\frac{\partial \bar{p}}{\partial \theta}\\&\quad+\nu\left[\frac{\partial^2\bar{u}_\theta}{\partial x^2}+\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \bar{u}_\theta}{\partial r}\right)+\frac{1}{r^2}\frac{\partial^2\bar{u}_\theta}{\partial \theta^2}+\frac{2}{r^2}\frac{\partial \bar{u}_r}{\partial \theta}-\frac{\bar{u}_\theta}{r^2}\right]-\frac{\partial \overline{u'_\theta u'_x}}{\partial x}-\frac{1}{r}\frac{\partial(r\overline{u'_\theta u'_r})}{\partial r}-\frac{1}{r}\frac{\partial \overline{u'_\theta u'_\theta}}{\partial \theta}-\frac{\overline{u'_\theta u'_r}}{r}\end{aligned}\tag{2.17}$$

Radial momentum equation

$$\begin{aligned}&\bar{u}_x\frac{\partial \bar{u}_r}{\partial x}+\bar{u}_r\frac{\partial \bar{u}_r}{\partial r}+\frac{\bar{u}_\theta}{r}\frac{\partial \bar{u}_r}{\partial \theta}-\frac{\bar{u}_\theta^2}{r}=-\frac{1}{\rho}\frac{\partial \bar{p}}{\partial r}\\&\quad+\nu\left[\frac{\partial^2\bar{u}_r}{\partial x^2}+\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \bar{u}_r}{\partial r}\right)+\frac{1}{r^2}\frac{\partial^2\bar{u}_r}{\partial \theta^2}-\frac{2}{r^2}\frac{\partial \bar{u}_\theta}{\partial \theta}-\frac{\bar{u}_r}{r^2}\right]-\frac{\partial \overline{u'_r u'_x}}{\partial x}-\frac{1}{r}\frac{\partial(r\overline{u'_r u'_r})}{\partial r}-\frac{1}{r}\frac{\partial \overline{u'_r u'_\theta}}{\partial \theta}+\frac{\overline{u'_\theta u'_\theta}}{r}\end{aligned}\tag{2.18}$$

Assuming an axisymmetric flow, the azimuthal velocities components and the azimuthal dependencies can be neglected. Consequently, the azimuthal momentum equation, along with all terms depending on θ, can be excluded. This leads to the simplified expressions given in Eqs. (2.19) and (2.20) [54]. Furthermore, this assumption implies that the momentum equation's streamwise and radial velocity components are coupled through the Stokes stream function. Consequently, only a single equation is required to describe this coupling. As a result, further steps will involve exclusively Eq. (2.19).

Axial momentum equation

$$\bar{u}_x\frac{\partial \bar{u}_x}{\partial x}+\bar{u}_r\frac{\partial \bar{u}_x}{\partial r}=-\frac{1}{\rho}\frac{\partial \bar{p}}{\partial x}+\nu\left[\frac{\partial^2\bar{u}_x}{\partial x^2}+\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \bar{u}_x}{\partial r}\right)\right]-\frac{\partial \overline{u'_x u'_x}}{\partial x}-\frac{1}{r}\frac{\partial(r\overline{u'_x u'_r})}{\partial r}\tag{2.19}$$

Radial momentum equation

$$\bar{u}_x\frac{\partial \bar{u}_r}{\partial x}+\bar{u}_r\frac{\partial \bar{u}_r}{\partial r}=-\frac{1}{\rho}\frac{\partial \bar{p}}{\partial r}+\nu\left[\frac{\partial^2\bar{u}_r}{\partial x^2}+\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \bar{u}_r}{\partial r}\right)-\frac{\bar{u}_r}{r^2}\right]-\frac{\partial \overline{u'_r u'_x}}{\partial x}-\frac{1}{r}\frac{\partial(r\overline{u'_r u'_r})}{\partial r}\tag{2.20}$$

By neglecting pressure gradients, assuming that outside the near wake the pressure can be disregarded, the axial momentum equation can be further simplified to Eq. (2.21). This is a significant simplification, rendering the momentum equation invalid in the near wake close to the rotor, as this simplification eliminates the wake expansion. However, Ainslie [54] argues that this assumption can be justified, as the effect of pressure gradients can be accounted for in the eddy viscosity model [54].

$$\bar{u}_x \frac{\partial \bar{u}_x}{\partial x} + \bar{u}_r \frac{\partial \bar{u}_x}{\partial r} = \nu \left[\frac{\partial^2 \bar{u}_x}{\partial x^2} + \frac{1}{r}\frac{\partial}{\partial r}\left(r \frac{\partial \bar{u}_x}{\partial r} \right) \right] - \frac{\partial \overline{u'_x u'_x}}{\partial x} - \frac{1}{r}\frac{\partial (r \overline{u'_x u'_r})}{\partial r} \tag{2.21}$$

Moreover, beyond the initial few diameters downstream, the gradients of mean quantities in the radial direction will be notably greater than those in the axial direction. Furthermore, by excluding diffusion in the axial direction, the problem remains parabolic. Otherwise, it would transition to an elliptic problem, necessitating definition across the entire boundary, which is not feasible. This results in the thin shear layer approximation. Lastly, dropping the viscous terms, the general equation for the Ainslie wake model is given in Eq. (2.22) [54].

$$\bar{u}_x \frac{\partial \bar{u}_x}{\partial x} + \bar{u}_r \frac{\partial \bar{u}_x}{\partial r} = -\frac{1}{r}\partial \frac{r\overline{u'_x u'_r}}{\partial r} \tag{2.22}$$

To describe the Reynolds stress term in Eq. (2.22), the turbulent viscosity concept is used, such that the Reynolds stress and the eddy viscosity can be defined as presented in Eq. (2.23) respectively, with $l_w(x)$ and $u_w(x)$ the suitable length and velocity scales to describe the wake shear layer and ϵ_a the ambient turbulence contribution. As evident from Eq. (2.23), two components characterise the eddy viscosity. The first component delineates turbulent mixing arising from turbulence within the wake's shear layer, contingent on the length and velocity scales defining the shear layer. The second component explicates the influence of ambient atmospheric turbulence on wake mixing [54, 61, 9, 60]. Implementing this formulation for the Reynolds stress term in Eq. (2.22) results in Eq. (2.24).

$$-\overline{u'_x u'_r} = \nu_t \frac{\partial \bar{u}_x}{\partial r} \qquad \text{with} \qquad \nu_t = l_w(x) u_w(x) + \epsilon_a \tag{2.23}$$

$$\bar{u}_x \frac{\partial \bar{u}_x}{\partial x} + \bar{u}_r \frac{\partial \bar{u}_x}{\partial r} = \nu_t \frac{1}{r}\frac{\partial}{\partial r}\left(r \frac{\partial \bar{u}_x}{\partial r} \right) \tag{2.24}$$

In practical terms, a modification is needed for the eddy viscosity equation in the near wake, specifically within about five diameters downstream of the rotor. This modification is essential due to the absence of equilibrium between the mean velocity field and the turbulence field in the near wake region, rendering the relationship in Eq. (2.23) inapplicable [54].

There exists a second equation to model the eddy viscosity, presented by Eq. (2.25), which is adopted in the IEC 61400-1 standard [42] edition 4. Here $\tilde{x} = \frac{x}{D}$ is a non-dimensional coordinate, R_W is the wake radius defined as the radius where the wake is at 95% of the incoming velocity, and F_1 and F_2 are filter functions. Considering the industrial application of the wake model, the preference is for this specific formulation of the eddy viscosity model [9].

$$\frac{2\nu_t}{D u_{r_{hub}}} = 0.023 F_1(\tilde{x}) \left(\frac{u_{x_{rms}}}{u_{r_{hub}}} \right)^3 + 0.016 F_2(\tilde{x}) \frac{R_W(\tilde{x})}{D} \left(1 - \frac{u_{x_{min}}(\tilde{x})}{u_{r_{hub}}} \right) \tag{2.25}$$

The definition of the filter functions can be established by fitting Eq. (2.25) to wind farm measurement data. Larsen et al. [63] conducted this fitting process for the Egmond aan Zee wind farm, yielding the subsequent filter definitions presented in Eqs. (2.26) and (2.27) [9].

$$F_1 = \begin{cases} \left(\frac{\tilde{x}}{8}\right)^{\frac{3}{2}} - \frac{\sin 2\pi \tilde{x}^{\frac{3}{2}}}{2\pi} & \text{if } 0 \leq \tilde{x} < 8 \\ 1 & \text{if } 8 \leq \tilde{x} \end{cases} \tag{2.26}$$

$$F_2 = \begin{cases} 0.0625 & \text{if } 0 \leq \tilde{x} < 4 \\ 0.025\tilde{x} - 0.0375 & \text{if } 4 \leq \tilde{x} < 12 \\ 0.00105(\tilde{x} - 12)^3 + 0.025\tilde{x} - 0.0375 & \text{if } 12 \leq \tilde{x} < 20 \\ 1 & \text{if } 20 \leq \tilde{x} \end{cases} \tag{2.27}$$

The consequences of the assumptions made by the Ainslie wake model remain uncertain. Notably, the assumption of an axisymmetric flow profile poses challenges, as certain flow characteristics introduce errors for this assumption. Firstly, wind turbine wakes inherently display asymmetry due to the presence of atmospheric shear layers. Specifically, the existence of a shear layer causes the incoming wind velocity to consistently feature a non-uniform wind profile, leading to lower wind speeds at the bottom of the rotor compared to the top. Consequently, the shape of the wake in the near-wake region is influenced. However, in the far wake, the wake structure is approximately axisymmetric, self-similar and Gaussian, making the axisymmetric assumption acceptable. However, to consider the asymmetry of the wake profile in the near-wake, an extension of the Ainslie wake model could involve incorporating the azimuthal flow component. Moreover, when attempting to comprehensively represent turbulent structures such as tip vortices, it is generally not recommended to depend on time-averaged flow models, as the average of the fluctuations may amount to zero [9].

2.2.4.6. Linearised Fuga Model

The Fuga model is initially designed as a wake model, serving to evaluate the impact of wake effects on power generation within a cluster of wind turbines. Its application extends to offshore sites and more broadly to homogeneous terrain [64]. Renowned for its robustness, simplicity, and computational efficiency, the FUGA wake model is based on the linearisation of the steady-state RANS equations, incorporating an actuator disk approach [65, 64].

The linearisation of the simplified RANS equations is achieved through Taylor expansion, focusing exclusively on terms of zero and first order. The zeroth-order equations pertain to the scenario where no perturbations are introduced to the flow, signifying the absence of turbines [24]. The linear equations are further simplified through a mixed-spectral formulation, which divides the problem into sets of coupled ordinary differential equations that are independent of each other [64]. Additionally, the model utilises look-up tables to characterise the velocity field behind the wind turbine and linear summation is employed to accommodate multiple wake scenarios. Moreover, to address the challenges that arise when solving a linearised model for flows over small values of roughness length, z_0, particularly relevant for offshore sites with low roughness lengths and more pronounced wakes, a new numerical scheme is implemented which is discussed in detail by Ott et al. [64, 24].

This straightforward wake modelling approach establishes Fuga as one of the most resilient CFD-based models for calculating wake effects [24]. Notably, its results, particularly regarding wind-speed deficits, closely align with those obtained from a nonlinear solution of the RANS equations [65, 64].

As a cost-effective linearised CFD alternative, the Fuga model accurately replicates the behaviour of the full CFD model in regions with minimal disturbances, such as in the far wake region. However, in the vicinity of the rotor, the process of linearisation might compromise the accuracy of the momentum budget. Nevertheless, empirical evidence indicates that the linearised wake tends to restore the disturbed momentum balance in the near field, achieving reasonable deficits in the far field. This inherent capability enhances the overall accuracy of linearised models [64].

Linearised models offer the advantage of being solvable without the need for a computational grid. This eliminates concerns related to numerical diffusion, which can be problematic in large computational domains. Additionally, issues associated with the generation of spurious mean pressure gradients, arising from systematic errors in the momentum balance, are entirely eliminated in linearised models [64].

2.3. Numerical Methods

Numerical wake models require numerical methods in order to discretise the partial differential equations (PDEs) inherent to numerical models. Two types of numerical methods can be distinguished, numerical methods based on local arguments and numerical methods based on global arguments. Finite element and finite difference methods are examples of numerical methods that have a local character, as the test and trial functions used for these methods have a local character with finite regularities. The spectral method on the other hand employs globally smooth test and trial functions [66].

In this subsection, different numerical methods will be discussed and their respective (dis)advantages will be presented.

2.3.1. Finite Difference Method

The finite difference method (FDM) is based on the application of a local Taylor expansion to approximate partial differential equations (PDEs), resulting in a linear algebraic equation, the finite-difference equation. Despite being the oldest numerical technique, the FDM approach is still the least advantageous when dealing with intricate geometries due to its reliance on structured meshes. This limitation arises from its use of a topologically square grid to discretise the partial differential equations (PDEs) [67, 68]. Another limitation of finite difference methods is that the physical principle of conservation is not automatically enforced. On the other hand, the FDM allows for higher-order approximations, resulting in high-order accuracy of the spatial discretisation [69].

2.3.2. Finite Volume Method

The Finite Volume Method (FVM) emerged as an extension of the finite difference method, originating from the integral formulation of conservation equations [70]. In contrast to the Finite Difference Method (FDM), which relies on nodal relations for differential equations, the FVM discretises the governing equations in integral form [71]. Moreover, most commercial CFD codes use FVM formulations.

From a physical perspective, the FVM describes the continuous equations in terms of local balances over each cell or finite volume. This involves formulating the inflow and outflow from a volume in terms of fluxes at the cell boundaries or facets [72]. The direct balancing of fluxes across adjacent control volumes gives FVM an advantage over the FDM by ensuring the conservation of quantities at the discretised level, such that mass, momentum, and energy remain conserved at a local scale. Additionally, the FVM allows for the utilisation of unstructured meshes [71].

Then, when it comes to enhancing the accuracy of the FVM, mesh refinement can be used similar to the Finite Element Method (FEM). However, unlike the FEM, constructing functions to estimate solutions at higher orders is not as straightforward for the FVM. This discrepancy represents a drawback of the FVM when compared to the FEM. However, the importance of higher order is mostly relevant for LES and DNS.

2.3.3. Finite Element Method

The content of this subsection is largely informed by the doctoral thesis authored by J. Maljaars [72]. The finite element method discretises the weak formulation of the governing conservation laws using a set of basis functions. Numerous finite element approaches are available, however, this section will focus on the Galerkin FEM method. This method is emphasised due to its widespread adoption and, to some extent, has become a synonym for the finite element method [72].

As an introduction to the Galerkin FEM method, an elliptic boundary value problem with non-homogeneous Dirichlet boundary conditions is considered. The elliptic operator in this problem serves as a foundational model for later applications in the advection-diffusion equation and the incompressible Navier-Stokes equations. The problem can be formulated as a system of first-order equations, as expressed in Eq. (2.28). This system seeks to determine $\boldsymbol{\sigma} \in \mathbf{W}$ and $u \in Q$ within the domain Ω, given a source term f and a Dirichlet condition g on the boundary $\partial\Omega$. Notably, when the source term is zero, Eq. (2.28b) shares mathematical resemblance with the incompressibility constraint [72, 73, 74].

$$\boldsymbol{\sigma} = \nabla u \qquad \text{in } \Omega, \tag{2.28a}$$
$$-\nabla \boldsymbol{\sigma} = f \qquad \text{in } \Omega, \tag{2.28b}$$
$$u = g \qquad \text{on } \partial\Omega. \tag{2.28c}$$

The weak formulation of the problem, as outlined in Eq. (2.29), is derived by multiplying Eqs. (2.28a) and (2.28b) by the respective test functions $\boldsymbol{w}$ and q and integrating over the domain Ω. Here, $\boldsymbol{\sigma}$ and u act as trial functions, while $\boldsymbol{w}$ and q are test functions belonging to the function spaces $\mathbf{W}$ and Q, respectively. Furthermore, it is assumed that u satisfies the Dirichlet boundary conditions on $\partial\Omega$, and the test functions q vanish on this boundary. The strength of the weak formulation lies in its ability to ease strict continuity demands on the solution space, allowing for the exploration of approximate solutions within finite-dimensional function spaces [72, 74].

$$\int_{\Omega} (\boldsymbol{\sigma} - \nabla u) \cdot \boldsymbol{w} \ \mathrm{d}\Omega + \int_{\Omega} (\nabla \cdot \boldsymbol{\sigma} + f) q \ \mathrm{d}\Omega = 0 \quad \forall \quad (\boldsymbol{w}, q) \in (\mathbf{W}, Q) \tag{2.29}$$

Transitioning from the weak formulation to a set of algebraic equations necessitates the selection of a finite-dimensional basis. To accomplish this, it is assumed that the variable of interest can be represented using a set of basis functions with known shapes [72].

2.3.3.1. Function Spaces and Mesh Partitioning

Before introducing the continuous and discrete Galerkin methods, functions spaces are introduced. The weak formulation specified in Eq. (2.29) remains valid provided that the gradients ∇u and $\nabla \cdot \boldsymbol{\sigma}$ are square integrable over the domain Ω. For a function ϕ to be considered square integrable, it must satisfy the condition stated in Eq. (2.30) [72].

$$\int_{\Omega} \phi^2 \ \mathrm{d}\Omega < \infty \tag{2.30}$$

Functions exhibiting this characteristic are classified as members of the Sobolev space of degree zero, represented by $\mathrm{L}^2(\Omega)$ for scalar-valued functions (i.e., $\phi \in \mathrm{L}^2(\Omega)$) and $\mathbf{L}^2(\Omega)$ for vector-valued functions. Additionally, for a positive integer k, functions that are square integrable and whose weak derivatives up to order k are square integrable belong to the Sobolev space of degree k, denoted as $\mathrm{H}^k(\Omega)$. Members of the Sobolev space of degree one are frequently used. If the function ϕ belongs to the Hilbert space $\mathrm{H}^1(\Omega)$ or $\mathbf{H}^1(\Omega)$ for vector-valued functions, it implies that Eq. (2.31) is satisfied. Moreover, it holds that $\mathrm{H}^1(\Omega) \subset \mathrm{L}^2(\Omega)$, and that functions $\phi \in \mathrm{H}^1(\Omega)$ are at least C^0 continuous [72, 73].

$$\int_{\Omega} (\phi^2 + \nabla\phi \cdot \nabla\phi) \ \mathrm{d}\Omega < \infty \tag{2.31}$$

Discrete function spaces are constructed based on dividing the domain of interest Ω into a set of separate simplicial elements K, each with its boundary ∂K, as illustrated in Fig. 2.4. The set of cells can be represented as $\mathcal{T} := \{K\}$, and the closure of a cell, encompassing both the cell and its boundaries, is denoted by $\bar{K} := K \cup \partial K$. Moreover, for neighbouring cells K_i and K_j (where i $\neq$ j) there exists a common facet $F = \partial K_i \cup \partial K_j$, and the set $\mathcal{F}$ encompasses all facets, including exterior boundary facets $F = \partial K \cup \partial\Omega$ [72].

Having partitioned the domain, discrete function spaces can be defined. Basis functions for these finite dimensional function spaces often rely on piecewise continuous polynomials denoted as P_k, where $k > 1$ represents the degree of the polynomial. Given this, the function space can be defined as presented in Eq. (2.32) [72].

$$W_h := \{w_h \in H^1(\mathcal{T}) : w_{h|\bar{K}} \in P_k(\bar{K}) \ \forall \ K \in \mathcal{T}\}. \tag{2.32}$$

In the context of $\mathcal{T}$, numerous possibilities exist for Eq. (2.32). However, to narrow down the focus, a decision is made to employ the Bubnov-Galerkin approach. This approach dictates that both trial and test functions originate from the same discrete function space. Consequently, in the presented weak formulation in Eq. (2.29), it follows that $\boldsymbol{\sigma}$ and $\boldsymbol{w}$ are derived from identical function spaces, as are u and q [72].

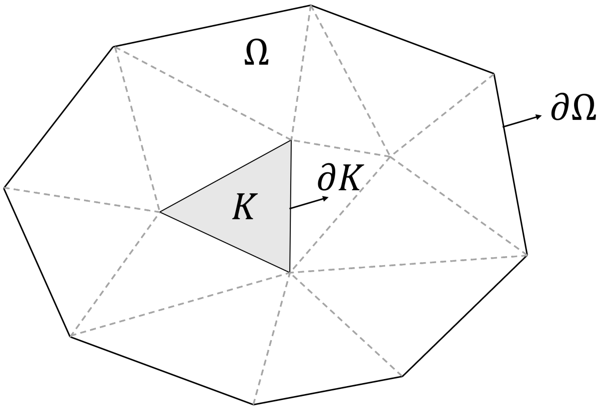


Figure 2.4: Domain partitioning into simplices [72].

2.3.3.2. Continuous Galerkin Method

In the continuous Galerkin (CG) method, the function spaces are structured to ensure that the basis functions maintain continuity across elements. In Eqs. (2.33) and (2.34), the piecewise continuous function spaces are introduced, where, $Q_{h,g}$ is distinguished from Q_h by the fulfilment of the Dirichlet boundary condition at the domain boundary ∂Q [72, 74].

$$Q_h := \{q_h \in H^1(\mathcal{T}) : q_{h|\bar{K}} \in P_k(\bar{K} \ \forall \ K \in \mathcal{T}\} \tag{2.33}$$

$$Q_{h,g} := \{q_h \in H^1(\mathcal{T}) : q_{h|\bar{K}} \in P_k(\bar{K} \ \forall \ K \in \mathcal{T}, q_h = g \text{ on } \partial\Omega\} \tag{2.34}$$

Given the definitions of these function spaces, an irreducible continuous Galerkin method of the mixed form Eq. (2.29) is realised through the substitution of $\boldsymbol{\sigma}$ with ∇u_h and $\boldsymbol{w}$ with ∇q_h. Through the application of integration by parts to ease the continuity constraint on the discrete function space, the variational problem expressed in Eq. (2.29) can be restated as follows: find $u_h \in Q_{h,g}$ such that Eq. (2.35) holds, where $\sum_K$ represents the summation over all cells, and $\vec{n}$ the outward pointing unit normal on the cell boundaries [72, 74].

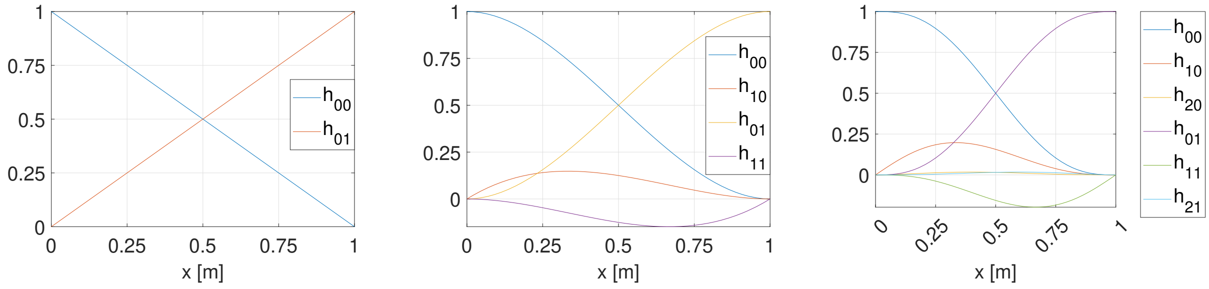
$$\int_\Omega \nabla u_h \cdot \nabla q_h \, \mathrm{d}\Omega - \sum_K \oint_{\partial K} \nabla u_h \cdot \vec{n} \; q_h \, \mathrm{d}\Gamma - \int_\Omega f q_h \, \mathrm{d}\Omega = 0 \qquad \forall \ q_h \in Q_h \tag{2.35}$$

Given the continuity of the function space Q_h across facets, the flux terms over the facets cancel, such that the problem stated in Eq. (2.35) becomes Eq. (2.36) [72].

$$\int_\Omega \nabla u_h \cdot \nabla q_h \, \mathrm{d}\Omega - \int_\Omega f q_h \, \mathrm{d}\Omega = 0 \qquad \forall \ q_h \in Q_h \tag{2.36}$$

Additionally, when it comes to basis functions, polynomial basis functions of the Hermite interpolation schemes serve as a beneficial option for the CG method as highlighted by Kumari et al. [75]. This is due to their ability to ensure the desired continuity of derivatives across cell boundaries. Notably, the odd Hermite basis functions contribute to modelling radial velocity by exhibiting a zero value at $r = 0$, aligning with the foundational requirements for this specific directional aspect. Conversely, the symmetric nature of even Hermite functions complements the axisymmetric assumption [9]. The drawback of these basis functions, however, is that they lack orthogonality.

Moreover, Fig. 2.5 presents the basis functions for the linear, cubic, and quintic Hermite basis respectively. The first index in the legend specifies whether the basis function defines the value on the left or right side of the interval, while the second index indicates whether it defines the value, the first derivative, or the second derivative.



(a) Linear Hermite basis. **(b)** Cubic Hermite basis. **(c)** Quintic Hermite basis.

Figure 2.5: Representation of the Hermite basis functions.

2.3.3.3. Discontinuous Galerkin Method

The discontinuous Galerkin (DG) method employs broken function spaces. In other words, functions within these function spaces are considered to be piecewise continuous within the interior of a cell K, but they may exhibit discontinuities, resulting in being double-valued at the facets ∂K, resulting in a weaker continuity requirement. The DG formulation of the weak form, as outlined in Eq. (2.29), can be derived by utilising the broken function space introduced in Eq. (2.37) [72, 74].

$$Q_h := \{q_h \in L^2(\mathcal{T}) : q_{h|\bar{K}} \in P_k(K)\ \forall\ K \in \mathcal{T}\} \tag{2.37}$$

A crucial distinction from the CG function spaces is that functions $q_h \in Q_h$ are only square-integrable on the set of cells $\mathcal{T}$. Within the interior of cells, these functions q_h are assumed to have square-integrable derivatives. Given these definitions and assuming $w_h = \nabla q_h$ for all $q_h \in Q_h$, the weak form in Eq. (2.29) is reformulated as a DG formulation in Eq. (2.38), where $\hat{\boldsymbol{\sigma}}_h$ and $\hat{u}_h$ are numerical fluxes on the facets ∂K [72, 74]. A detailed derivation can be found in [74].

$$\int_\Omega \nabla u_h \cdot \nabla q_h \ \mathrm{d}\Omega + \sum_K \oint_{\partial K} (\hat{u}_h - u_h)\vec{n} \cdot \nabla q_h \ \mathrm{d}\Gamma - \sum_K \oint_{\partial K} \hat{\boldsymbol{\sigma}}_h \cdot \vec{n}\ q_h \ \mathrm{d}\Gamma - \int_\Omega f g_h \ \mathrm{d}\Omega = 0 \quad \forall\ q_h \in Q_h \tag{2.38}$$

With trial and test functions (u_h, q_h) belonging to Q_h and exhibiting discontinuity between elements, the discontinuous Galerkin (DG) formulation essentially frames the problem as a local balance for each cell. The imposition of continuity between elements involves the careful selection of numerical fluxes, with various formulations proposed in the literature (see [76]). Crucially, these numerical fluxes are expressed in terms of the traces of the primal variable u_h, which possesses a double-valued nature at facets, namely $\hat{\boldsymbol{\sigma}}_h(u_h)$ and $\hat{u}_h(u_h)$ [72].

Additionally, with regard to basis functions, Legendre polynomials are regarded as a suitable choice for the DG method. This is attributed to their definition within the interval from -1 to 1, as depicted in Fig. 2.6, and their lack of dependency on neighbouring cells. Additionally, the orthogonal property of Legendre polynomials enables the efficient approximation of projection integrals using Gaussian quadrature.

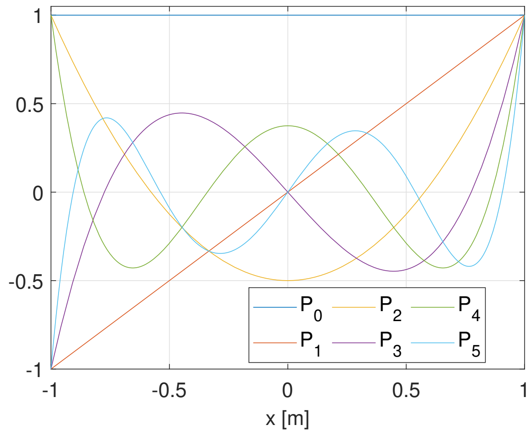


Figure 2.6: Legendre polynomials P_n of degree n.

DG methods are becoming increasingly popular for solving partial differential equations. In contrast to the classical CG method, DG methods offer advantages in accommodating non-conforming meshes and functional adaptivity, while retaining local conservativity and yielding more robust discretisation, particularly in simulations involving high Reynolds numbers [77, 78]. Furthermore, the DG method offers the benefit of weak boundary conditions that do not compromise the accuracy of both the continuity and momentum equations [79].

The flexibility of DG methods is evident from its ease of implementation for refinement and derefinement, allowing for different polynomial orders on various elements. The absence of continuity restraints facilitates the introduction of as many new nodes as needed [80]. Furthermore, the DG method allows for the implementation of different orders of polynomials on distinct elements, contributing to its versatility. A notable feature is the capability of DG methods to be reduced to a balance equation on each element, ensuring local mass conservation. This enables tracking the amount of mass passing through the boundary to other elements, in contrast to other finite element methods that provide only a global mass balance [80].

Despite their versatility, DG methods introduce complexity compared to CG methods, involving intricacies in algorithm formulation, meshing, and data transfer across element boundaries. Moreover, the principal drawback of the DG method lies in its computational efficiency, as DG methods can significantly surpass CG methods in terms of computational expense, especially when dealing with an unstructured mesh. This increased computational demand stems from the fact that DG methods utilise element-wise degrees of freedom rather than nodal ones [79].

2.3.4. Spectral Methods

Spectral methods utilise a set of analytical global basis functions that cover the entire domain for representing flow variables. Typically, trigonometric or polynomial functions are employed for this purpose. Substituting these basis functions into the differential equations yields analytical expressions directly, allowing for the evaluation of the evolution of the flow field. In nonlinear problem scenarios, this approach requires an orthogonal projection of the analytical solution onto the basis functions [74].

Although the spectral method has proven effective in the linearised Fuga model, where a mixed-spectral formulation was implemented [64], it has also been observed to face mathematical challenges, especially in the computation of the triple product integral when applied to solve the Ainslie wake model, and challenges when it comes to the mass and momentum conservation as emphasised in the study by Lokken et al. [9].

2.4. Synthesis

Considering the goal of obtaining a wake model with low computational cost as highlighted in the introduction, the Ainslie wake model emerges as the optimal choice. However, to enhance fidelity, certain assumptions made during the derivation of the Ainslie wake model will be relaxed. More specifically, a pressure term will be implemented by making a solver for the pressure Poisson equation.

Furthermore, for simulating the wind turbine, an actuator disk forcing model will be incorporated. This choice is motivated by its simplicity, ensuring the lowest computational cost. This strategic combination aims to balance computational efficiency with enhanced fidelity, making it a well-suited approach for the intended modelling objectives.

3

Two-Dimensional Ainslie Wake Model

In this chapter, the focus is on the implementation of the continuous FEM for the Ainslie wake model. This method involves utilising Hermite interpolation functions as basis functions in both the radial and axial directions, alongside employing a marching scheme in the axial direction. First in Section 3.1, various approaches for solving the Ainslie wake model with a FEM will be outlined. Subsequently, in Section 3.2 of this chapter, the mathematical implementation of the continuity equation and the momentum equation for the Stokes stream function approach will be derived. While, in Section 3.3 the same is done but for the velocity component approach. Through this discussion, the aim is to provide a comprehensive understanding of the mathematical framework employed in simulating the Ainslie wake model. Furthermore, Section 3.4 provides insight into how the tensor multiplications and matrix multiplications are carried out, while Section 3.5 discusses the non-linear solver used to solve the non-linear governing equations. Finally, Section 3.6 discusses the implementation of the boundary conditions.

3.1. Approach and Boundary Conditions

The implementation strategy for solving the Ainlsie wake model, presented in Subsection 2.2.4.5, revolves around three key aspects: selecting the implementation approach, addressing the implementation of boundary conditions, and determining the necessary order of the basis functions. Each of the three aspects will be elaborated upon in Subsections 3.1.1 to 3.1.3, respectively. Subsequently, a conclusion will be provided in Subsection 3.1.4.

3.1.1. Different Implementation Approaches

There are two primary approaches to solving the Ainslie wake model. The first approach involves treating the radial and axial velocity components as distinct quantities, while the second approach incorporates the Stokes stream function, defined in Eq. (3.1). Opting for the latter method allows for a reduction in the number of equations associated with the Ainslie wake model, leading to a reduction in the number of degrees-of-freedom. Additionally, this approach ensures local conservation of mass, satisfying the strong form of the continuity equation, while the former method only ensures global mass conservation. However, the main disadvantage of this approach is the requirement for higher-order basis functions to ensure fully defined boundary conditions and a fully defined system of equations as the implementation of the Stokes stream function results in higher-order derivatives, reducing the order in both the axial and radial direction.

$$V = -\frac{1}{r}\frac{\partial \psi}{\partial x} \qquad U = \frac{1}{r}\frac{\partial \psi}{\partial r} \tag{3.1}$$

3.1.2. Considerations regarding the Boundary Conditions

For each of the two implementation strategies presented in Subsection 3.1.1, the boundary conditions are presented in Table 3.1, where the boundary conditions for the Stokes stream function approach are derived from those defined for the velocity component approach, utilising the expressions for the radial and axial velocity components in terms of the stream function, as presented in Eq. (3.1).

Upon closer inspection of the boundary conditions at $r = R_{max}$, for the stream function approach, it becomes apparent that particularly the Neumann boundary condition is not well-defined, potentially resulting in undesirably high derivatives of the Stokes stream function at the boundary. To ensure well-defined boundary conditions, a modified version of the stream function, more specifically, the deficit stream function (Eq. (3.2)), could be employed instead of the general Stokes stream function. This modification would enforce $\frac{\partial \psi(R_{max},x)}{\partial r} = 0$ and $\frac{\partial^2 \psi(R_{max},x)}{\partial r^2} = 0$.

$$\psi_D = \psi - U_\infty \frac{r^2}{2} \tag{3.2}$$

Furthermore, regarding the boundary conditions at $r = R_{max}$ in the velocity component approach, particularly for the axial velocity component, it remains undecided whether a linear combination of boundary conditions, Robin boundary conditions, or the separate implementation of the Neumann or Dirichlet boundary conditions would prove to be more effective.

Boundary conditions	Velocity Component Approach		Stokes Stream Function Approach	
at $r = R_{max}$	$V(R_{max}, x) = 0$	Dirichlet	$\psi(R_{max}, x) = \text{constant}$	Dirichlet
	$U(R_{max}, x) = U_\infty$	Dirichlet	$\frac{\partial \psi(R_{max},x)}{\partial r} = U_\infty R_{max}$	Neumann
	$\frac{\partial U(R_{max},x)}{\partial \vec{n}} = \frac{\partial U(R_{max},x)}{\partial r} = 0$	Neumann	$\frac{\partial^2 \psi(R_{max},x)}{\partial r^2} = U_\infty$	
at $r = 0$ m	$V(0, x) = 0$	Dirichlet	$\psi(0, x) = \text{constant}$	Dirichlet
	$\frac{\partial U(0,x)}{\partial r} = 0$	Neumann	$\frac{\partial \psi(0,x)}{\partial r} = rU(0, x) = 0$	Neumann
	$\frac{\partial^{2n+1} U(0,x)}{\partial r^{2n+1}} = 0$		$\frac{\partial^2 \psi(0,x)}{\partial r^2} = 0$	
	$\frac{\partial^{2n} V(0,x)}{\partial r^{2n}} = 0$			

Table 3.1: Boundary conditions for the different implementation approaches.

3.1.3. Required Order of the Basis Functions

The minimum degree of the basis functions to ensure continuity across the entire domain (i.e. within the cells and across the cell boundaries) of the differential operator after having applied integration by parts relies on the highest order derivative present in both the boundary conditions and the equations governing the wake model.

Starting with the basis functions used in the velocity components approach. The highest order derivative in the radial direction when considering both the boundary conditions specified in Table 3.1 and the governing equations, is the second order derivative originating from the diffusion term in the momentum equation. Applying integration by parts to the diffusion term requires the first-order derivative to be continuous over the boundaries, similar to the requirements stemming from the boundary conditions. Consequently, at least cubic Hermite interpolation for the basis functions in the radial direction is required to ensure the continuity of the differential operator. Higher-order Hermite interpolation basis functions are also a possibility but not necessary and therefore not considered for this approach. Furthermore, for the basis functions in the axial direction, the lowest order of basis functions applicable are piecewise linear basis functions, as there are no first-order derivatives present that need to be evaluated at the boundaries.

Then, considering the boundary conditions and governing equations for the Stokes stream function approach, the highest-order derivatives in radial direction evaluated at the boundaries are second-order derivatives, originating from both the boundary evaluation term of the diffusion term after applying integration by parts, and from the boundary conditions, as observed from Table 3.1. Therefore, compared to the velocity component approach, it can be inferred that the cubic Hermite interpolation basis functions in the radial direction become less viable due to the discontinuities observed in their second derivatives at the cell boundaries. Consequently, upgrading the basis functions to quintic order in the radial direction becomes necessary for the stream function approach. However, for the axial direction, piecewise linear basis functions are still sufficient.

3.1.4. Conclusion

Following the discussion in this section, two primary options are proposed for solving the Ainslie wake model using the FEM. The first option involves implementing the velocity component approach, employing cubic Hermite interpolation basis functions in the radial direction and piecewise linear basis functions in the axial direction. Conversely, the second approach employs the Stokes stream function with quintic Hermite interpolation basis functions in the radial direction and piecewise linear basis functions in the axial direction. Furthermore, for this latter approach, it is preferable to utilise the deficit stream function to ensure well-defined boundary conditions.

Both approaches resulting from the discussion have their respective advantages and disadvantages. In theory, when prioritising computational efficiency and emphasising the fulfillment of the continuity equation in a strong sense, the stream function approach is favoured over the velocity component approach. However, during the implementation phase, new advantages and disadvantages of both approaches may emerge, potentially leading to a slightly altered approach. As such, both implementation methods will be elaborated upon in the subsequent sections.

3.2. Implementation Stokes Stream Function Approach

In this section, first, the continuity equation for the Stokes stream function approach will be discussed in Subsection 3.2.1. Subsequently, a comprehensive discussion on the discretisation of the momentum equation for the Stokes stream function approach using the FEM will be presented in Subsection 3.2.2.

3.2.1. Implementation Continuity Equation

The starting point in this section is the axisymmetric, time-averaged, incompressible continuity equation, as outlined in Eq. (3.3), where U is the time-averaged axial velocity and V the time-averaged radial velocity.

$$\frac{\partial U(r,x)}{\partial x} + \frac{1}{r}\frac{\partial}{\partial r}(rV(x,r)) = 0 \tag{3.3}$$

The stream function approach achieves a pointwise satisfaction of the continuity equation as discussed in Section 3.1. This can be shown by applying the definition of the stream function outlined in Eq. (3.1) in Eq. (3.3), as presented in Eq. (3.4).

$$\frac{\partial}{\partial x}\left(\frac{1}{r}\frac{\partial \psi(x,r)}{\partial r}\right) + \frac{1}{r}\frac{\partial}{\partial r}\left(r\left(\frac{-1}{r}\frac{\partial \psi(x,r)}{\partial x}\right)\right) = \frac{1}{r}\frac{\partial \psi(x,r)}{\partial x \partial r} - \frac{1}{r}\frac{\partial \psi(x,r)}{\partial x \partial r} = 0 \tag{3.4}$$

3.2.2. Implementation Momentum Equation

The momentum equation considered for this analysis follows the form outlined in Eq. (3.5), which is derived in Subsection 2.2.4.5.

$$U\frac{\partial U}{\partial x} + V\frac{\partial U}{\partial r} = \frac{1}{r}\frac{\partial}{\partial r}\left(r\left(\nu_t\frac{\partial U}{\partial r}\right)\right) \tag{3.5}$$

Furthermore, when considering the stream function approach for the transport equation, as outlined in Eq. (3.5), one may choose to initiate the analysis from either the convective form of the momentum equation, outlined in Eq. (3.6), or from the Cauchy momentum equation, as depicted in Eq. (3.7). For this thesis, the decision has been made to employ the convective form of the momentum equation for the analysis.

$$\left(\frac{1}{r}\frac{\partial \psi}{\partial r}\right)\left(\frac{1}{r}\frac{\partial^2 \psi}{\partial x \partial r}\right) + \left(-\frac{1}{r}\frac{\partial \psi}{\partial x}\right)\left(\frac{\partial}{\partial r}\left[\frac{1}{r}\frac{\partial \psi}{\partial r}\right]\right) = \frac{1}{r}\frac{\partial}{\partial r}\left(r\left(\nu_t\frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial \psi}{\partial r}\right)\right)\right) \tag{3.6}$$

$$\nabla \cdot \left(\begin{pmatrix} -\frac{1}{r}\frac{\partial \psi}{\partial x} \\ \frac{1}{r}\frac{\partial \psi}{\partial r} \end{pmatrix}\frac{1}{r}\frac{\partial \psi}{\partial r}\right) = \nabla \cdot \left(\underline{\nu_t}\begin{pmatrix} \frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial \psi}{\partial r}\right) \\ 0 \end{pmatrix}\right) \tag{3.7}$$

Prior to discretising the terms, it is crucial to explore and determine an appropriate discretisation approach. One possibility is the utilisation of a marching scheme in the axial direction. This approach involves systematically progressing through a computational domain, advancing step by step from one boundary to another. The procedure starts with established values at the left boundary, and the algorithm takes consecutive steps in the axial direction, calculating values at each successive right cell boundary until it reaches the right boundary at the end of the domain. This means that instead of solving a global $M \times N$ system an order M system is solved N times. It should be noted that the use of the marching scheme is justified by the fact that the parabolised form of the Navier-Stokes equations is being solved.

The marching scheme's advantage lies in its effectiveness for problems that require a systematic, incremental strategy, offering a clear and organised approach to propagating information throughout the computational domain. However, in case of backflow of information, indicating a negative velocity, the marching scheme becomes unsuitable for CFD simulations, necessitating a direct solution. To overcome this challenge, one needs to ensure the absence of adverse pressure gradients. This effectively prevents the downstream flow of information, ensuring the axial velocity stays non-negative. However, this modification restricts the approach to situations dominated by advection, potentially presenting a challenge in maintaining satisfactory continuity and information flow in fluid simulations [9]. In this thesis, a marching scheme approach has been selected for implementation.

3.2.2.1. Diffusion Term

In this section, the diffusion term described in Eq. (3.8) will be discretised. For this purpose, the stream function will be expressed as a linear combination of basis functions in both the axial and radial directions, as presented in Eq. (3.9). It should be noted that to mitigate issues arising from terms like $\frac{1}{r}$ when r approaches zero, the basis function in the radial direction is modified to be a function of r^2 rather than r.

$$RHS = \frac{1}{r}\frac{\partial}{\partial r}\left(r\left(\nu_t \frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial \psi}{\partial r}\right)\right)\right) \tag{3.8}$$

$$\psi(r^2, x) = \sum_{n=0}^{N_r}\sum_{k=0}^{N_x} \psi_{(n,k)} b_n(r^2) a_k(x) \tag{3.9}$$

Furthermore, when employing the marching scheme, the axial component of the stream function can be divided into two parts: known components derived from either initial conditions or the preceding step, and unknown components. The known coefficients of the stream function are linked to its values on the left side of the elements, denoted by the subscript l, whereas the coefficients associated with unknown values are attributed to the right side of the elements, indicated by the subscript r. The same notation is applied to the basis functions forming the velocity values on both the left and right sides of the elements, resulting in Eq. (3.10).

$$\psi(r, x) = \sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2} \psi_{(n,k_l)} b_n(r^2) a_{k_l}(x) + \psi_{(n,k_r)} b_n(r^2) a_{k_r}(x) \tag{3.10}$$

The first step in the discretisation process involves implementing the linear combination for the stream function in Eq. (3.8). However, before incorporating the linear combination for the stream function, the expression for the diffusion term is slightly modified to simplify the subsequent steps, as presented in Eq. (3.11). The resulting expression is outlined in Eq. (3.12), which incorporates the characteristics of $a_k(x)$ and $b_n(r^2)$ being dependent solely on the axial and radial directions, respectively, and ν_t being dependent exclusively on the axial direction.

$$RHS = \frac{1}{r}\frac{\partial}{\partial r}\left(\nu_t \left[\frac{-1}{r}\frac{\partial \psi}{\partial r} + \frac{\partial^2 \psi}{\partial r^2}\right]\right) \tag{3.11}$$

$$
\begin{aligned}
RHS = & \sum_{n=0}^{N_r} \sum_{k=0}^{N_x/2} \psi_{(n,k_l)} \left(\frac{1}{r} \frac{\partial}{\partial r} \left(\nu_t \left[\frac{-1}{r} \frac{\partial}{\partial r} \left(b_n(r^2) a_{k_l}(x) \right) + \frac{\partial^2}{\partial r^2} \left(b_n(r^2) a_{k_l}(x) \right) \right] \right) \right) \\
& + \sum_{n=0}^{N_r} \sum_{k=0}^{N_x/2} \psi_{(n,k_r)} \left(\frac{1}{r} \frac{\partial}{\partial r} \left(\nu_t \left[\frac{-1}{r} \frac{\partial}{\partial r} \left(b_n(r^2) a_{k_r}(x) \right) + \frac{\partial^2}{\partial r^2} \left(b_n(r^2) a_{k_r}(x) \right) \right] \right) \right) \\
= & \sum_{n=0}^{N_r} \sum_{k=0}^{N_x/2} \psi_{(n,k_l)} \left(\frac{1}{r} \frac{\partial}{\partial r} \left(\frac{-1}{r} \frac{\partial b_n(r^2)}{\partial r} + \frac{\partial^2 b_n(r^2)}{\partial r^2} \right) \right) \left(\nu_t a_{k_l}(x) \right) \\
& + \sum_{n=0}^{N_r} \sum_{k=0}^{N_x/2} \psi_{(n,k_r)} \left(\frac{1}{r} \frac{\partial}{\partial r} \left(\frac{-1}{r} \frac{\partial b_n(r^2)}{\partial r} + \frac{\partial^2 b_n(r^2)}{\partial r^2} \right) \right) \left(\nu_t a_{k_r}(x) \right)
\end{aligned}
\tag{3.12}
$$

Applying the Galerkin projection in the axial and radial directions onto the basis functions $a_i(x)$ and $b_m(r^2)$, respectively, using a marching scheme approach, yields Eq. (3.13). During the implementation of the marching scheme, it is essential to emphasise that in order to have a square matrix, which is necessary for taking the inverse operation required to solve the equations, it is sufficient to project either onto the left or the right basis functions, or another set of linearly independent basis functions which satisfy this. In this case a projection onto the right axial basis functions, denoted by $a_{i_r}(x)$, has been opted for.

$$
\begin{aligned}
\int_{X_l}^{X_r} \int_0^{R_{max}} RHS \; r \mathrm{d}r \mathrm{d}x = & \sum_{n=0}^{N_r} \sum_{k=0}^{N_x/2} \psi_{(n,k_l)} \int_0^{R_{max}} b_m(r^2) \frac{\partial}{\partial r} \left(\frac{-1}{r} \frac{\partial b_n(r^2)}{\partial r} \right) \mathrm{d}r \int_{X_l}^{X_r} \nu_t a_{i_r}(x) a_{k_l}(x) \mathrm{d}x \\
& + \sum_{n=0}^{N_r} \sum_{k=0}^{N_x/2} \psi_{(n,k_r)} \int_0^{R_{max}} b_m(r^2) \frac{\partial}{\partial r} \left(\frac{-1}{r} \frac{\partial b_n(r^2)}{\partial r} \right) \mathrm{d}r \int_{X_l}^{X_r} \nu_t a_{i_r}(x) a_{k_r}(x) \mathrm{d}x \\
& + \sum_{n=0}^{N_r} \sum_{k=0}^{N_x/2} \psi_{(n,k_l)} \int_0^{R_{max}} b_m(r^2) \frac{\partial}{\partial r} \left(\frac{\partial^2 b_n(r^2)}{\partial r^2} \right) \mathrm{d}r \int_{X_l}^{X_r} \nu_t a_{i_r}(x) a_{k_l}(x) \mathrm{d}x \\
& + \sum_{n=0}^{N_r} \sum_{k=0}^{N_x/2} \psi_{(n,k_r)} \int_0^{R_{max}} b_m(r^2) \frac{\partial}{\partial r} \left(\frac{\partial^2 b_n(r^2)}{\partial r^2} \right) \mathrm{d}r \int_{X_l}^{X_r} \nu_t a_{i_r}(x) a_{k_r}(x) \mathrm{d}x
\end{aligned}
\tag{3.13}
$$

Subsequently, integration by parts is utilised on the second-order derivative term in the axial direction, as well as on the second- and third-order derivatives in the radial direction. This approach serves to relax the constraint on the specific order of the radial basis functions and to prevent encountering derivatives of a product of dependent terms, to ease the coding implementation. Implementing this yields Eq. (3.14).

$$
\begin{aligned}
&\int_{X_l}^{X_r}\int_{0}^{R_{max}} RHS\ r\mathrm{d}r\mathrm{d}x \\
=&\sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2}\psi_{(n,k_l)}\left(\underbrace{b_m(r^2)\left(\frac{-1}{r}\frac{\partial b_n(r^2)}{\partial r}\right)\Bigg|_{0}^{R_{max}} - \int_{0}^{R_{max}}\frac{\partial b_m(r^2)}{\partial r}\left(\frac{-1}{r}\frac{\partial b_n(r^2)}{\partial r}\right)\mathrm{d}r}_{\text{Matrix A}_1}\right)\underbrace{\int_{X_l}^{X_r}\nu_t a_{i_r}(x)a_{k_l}(x)\mathrm{d}x}_{\text{Matrix B}_l} \\
+&\sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2}\psi_{(n,k_r)}\left(\underbrace{b_m(r^2)\left(\frac{-1}{r}\frac{\partial b_n(r^2)}{\partial r}\right)\Bigg|_{0}^{R_{max}} - \int_{0}^{R_{max}}\frac{\partial b_m(r^2)}{\partial r}\left(\frac{-1}{r}\frac{\partial b_n(r^2)}{\partial r}\right)\mathrm{d}r}_{\text{Matrix A}_1}\right)\underbrace{\int_{X_l}^{X_r}\nu_t a_{i_r}(x)a_{k_r}(x)\mathrm{d}x}_{\text{Matrix B}_r} \\
+&\sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2}\psi_{(n,k_l)}\left(\underbrace{b_m(r^2)\frac{\partial^2 b_n(r^2)}{\partial r^2}\Bigg|_{0}^{R_{max}} - \int_{0}^{R_{max}}\frac{\partial b_m(r^2)}{\partial r}\frac{\partial^2 b_n(r^2)}{\partial r^2}\mathrm{d}r}_{\text{Matrix A}_2}\right)\underbrace{\int_{X_l}^{X_r}\nu_t a_{i_r}(x)a_{k_l}(x)\mathrm{d}x}_{\text{Matrix B}_l} \\
+&\sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2}\psi_{(n,k_r)}\left(\underbrace{b_m(r^2)\frac{\partial^2 b_n(r^2)}{\partial r^2}\Bigg|_{0}^{R_{max}} - \int_{0}^{R_{max}}\frac{\partial b_m(r^2)}{\partial r}\frac{\partial^2 b_n(r^2)}{\partial r^2}\mathrm{d}r}_{\text{Matrix A}_2}\right)\underbrace{\int_{X_l}^{X_r}\nu_t a_{i_r}(x)a_{k_r}(x)\mathrm{d}x}_{\text{Matrix B}_r}
\end{aligned}
\tag{3.14}
$$

However, Eq. (3.14) is not completely refined yet, since the radial basis functions are operated upon by r^2 instead of r. For this purpose, r^2 is referred to by the variable s. Before rewriting the radially dependent terms in Eq. (3.14) first some relations between r and s that are necessary for the derivation are presented. More specifically, the general relation between r and s as well as the relation between $\mathrm{d}r$ and $\mathrm{d}s$ is provided in Eq. (3.15). Furthermore, also the term $\frac{\partial b(r^2)}{\partial r}$ is worked out before starting the derivation as outlined in Eq. (3.16), where $b(r^2)$ is a general basis function dependent on r^2.

$$r = \sqrt{s} \qquad \text{and} \qquad \mathrm{d}r = \frac{1}{2\sqrt{s}}\mathrm{d}s \tag{3.15}$$

$$\frac{\partial b(r^2)}{\partial r} = \frac{\partial b(s)}{\partial r} = \frac{\partial b(s)}{\partial s}\frac{\partial s}{\partial r} = \frac{\partial b(s)}{\partial s}2\sqrt{s} \tag{3.16}$$

The first term that will be worked out is Matrix A_1 in Eq. (3.14). For this, it is important to note that changing the integration variable necessitates corresponding adjustments to the integration limits. The result is given by Eq. (3.17).

$$
\begin{aligned}
&b_m(r^2)\left(\frac{-1}{r}\frac{\partial b_n(r^2)}{\partial r}\right)\Bigg|_0^{R_{max}} - \int\limits_0^{R_{max}} \frac{\partial b_m(r^2)}{\partial r}\left(\frac{-1}{r}\frac{\partial b_n(r^2)}{\partial r}\right)\mathrm{d}r \\
&= b_m(s)\left(\frac{-1}{\sqrt{s}}\frac{\partial b_n(s)}{\partial r}\right)\Bigg|_0^{R_{max}} - \int\limits_0^{R_{max}} \frac{\partial b_m(s)}{\partial r}\left(\frac{-1}{\sqrt{s}}\frac{\partial b_n(s)}{\partial r}\right)\mathrm{d}r \\
&= b_m(s)\left(\frac{-1}{\sqrt{s}}\frac{\partial b_n(s)}{\partial s}2\sqrt{s}\right)\Bigg|_0^{R_{max}^2} - \int\limits_0^{R_{max}} \frac{\partial b_m(s)}{\partial s}2\sqrt{s}\left(\frac{-1}{\sqrt{s}}\frac{\partial b_n(s)}{\partial s}2\sqrt{s}\right)\frac{1}{2\sqrt{s}}\mathrm{d}s \\
&= -2b_m(s)\frac{\partial b_n(s)}{\partial s}\Bigg|_0^{R_{max}^2} + \int\limits_0^{R_{max}^2} 2\frac{\partial b_m(s)}{\partial t}\frac{\partial b_n(s)}{\partial s}\mathrm{d}s
\end{aligned}
\tag{3.17}
$$

Proceeding with Matrix A$_2$, the worked-out derivation is provided in Eq. (3.18).

$$
\begin{aligned}
&b_m(r^2)\frac{\partial^2 b_n(r^2)}{\partial r^2}\Bigg|_0^{R_{max}} - \int\limits_0^{R_{max}} \frac{\partial b_m(r^2)}{\partial r}\frac{\partial^2 b_n(r^2)}{\partial r^2}\mathrm{d}r \\
&= b_m(s)\frac{\partial}{\partial r}\left(\frac{\partial b_n(s)}{\partial r}\right)\Bigg|_0^{R_{max}} - \int\limits_0^{R_{max}} \frac{\partial b_m(s)}{\partial r}\frac{\partial}{\partial r}\left(\frac{\partial b_n(s)}{\partial r}\right)\mathrm{d}r \\
&= b_m(s)\frac{\partial}{\partial r}\left(\frac{\partial b_n(s)}{\partial s}2\sqrt{s}\right)\Bigg|_0^{R_{max}^2} - \int\limits_0^{R_{max}^2} \left(\frac{\partial b_m(s)}{\partial s}2\sqrt{s}\right)\frac{\partial}{\partial r}\left(\frac{\partial b_n(s)}{\partial s}2\sqrt{s}\right)\frac{1}{2\sqrt{s}}\mathrm{d}s \\
&= b_m(s)\left(2\frac{\partial b_n(s)}{\partial s} + \frac{\partial}{\partial s}\left(\frac{\partial b_n(s)}{\partial r}\right)2\sqrt{s}\right)\Bigg|_0^{R_{max}^2} - \int\limits_0^{R_{max}^2} \left(\frac{\partial b_m(s)}{\partial s}2\sqrt{s}\right)\left(2\frac{\partial b_n(s)}{\partial s} + \frac{\partial}{\partial s}\left(\frac{\partial b_n(s)}{\partial r}\right)2\sqrt{s}\right)\frac{1}{2\sqrt{s}}\mathrm{d}s \\
&= b_m(s)\left(2\frac{\partial b_n(s)}{\partial s} + \frac{\partial}{\partial s}\left(\frac{\partial b_n(s)}{\partial s}2\sqrt{s}\right)2\sqrt{s}\right)\Bigg|_0^{R_{max}^2} - \int\limits_0^{R_{max}^2} \left(\frac{\partial b_m(s)}{\partial s}2\sqrt{s}\right)\left(2\frac{\partial b_n(s)}{\partial s} + \frac{\partial}{\partial s}\left(\frac{\partial b_n(s)}{\partial s}2\sqrt{s}\right)2\sqrt{s}\right)\frac{1}{2\sqrt{s}}\mathrm{d}s \\
&= 4b_m(s)\frac{\partial b_n(s)}{\partial s} + 4s\; b_m(s)\frac{\partial^2 b_n(s)}{\partial s^2}\Bigg|_0^{R_{max}^2} - \int\limits_0^{R_{max}^2} 4\frac{\partial b_m(s)}{\partial s}\frac{\partial b_n(s)}{\partial s}\mathrm{d}s - \int\limits_0^{R_{max}^2} 4s\frac{\partial b_m(s)}{\partial s}\frac{\partial^2 b_n(s)}{\partial s^2}\mathrm{d}s
\end{aligned}
\tag{3.18}
$$

This leads to the final expression for the diffusion term in terms of s (or r^2), as depicted in Eq. (3.19). Notably, Matrix A$_1$ and Matrix A$_{2,1}$, can be combined into one term resulting in the final expression for the diffusion term presented in Eq. (3.20).

$$
\begin{aligned}
\int\limits_{X_l}^{X_r}\int\limits_{0}^{R_{max}} RHS\ r\mathrm{d}r\mathrm{d}x = {} & \sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2}\psi_{(n,k_l)}\left(\underbrace{-2b_m(s)\frac{\partial b_n(s)}{\partial s}\bigg|_{0}^{R_{max}^2} + \int\limits_{0}^{R_{max}^2} 2\frac{\partial b_m(s)}{\partial s}\frac{\partial b_n(s)}{\partial s}\mathrm{d}s}_{\text{Matrix A}_1}\right)\underbrace{\int\limits_{X_l}^{X_r}\nu_t a_{i_r}(x)a_{k_l}(x)\mathrm{d}x}_{\text{Matrix B}_l} \\
& +\sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2}\psi_{(n,k_r)}\left(\underbrace{-2b_m(s)\frac{\partial b_n(s)}{\partial s}\bigg|_{0}^{R_{max}^2} + \int\limits_{0}^{R_{max}^2} 2\frac{\partial b_m(s)}{\partial s}\frac{\partial b_n(s)}{\partial s}\mathrm{d}s}_{\text{Matrix A}_1}\right)\underbrace{\int\limits_{X_l}^{X_r}\nu_t a_{i_r}(x)a_{k_r}(x)\mathrm{d}x}_{\text{Matrix B}_r} \\
& +\sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2}\psi_{(n,k_l)}\left(\underbrace{4b_m(s)\frac{\partial b_n(s)}{\partial s}\bigg|_{0}^{R_{max}^2} - \int\limits_{0}^{R_{max}^2} 4\frac{\partial b_m(s)}{\partial s}\frac{\partial b_n(s)}{\partial s}\mathrm{d}s}_{\text{Matrix A}_{2,1}}\right)\underbrace{\int\limits_{X_l}^{X_r}\nu_t a_{i_r}(x)a_{k_l}(x)\mathrm{d}x}_{\text{Matrix B}_l} \\
& +\sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2}\psi_{(n,k_r)}\left(\underbrace{4b_m(s)\frac{\partial b_n(s)}{\partial s}\bigg|_{0}^{R_{max}^2} - \int\limits_{0}^{R_{max}^2} 4\frac{\partial b_m(s)}{\partial s}\frac{\partial b_n(s)}{\partial s}\mathrm{d}s}_{\text{Matrix A}_{2,1}}\right)\underbrace{\int\limits_{X_l}^{X_r}\nu_t a_{i_r}(x)a_{k_r}(x)\mathrm{d}x}_{\text{Matrix B}_r} \\
& +\sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2}\psi_{(n,k_l)}\left(\underbrace{4s\ b_m(s)\frac{\partial^2 b_n(s)}{\partial s^2}\bigg|_{0}^{R_{max}^2} - \int\limits_{0}^{R_{max}^2} 4s\frac{\partial b_m(s)}{\partial s}\frac{\partial^2 b_n(s)}{\partial s^2}\mathrm{d}s}_{\text{Matrix A}_{2,2}}\right)\underbrace{\int\limits_{X_l}^{X_r}\nu_t a_{i_r}(x)a_{k_l}(x)\mathrm{d}x}_{\text{Matrix B}_l} \\
& +\sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2}\psi_{(n,k_r)}\left(\underbrace{4s\ b_m(s)\frac{\partial^2 b_n(s)}{\partial s^2}\bigg|_{0}^{R_{max}^2} - \int\limits_{0}^{R_{max}^2} 4s\frac{\partial b_m(s)}{\partial s}\frac{\partial^2 b_n(s)}{\partial s^2}\mathrm{d}s}_{\text{Matrix A}_{2,2}}\right)\underbrace{\int\limits_{X_l}^{X_r}\nu_t a_{i_r}(x)a_{k_r}(x)\mathrm{d}x}_{\text{Matrix B}_r}
\end{aligned}
\tag{3.19}
$$

$$\begin{aligned}
\int_{X_l}^{X_r}\int_{0}^{R_{max}} RHS\ r\mathrm{d}r\mathrm{d}x =& \sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2}\psi_{(n,k_l)}\left(\underbrace{2b_m(s)\frac{\partial b_n(s)}{\partial s}\bigg|_{0}^{R^2_{max}} - \int_{0}^{R^2_{max}} 2\frac{\partial b_m(s)}{\partial s}\frac{\partial b_n(s)}{\partial s}\mathrm{d}s}_{\text{Matrix A}_1+\text{A}_{2,1}}\right)\underbrace{\int_{X_l}^{X_r}\nu_t a_{i_r}(x)a_{k_l}(x)\mathrm{d}x}_{\text{Matrix B}_l}\\
&+\sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2}\psi_{(n,k_r)}\left(\underbrace{2b_m(s)\frac{\partial b_n(s)}{\partial s}\bigg|_{0}^{R^2_{max}} - \int_{0}^{R^2_{max}} 2\frac{\partial b_m(s)}{\partial s}\frac{\partial b_n(s)}{\partial s}\mathrm{d}s}_{\text{Matrix A}_1+\text{A}_{2,1}}\right)\underbrace{\int_{X_l}^{X_r}\nu_t a_{i_r}(x)a_{k_r}(x)\mathrm{d}x}_{\text{Matrix B}_r}\\
&+\sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2}\psi_{(n,k_l)}\left(\underbrace{4s\ b_m(s)\frac{\partial^2 b_n(s)}{\partial s^2}\bigg|_{0}^{R^2_{max}} - \int_{0}^{R^2_{max}} 4s\frac{\partial b_m(s)}{\partial s}\frac{\partial^2 b_n(s)}{\partial s^2}\mathrm{d}s}_{\text{Matrix A}_{2,2}}\right)\underbrace{\int_{X_l}^{X_r}\nu_t a_{i_r}(x)a_{k_l}(x)\mathrm{d}x}_{\text{Matrix B}_l}\\
&+\sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2}\psi_{(n,k_r)}\left(\underbrace{4s\ b_m(s)\frac{\partial^2 b_n(s)}{\partial s^2}\bigg|_{0}^{R^2_{max}} - \int_{0}^{R^2_{max}} 4s\frac{\partial b_m(s)}{\partial s}\frac{\partial^2 b_n(s)}{\partial s^2}\mathrm{d}s}_{\text{Matrix A}_{2,2}}\right)\underbrace{\int_{X_l}^{X_r}\nu_t a_{i_r}(x)a_{k_r}(x)\mathrm{d}x}_{\text{Matrix B}_r}
\end{aligned}\tag{3.20}$$

Lastly, implementing the deficit stream function into Eq. (3.11) yields a diffusion term formulation that remains consistent, though necessitating adaptation of the coefficients to suit the deficit stream function, as derived in Eq. (3.21).

$$\begin{aligned}
RHS =& \frac{1}{r}\frac{\partial}{\partial r}\left(r\left(\nu_t\frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial}{\partial r}\left(\psi_D + U_\infty\frac{r^2}{2}\right)\right)\right)\right)\\
=& \frac{1}{r}\frac{\partial}{\partial r}\left(r\left(\nu_t\frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial \psi_D}{\partial r} + \frac{1}{r}(U_\infty r)\right)\right)\right)\\
=& \frac{1}{r}\frac{\partial}{\partial r}\left(\nu_t\left[\frac{-1}{r}\frac{\partial \psi_D}{\partial r} + \frac{\partial^2 \psi_D}{\partial r^2}\right]\right)
\end{aligned}\tag{3.21}$$

3.2.2.2. Non-Linear Advection Term

In this section, the non-linear advection term described in Eq. (3.22) will be discretised. The first step in this discretisation process is the implementation of the deficit stream function, yielding Eq. (3.23).

$$LHS = \left(\frac{1}{r}\frac{\partial \psi}{\partial r}\right)\left(\frac{1}{r}\frac{\partial^2 \psi}{\partial x \partial r}\right) + \left(-\frac{1}{r}\frac{\partial \psi}{\partial x}\right)\left(\frac{\partial}{\partial r}\left[\frac{1}{r}\frac{\partial \psi}{\partial r}\right]\right)\tag{3.22}$$

$$\begin{aligned}
LHS =& \left(\frac{1}{r}\frac{\partial}{\partial r}\left(\psi_D + U_\infty\frac{r^2}{2}\right)\right)\left(\frac{1}{r}\frac{\partial^2}{\partial x \partial r}\left(\psi_D + U_\infty\frac{r^2}{2}\right)\right) + \left(-\frac{1}{r}\frac{\partial}{\partial x}\left(\psi_D + U_\infty\frac{r^2}{2}\right)\right)\left(\frac{\partial}{\partial r}\left[\frac{1}{r}\frac{\partial}{\partial r}\left(\psi_D + U_\infty\frac{r^2}{2}\right)\right]\right)\\
=& \left(\frac{1}{r}\frac{\partial \psi_D}{\partial r} + U_\infty\right)\left(\frac{1}{r}\frac{\partial^2 \psi_D}{\partial x \partial r}\right) + \left(-\frac{1}{r}\frac{\partial \psi_D}{\partial x}\right)\left(\frac{\partial}{\partial r}\left[\frac{1}{r}\frac{\partial \psi_D}{\partial r}\right]\right)
\end{aligned}\tag{3.23}$$

After having implemented the deficit stream function, the subsequent step involves implementing the linear combination representation of the stream function. For this objective, the stream function will be formulated using the same linear combination as for the diffusion term, as presented in Eq. (3.10). Employing a marching scheme approach yields the result shown in Eq. (3.24), which can be rewritten to obtain the expression outlined in Eq. (3.25).

$$
\begin{aligned}
LHS = &\sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2}\left(\frac{1}{r}\frac{\partial}{\partial r}\left(\psi_{D,(n,j_l)}b_n(r^2)a_{j_l}(x)+\psi_{D,(n,j_r)}b_n(r^2)a_{j_r}(x)\right)+U_\infty\right)\left(\frac{1}{r}\frac{\partial^2}{\partial x\partial r}\left(\psi_{D,(p,k_l)}b_p(r^2)a_{k_l}(x)+\psi_{D,(p,k_r)}b_p(r^2)a_{k_r}(x)\right)\right)\\
&+\sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2}\left(-\frac{1}{r}\frac{\partial}{\partial x}\left(\psi_{D,(n,j_l)}b_n(r^2)a_{j_l}(x)+\psi_{D,(n,j_r)}b_n(r^2)a_{j_r}(x)\right)\right)\left(\frac{\partial}{\partial r}\left[\frac{1}{r}\frac{\partial}{\partial r}\left(\psi_{D,(p,k_l)}b_p(r^2)a_{k_l}(x)+\psi_{D,(p,k_r)}b_p(r^2)a_{k_r}(x)\right)\right]\right)
\end{aligned}
\tag{3.24}
$$

$$
\begin{aligned}
LHS = &\sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2}\psi_{D,(n,j_l)}\psi_{D,(p,k_l)}\left(\frac{1}{r}\frac{\partial b_n(r^2)}{\partial r}\frac{1}{r}\frac{\partial b_p(r^2)}{\partial r}\right)\left(a_{j_l}(x)\frac{\partial a_{k_l}(x)}{\partial x}\right)\\
&+\sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2}\psi_{D,(n,j_l)}\psi_{D,(p,k_r)}\left(\frac{1}{r}\frac{\partial b_n(r^2)}{\partial r}\frac{1}{r}\frac{\partial b_p(r^2)}{\partial r}\right)\left(a_{j_l}(x)\frac{\partial a_{k_r}(x)}{\partial x}\right)\\
&+\sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2}\psi_{D,(n,j_r)}\psi_{D,(p,k_l)}\left(\frac{1}{r}\frac{\partial b_n(r^2)}{\partial r}\frac{1}{r}\frac{\partial b_p(r^2)}{\partial r}\right)\left(a_{j_r}(x)\frac{\partial a_{k_l}(x)}{\partial x}\right)\\
&+\sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2}\psi_{D,(n,j_r)}\psi_{D,(p,k_r)}\left(\frac{1}{r}\frac{\partial b_n(r^2)}{\partial r}\frac{1}{r}\frac{\partial b_p(r^2)}{\partial r}\right)\left(a_{j_r}(x)\frac{\partial a_{k_r}(x)}{\partial x}\right)\\
&+\sum_{p=0}^{N_r}\sum_{k=0}^{N_x/2}\psi_{D,(p,k_l)}U_\infty\left(\frac{1}{r}\frac{\partial b_p(r^2)}{\partial r}\right)\left(\frac{\partial a_{k_l}(x)}{\partial x}\right)+\sum_{p=0}^{N_r}\sum_{k=0}^{N_x/2}\psi_{D,(p,k_r)}U_\infty\left(\frac{1}{r}\frac{\partial b_p(r^2)}{\partial r}\right)\left(\frac{\partial a_{k_r}(x)}{\partial x}\right)\\
&-\sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2}\psi_{D,(n,j_l)}\psi_{D,(p,k_l)}\left(\frac{1}{r}b_n(r^2)\frac{\partial}{\partial r}\left[\frac{1}{r}\frac{\partial b_p(r^2)}{\partial r}\right]\right)\left(\frac{\partial a_{j_l}(x)}{\partial x}a_{k_l}(x)\right)\\
&-\sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2}\psi_{D,(n,j_l)}\psi_{D,(p,k_r)}\left(\frac{1}{r}b_n(r^2)\frac{\partial}{\partial r}\left[\frac{1}{r}\frac{\partial b_p(r^2)}{\partial r}\right]\right)\left(\frac{\partial a_{j_l}(x)}{\partial x}a_{k_r}(x)\right)\\
&-\sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2}\psi_{D,(n,j_r)}\psi_{D,(p,k_l)}\left(\frac{1}{r}b_n(r^2)\frac{\partial}{\partial r}\left[\frac{1}{r}\frac{\partial b_p(r^2)}{\partial r}\right]\right)\left(\frac{\partial a_{j_r}(x)}{\partial x}a_{k_l}(x)\right)\\
&-\sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2}\psi_{D,(n,j_r)}\psi_{D,(p,k_r)}\left(\frac{1}{r}b_n(r^2)\frac{\partial}{\partial r}\left[\frac{1}{r}\frac{\partial b_p(r^2)}{\partial r}\right]\right)\left(\frac{\partial a_{j_r}(x)}{\partial x}a_{k_r}(x)\right)
\end{aligned}
\tag{3.25}
$$

This is followed by applying the Galerkin projection onto basis functions $a_{i_r}(x)$ and $b_m(r^2)$, which results in Eq. (3.26). Furthermore, no integration by parts is required for the non-linear advection term as only first-order derivatives are involved.

$$
\begin{aligned}
\int_{X_l}^{X_r}\int_{0}^{R_{max}} LHS\ r\mathrm{d}r\mathrm{d}x = & \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2} \psi_{D,(n,j_l)}\psi_{D,(p,k_l)} \underbrace{\int_0^{R_{max}} \left(b_m(r^2)\frac{\partial b_n(r^2)}{\partial r}\frac{\partial b_p(r^2)}{\partial r}\right)\frac{1}{r}\mathrm{d}r}_{\text{Tensor A}} \underbrace{\int_{X_l}^{X_r}\left(a_{i_r}(x)a_{j_l}(x)\frac{\partial a_{k_l}(x)}{\partial x}\right)\mathrm{d}x}_{\text{Tensor C}_{l,l}} \\
& + \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2} \psi_{D,(n,j_l)}\psi_{D,(p,k_r)} \underbrace{\int_0^{R_{max}} \left(b_m(r^2)\frac{\partial b_n(r^2)}{\partial r}\frac{\partial b_p(r^2)}{\partial r}\right)\frac{1}{r}\mathrm{d}r}_{\text{Tensor A}} \underbrace{\int_{X_l}^{X_r}\left(a_{i_r}(x)a_{j_l}(x)\frac{\partial a_{k_r}(x)}{\partial x}\right)\mathrm{d}x}_{\text{Tensor C}_{l,r}} \\
& + \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2} \psi_{D,(n,j_r)}\psi_{D,(p,k_l)} \underbrace{\int_0^{R_{max}} \left(b_m(r^2)\frac{\partial b_n(r^2)}{\partial r}\frac{\partial b_p(r^2)}{\partial r}\right)\frac{1}{r}\mathrm{d}r}_{\text{Tensor A}} \underbrace{\int_{X_l}^{X_r}\left(a_{i_r}(x)a_{j_r}(x)\frac{\partial a_{k_l}(x)}{\partial x}\right)\mathrm{d}x}_{\text{Tensor C}_{r,l}} \\
& + \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2} \psi_{D,(n,j_r)}\psi_{D,(p,k_r)} \underbrace{\int_0^{R_{max}} \left(b_m(r^2)\frac{\partial b_n(r^2)}{\partial r}\frac{\partial b_p(r^2)}{\partial r}\right)\frac{1}{r}\mathrm{d}r}_{\text{Tensor A}} \underbrace{\int_{X_l}^{X_r}\left(a_{i_r}(x)a_{j_r}(x)\frac{\partial a_{k_r}(x)}{\partial x}\right)\mathrm{d}x}_{\text{Tensor C}_{r,r}} \\
& + \sum_{p=0}^{N_r}\sum_{k=0}^{N_x/2} \psi_{D,(p,k_l)}U_\infty \underbrace{\int_0^{R_{max}} \left(b_m(r^2)\frac{\partial b_p(r^2)}{\partial r}\right)\mathrm{d}r}_{\text{Matrix C}} \underbrace{\int_{X_l}^{X_r}\left(a_{i_r}(x)\frac{\partial a_{k_l}(x)}{\partial x}\right)\mathrm{d}x}_{\text{Matrix D}_l} \\
& + \sum_{p=0}^{N_r}\sum_{k=0}^{N_x/2} \psi_{D,(p,k_r)}U_\infty \underbrace{\int_0^{R_{max}} \left(b_m(r^2)\frac{\partial b_p(r^2)}{\partial r}\right)\mathrm{d}r}_{\text{Matrix C}} \underbrace{\int_{X_l}^{X_r}\left(a_{i_r}(x)\frac{\partial a_{k_r}(x)}{\partial x}\right)\mathrm{d}x}_{\text{Matrix D}_r} \\
& - \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2} \psi_{D,(n,j_l)}\psi_{D,(p,k_l)} \underbrace{\int_0^{R_{max}} \left(b_m(r^2)b_n(r^2)\frac{\partial}{\partial r}\left[\frac{1}{r}\frac{\partial b_p(r^2)}{\partial r}\right]\right)\mathrm{d}r}_{\text{Tensor B}} \underbrace{\int_{X_l}^{X_r}\left(a_{i_r}(x)\frac{\partial a_{j_l}(x)}{\partial x}a_{k_l}(x)\right)\mathrm{d}x}_{\text{Tensor B}_{l,l}} \\
& - \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2} \psi_{D,(n,j_l)}\psi_{D,(p,k_r)} \underbrace{\int_0^{R_{max}} \left(b_m(r^2)b_n(r^2)\frac{\partial}{\partial r}\left[\frac{1}{r}\frac{\partial b_p(r^2)}{\partial r}\right]\right)\mathrm{d}r}_{\text{Tensor B}} \underbrace{\int_{X_l}^{X_r}\left(a_{i_r}(x)\frac{\partial a_{j_l}(x)}{\partial x}a_{k_r}(x)\right)\mathrm{d}x}_{\text{Tensor B}_{l,r}} \\
& - \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2} \psi_{D,(n,j_r)}\psi_{D,(p,k_l)} \underbrace{\int_0^{R_{max}} \left(b_m(r^2)b_n(r^2)\frac{\partial}{\partial r}\left[\frac{1}{r}\frac{\partial b_p(r^2)}{\partial r}\right]\right)\mathrm{d}r}_{\text{Tensor B}} \underbrace{\int_{X_l}^{X_r}\left(a_{i_r}(x)\frac{\partial a_{j_r}(x)}{\partial x}a_{k_l}(x)\right)\mathrm{d}x}_{\text{Tensor B}_{r,l}} \\
& - \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2} \psi_{D,(n,j_r)}\psi_{D,(p,k_r)} \underbrace{\int_0^{R_{max}} \left(b_m(r^2)b_n(r^2)\frac{\partial}{\partial r}\left[\frac{1}{r}\frac{\partial b_p(r^2)}{\partial r}\right]\right)\mathrm{d}r}_{\text{Tensor B}} \underbrace{\int_{X_l}^{X_r}\left(a_{i_r}(x)\frac{\partial a_{j_r}(x)}{\partial x}a_{k_r}(x)\right)\mathrm{d}x}_{\text{Tensor B}_{r,r}}
\end{aligned}
\tag{3.26}
$$

The non-linear advection term expressed in Eq. (3.26) requires further adjustments due to a discrepancy between the radial basis functions, which are formulated in terms of r^2, while the derivatives and integrals with respect to the radial direction are a function of r. Hence, a similar process to that employed for the diffusion term will be undertaken to ensure consistency.

The first terms that will be worked out are Tensor A and Tensor B, using the relations presented in Eqs. (3.15) and (3.16). The resulting tensor expressions are presented in Eqs. (3.27) and (3.28) respectively.

$$
\begin{aligned}
\int_0^{R_{max}} \left(b_m(r^2) \frac{\partial b_n(r^2)}{\partial r} \frac{\partial b_p(r^2)}{\partial r} \right) \frac{1}{r} \mathrm{d}r &= \int_0^{R_{max}} \left(b_m(s) \frac{\partial b_n(s)}{\partial r} \frac{\partial b_p(s)}{\partial r} \right) \frac{1}{\sqrt{s}} \mathrm{d}r \\
&= \int_0^{R_{max}^2} \left(b_m(s) \left(\frac{\partial b_n(s)}{\partial s} 2\sqrt{s} \right) \left(\frac{\partial b_p(s)}{\partial s} 2\sqrt{s} \right) \right) \frac{1}{\sqrt{s}} \frac{1}{2\sqrt{s}} \mathrm{d}s \\
&= \int_0^{R_{max}^2} 2b_m(s) \frac{\partial b_n(s)}{\partial s} \frac{\partial b_p(s)}{\partial s} \mathrm{d}s
\end{aligned}
\tag{3.27}
$$

$$
\begin{aligned}
\int_0^{R_{max}} \left(b_m(r^2) b_n(r^2) \frac{\partial}{\partial r} \left[\frac{1}{r} \frac{\partial b_p(r^2)}{\partial r} \right] \right) \mathrm{d}r &= \int_0^{R_{max}} \left(b_m(s) b_n(s) \frac{\partial}{\partial r} \left[\frac{1}{\sqrt{s}} \frac{\partial b_p(s)}{\partial r} \right] \right) \mathrm{d}r \\
&= \int_0^{R_{max}^2} \left(b_m(s) b_n(s) \frac{\partial}{\partial r} \left[\frac{1}{\sqrt{s}} \frac{\partial b_p(s)}{\partial s} 2\sqrt{s} \right] \right) \frac{1}{2\sqrt{s}} \mathrm{d}s \\
&= \int_0^{R_{max}^2} \left(b_m(s) b_n(s) \frac{\partial^2 b_p(s)}{\partial s^2} 4\sqrt{s} \right) \frac{1}{2\sqrt{s}} \mathrm{d}s \\
&= \int_0^{R_{max}^2} 2b_m(s) b_n(s) \frac{\partial^2 b_p(s)}{\partial s^2} \mathrm{d}s
\end{aligned}
\tag{3.28}
$$

Lastly, Matrix C will be worked out, resulting in the expression provided by Eq. (3.29).

$$
\int_0^{R_{max}} \left(b_m(r^2) \frac{\partial b_p(r^2)}{\partial r} \right) \mathrm{d}r = \int_0^{R_{max}} \left(b_m(s) \frac{\partial b_p(s)}{\partial r} \right) \mathrm{d}r = \int_0^{R_{max}^2} \left(b_m(s) \frac{\partial b_p(s)}{\partial s} 2\sqrt{s} \right) \frac{1}{2\sqrt{s}} \mathrm{d}s = \int_0^{R_{max}^2} b_m(s) \frac{\partial b_p(s)}{\partial s} \mathrm{d}s
\tag{3.29}
$$

Implementing the new expressions for Tensor A, Tensor B and Matrix C in Eq. (3.30) results in Eq. (3.30). The final expression of the non-linear advection term reveals two distinct components: a linear term dependent on the incoming flow velocity, U_∞, and non-linear terms characterised by triple product integrals.

$$
\begin{aligned}
\int\limits_{X_l}^{X_r}\int\limits_{0}^{R_{max}} LHS\ r\mathrm{d}r\mathrm{d}x = {} & \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2}\psi_{D,(n,j_l)}\psi_{D,(p,k_l)} \underbrace{\int\limits_{0}^{R^2_{max}} 2b_m(s)\frac{\partial b_n(s)}{\partial s}\frac{\partial b_p(s)}{\partial s}\mathrm{d}s}_{\text{Tensor A}} \underbrace{\int\limits_{X_l}^{X_r}\left(a_{i_r}(x)a_{j_l}(x)\frac{\partial a_{k_l}(x)}{\partial x}\right)\mathrm{d}x}_{\text{Tensor C}_{l,l}} \\
& + \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2}\psi_{D,(n,j_l)}\psi_{D,(p,k_r)} \underbrace{\int\limits_{0}^{R^2_{max}} 2b_m(s)\frac{\partial b_n(s)}{\partial s}\frac{\partial b_p(s)}{\partial s}\mathrm{d}s}_{\text{Tensor A}} \underbrace{\int\limits_{X_l}^{X_r}\left(a_{i_r}(x)a_{j_l}(x)\frac{\partial a_{k_r}(x)}{\partial x}\right)\mathrm{d}x}_{\text{Tensor C}_{l,r}} \\
& + \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2}\psi_{D,(n,j_r)}\psi_{D,(p,k_l)} \underbrace{\int\limits_{0}^{R^2_{max}} 2b_m(s)\frac{\partial b_n(s)}{\partial s}\frac{\partial b_p(s)}{\partial s}\mathrm{d}s}_{\text{Tensor A}} \underbrace{\int\limits_{X_l}^{X_r}\left(a_{i_r}(x)a_{j_r}(x)\frac{\partial a_{k_l}(x)}{\partial x}\right)\mathrm{d}x}_{\text{Tensor C}_{r,l}} \\
& + \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2}\psi_{D,(n,j_r)}\psi_{D,(p,k_r)} \underbrace{\int\limits_{0}^{R^2_{max}} 2b_m(s)\frac{\partial b_n(s)}{\partial s}\frac{\partial b_p(s)}{\partial s}\mathrm{d}s}_{\text{Tensor A}} \underbrace{\int\limits_{X_l}^{X_r}\left(a_{i_r}(x)a_{j_r}(x)\frac{\partial a_{k_r}(x)}{\partial x}\right)\mathrm{d}x}_{\text{Tensor C}_{r,r}} \\
& + \sum_{p=0}^{N_r}\sum_{k=0}^{N_x/2}\psi_{D,(p,k_l)}U_\infty \underbrace{\int\limits_{0}^{R^2_{max}} b_m(s)\frac{\partial b_p(s)}{\partial s}\mathrm{d}s}_{\text{Matrix C}} \underbrace{\int\limits_{X_l}^{X_r}\left(a_{i_r}(x)\frac{\partial a_{k_l}(x)}{\partial x}\right)\mathrm{d}x}_{\text{Matrix D}_l} \\
& + \sum_{p=0}^{N_r}\sum_{k=0}^{N_x/2}\psi_{D,(p,k_r)}U_\infty \underbrace{\int\limits_{0}^{R^2_{max}} b_m(s)\frac{\partial b_p(s)}{\partial s}\mathrm{d}s}_{\text{Matrix C}} \underbrace{\int\limits_{X_l}^{X_r}\left(a_{i_r}(x)\frac{\partial a_{k_r}(x)}{\partial x}\right)\mathrm{d}x}_{\text{Matrix D}_r} \\
& - \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2}\psi_{D,(n,j_l)}\psi_{D,(p,k_l)} \underbrace{\int\limits_{0}^{R^2_{max}} 2b_m(s)b_n(s)\frac{\partial^2 b_p(s)}{\partial s^2}\mathrm{d}s}_{\text{Tensor B}} \underbrace{\int\limits_{X_l}^{X_r}\left(a_{i_r}(x)\frac{\partial a_{j_l}(x)}{\partial x}a_{k_l}(x)\right)\mathrm{d}x}_{\text{Tensor B}_{l,l}} \\
& - \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2}\psi_{D,(n,j_l)}\psi_{D,(p,k_r)} \underbrace{\int\limits_{0}^{R^2_{max}} 2b_m(s)b_n(s)\frac{\partial^2 b_p(s)}{\partial s^2}\mathrm{d}s}_{\text{Tensor B}} \underbrace{\int\limits_{X_l}^{X_r}\left(a_{i_r}(x)\frac{\partial a_{j_l}(x)}{\partial x}a_{k_r}(x)\right)\mathrm{d}x}_{\text{Tensor B}_{l,r}} \\
& - \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2}\psi_{D,(n,j_r)}\psi_{D,(p,k_l)} \underbrace{\int\limits_{0}^{R^2_{max}} 2b_m(s)b_n(s)\frac{\partial^2 b_p(s)}{\partial s^2}\mathrm{d}s}_{\text{Tensor B}} \underbrace{\int\limits_{X_l}^{X_r}\left(a_{i_r}(x)\frac{\partial a_{j_r}(x)}{\partial x}a_{k_l}(x)\right)\mathrm{d}x}_{\text{Tensor B}_{r,l}} \\
& - \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2}\psi_{D,(n,j_r)}\psi_{D,(p,k_r)} \underbrace{\int\limits_{0}^{R^2_{max}} 2b_m(s)b_n(s)\frac{\partial^2 b_p(s)}{\partial s^2}\mathrm{d}s}_{\text{Tensor B}} \underbrace{\int\limits_{X_l}^{X_r}\left(a_{i_r}(x)\frac{\partial a_{j_r}(x)}{\partial x}a_{k_r}(x)\right)\mathrm{d}x}_{\text{Tensor B}_{r,r}}
\end{aligned}
\tag{3.30}
$$

3.3. Implementation Velocity Component Approach

In this section, first, the continuity equation for the velocity component approach will be discretised in Subsection 3.3.1. Subsequently, a comprehensive discussion on the discretisation of the momentum equation for the velocity component approach using the FEM will be presented in Subsection 3.3.2.

3.3.1. Implementation Continuity Equation

The starting point for this section is again the two-dimensional, time-averaged, incompressible continuity equation, as outlined in Eq. (3.3). Before performing the discretisation of the continuity equation, a decision needs to be made regarding how to define the radial velocity components. Two possible options exist for this purpose. One option involves separately defining the axial and radial velocity components, while the other option is to define the radial velocity in terms of the axial velocity as derived from the continuity equation. The latter approach offers the benefit of reducing the number of equations, resulting in lower computational costs. Additionally, this method also ensures local mass conservation, satisfying the continuity equation in a strong sense, similar to the stream function approach. However, the aforementioned benefit is offset by the drawback of the resulting dense matrices. In contrast, the stream function approach and the former method retain sparsity. Moreover, another drawback is the necessity for higher-order basis functions, due to a reduction in the order of the radial velocity in the axial direction. Considering that the stream function approach shares similar advantages and disadvantages with the method where the radial velocity component is defined in terms of the axial velocity component, except for the significant advantage of sparsity, the latter approach will therefore not be considered here. Instead, only the approach involving separate velocity components will be discussed in detail.

For the discretisation of Eq. (3.3) the axial and radial velocity components are expressed as a linear combination of axial and radial basis functions, as presented in Eq. (3.31) and Eq. (3.32) respectively.

$$U(r,x) = \sum_{n=0}^{N_r}\sum_{k=0}^{N_x} U_{(n,k)} b_n(r) a_k(x) \tag{3.31}$$

$$V(r,x) = \sum_{n=0}^{N_r}\sum_{k=0}^{N_x} V_{(n,k)} b_n(r) a_k(x) \tag{3.32}$$

Furthermore, similar to the stream function approach, when implementing a marching scheme in the axial direction, the dependency of velocity components along the axial direction can be partitioned into two components: a known component and an unknown component. Coefficients corresponding to known velocity values are associated with the left side of the elements (indicated by the subscript l), while coefficients associated with unknown values are assigned to the right side of the elements (indicated by the subscript r). This notation remains consistent for the basis functions forming velocity values on both the left and right sides. Taking this into account, the axial and radial velocity components in Eq. (3.31) and Eq. (3.32) can be rewritten as given in Eq. (3.33), and Eq. (3.34) respectively.

$$U(r,x) = \sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2} U_{(n,k_l)} b_n(r) a_{k_l}(x) + U_{(n,k_r)} b_n(r) a_{k_r}(x) \tag{3.33}$$

$$V(r,x) = \sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2} V_{(n,k_l)} b_n(r) a_{k_l}(x) + V_{(n,k_r)} b_n(r) a_{k_r}(x) \tag{3.34}$$

Implementing these expressions for the velocity components into the continuity equation results in Eq. (3.35).

$$\begin{aligned} LHS = &\sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2} U_{(n,k_l)} b_n(r) \frac{\partial a_{k_l}(x)}{\partial x} + \sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2} U_{(n,k_r)} b_n(r) \frac{\partial a_{k_r}(x)}{\partial x} \\ &+ \sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2} V_{(n,k_l)} \frac{1}{r}\frac{\partial}{\partial r}\left(r b_n(r)\right) a_{k_l}(x) + \sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2} V_{(n,k_r)} \frac{1}{r}\frac{\partial}{\partial r}\left(r b_n(r)\right) a_{k_r}(x) = 0 \end{aligned} \tag{3.35}$$

Subsequently, the Galerkin projection onto the axial and radial basis functions $a_i(x)$ and $b_m(r)$, respectively, is implemented, using a marching scheme approach, resulting in Eq. (3.36). In the context of using a marching scheme approach, projection onto the right axial basis functions, denoted by $a_{i_r}(x)$, is employed to ensure the formation of square matrices.

$$\begin{aligned} \int_{X_l}^{X_r}\int_0^{R_{max}} LHS \; r\mathrm{d}r\mathrm{d}x = &\sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2} U_{(n,k_l)} \int_0^{R_{max}} b_m(r) b_n(r) r\mathrm{d}r \int_{X_l}^{X_r} a_{i_r}(x) \frac{\partial a_{k_l}(x)}{\partial x}\mathrm{d}x \\ &+ \sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2} U_{(n,k_r)} \int_0^{R_{max}} b_m(r) b_n(r) r\mathrm{d}r \int_{X_l}^{X_r} a_{i_r}(x) \frac{\partial a_{k_r}(x)}{\partial x}\mathrm{d}x \\ &+ \sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2} V_{(n,k_l)} \int_0^{R_{max}} b_m(r) \frac{\partial}{\partial r}\left(r b_n(r)\right)\mathrm{d}r \int_{X_l}^{X_r} a_{i_r}(x) a_{k_l}(x)\mathrm{d}x \\ &+ \sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2} V_{(n,k_r)} \int_0^{R_{max}} b_m(r) \frac{\partial}{\partial r}\left(r b_n(r)\right)\mathrm{d}r \int_{X_l}^{X_r} a_{i_r}(x) a_{k_r}(x)\mathrm{d}x = 0 \end{aligned} \tag{3.36}$$

The last step involves a slight adjustment to the term $\frac{\partial}{\partial r}\left(r b_n(r)\right)$ to avoid differentiating a product of terms, which results in expression Eq. (3.37).

$$\begin{aligned} \int_{X_l}^{X_r}\int_0^{R_{max}} LHS \; r\mathrm{d}r\mathrm{d}x = &\sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2} U_{(n,k_l)} \int_0^{R_{max}} b_m(r) b_n(r) r\mathrm{d}r \int_{X_l}^{X_r} a_{i_r}(x) \frac{\partial a_{k_l}(x)}{\partial x}\mathrm{d}x \\ &+ \sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2} U_{(n,k_r)} \int_0^{R_{max}} b_m(r) b_n(r) r\mathrm{d}r \int_{X_l}^{X_r} a_{i_r}(x) \frac{\partial a_{k_r}(x)}{\partial x}\mathrm{d}x \\ &+ \sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2} V_{(n,k_l)} \int_0^{R_{max}} \left(b_m(r) b_n(r) + b_m(r) r \frac{\partial b_n(r)}{\partial r}\right)\mathrm{d}r \int_{X_l}^{X_r} a_{i_r}(x) a_{k_l}(x)\mathrm{d}x \\ &+ \sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2} V_{(n,k_r)} \int_0^{R_{max}} \left(b_m(r) b_n(r) + b_m(r) r \frac{\partial b_n(r)}{\partial r}\right)\mathrm{d}r \int_{X_l}^{X_r} a_{i_r}(x) a_{k_r}(x)\mathrm{d}x = 0 \end{aligned} \tag{3.37}$$

3.3.2. Implementation Momentum Equation

In this section, the advection and diffusion terms of the momentum equation presented in Eq. (3.5) will be discretised, using a marching scheme in the axial direction.

3.3.2.1. Diffusion Term

In this section, the diffusion term presented in Eq. (3.38) will be discretised based on a marching scheme approach. For this purpose, the axial velocity component is expressed as a linear combination of axial and radial basis functions, as previously outlined in Eq. (3.33).

$$RHS = \frac{1}{r}\frac{\partial}{\partial r}\left(\nu_t\left(r\frac{\partial U}{\partial r}\right)\right) \tag{3.38}$$

Having established the linear combination for the axial velocity component in the framework of using a marching scheme, the subsequent step entails integrating it into Eq. (3.38), which results in the equation outlined in Eq. (3.39).

$$RHS = \sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2} U_{(n,k_l)}\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial b_n(r)}{\partial r}\right)(\nu_t a_{k_l}(x)) + \sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2} U_{(n,k_r)}\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial b_n(r)}{\partial r}\right)(\nu_t a_{k_r}(x)) \tag{3.39}$$

Applying the Galerkin projection onto the axial and radial basis functions $a_{i_r}(x)$ and $b_m(r)$, respectively, using a marching scheme approach, results in Eq. (3.40).

$$\begin{aligned}\int_{X_l}^{X_r}\int_0^{R_{max}} RHS r\mathrm{d}r\mathrm{d}x =& \sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2} U_{(n,k_l)}\int_0^{R_{max}} b_m(r)\frac{\partial}{\partial r}\left(r\frac{\partial b_n(r)}{\partial r}\right)\mathrm{d}r\int_{X_l}^{X_r}\nu_t a_{i_r}(x)a_{k_l}(x)\mathrm{d}x \\ &+\sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2} U_{(n,k_r)}\int_0^{R_{max}} b_m(r)\frac{\partial}{\partial r}\left(r\frac{\partial b_n(r)}{\partial r}\right)\mathrm{d}r\int_{X_l}^{X_r}\nu_t a_{i_r}(x)a_{k_r}(x)\mathrm{d}x\end{aligned} \tag{3.40}$$

Subsequently, integration by parts is utilised on the second-order derivative terms, which lowers the order of the radial derivatives and also prevents encountering a derivative of a multiplication of dependent terms. Implementing this yields Eq. (3.41).

$$\begin{aligned}\int_{X_l}^{X_r}\int_0^{R_{max}} RHS r\mathrm{d}r\mathrm{d}x =& \sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2} U_{(n,k_l)}\left(\underbrace{b_m(r)r\frac{\partial b_n(r)}{\partial r}\bigg|_0^{R_{max}} - \int_0^{R_{max}}\frac{\partial b_m(r)}{\partial r}\frac{\partial b_n(r)}{\partial r}r\mathrm{d}r}_{\text{Matrix A}}\right)\underbrace{\int_{X_l}^{X_r}\nu_t a_{i_r}(x)a_{k_l}(x)\mathrm{d}x}_{\text{Matrix B}_l} \\ &+\sum_{n=0}^{N_r}\sum_{k=0}^{N_x/2} U_{(n,k_r)}\left(\underbrace{b_m(r)r\frac{\partial b_n(r)}{\partial r}\bigg|_0^{R_{max}} - \int_0^{R_{max}}\frac{\partial b_m(r)}{\partial r}\frac{\partial b_n(r)}{\partial r}r\mathrm{d}r}_{\text{Matrix A}}\right)\underbrace{\int_{X_l}^{X_r}\nu_t a_{i_r}(x)a_{k_r}(x)\mathrm{d}x}_{\text{Matrix B}_r}\end{aligned} \tag{3.41}$$

3.3.2.2. Non-Linear Advection Term

In this section, the non-linear advection term for the velocity component approach as presented in Eq. (3.42) will be discretised. The first step in this process involves implementing the linear combinations for the axial and radial velocity components, as expressed in Eq. (3.33) and Eq. (3.34) respectively, resulting in Eq. (3.43).

$$LHS = U\frac{\partial U}{\partial x} + V\frac{\partial U}{\partial r} \tag{3.42}$$

$$
\begin{aligned}
\int\limits_{0}^{R_{max}} LHS = & \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2} U_{(n,j_l)}U_{(p,k_l)}b_n(r)b_p(r)a_{j_l}(x)\frac{\partial a_{k_l}(x)}{\partial x} + \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2} U_{(n,j_l)}U_{(p,k_r)}b_n(r)b_p(r)a_{j_l}(x)\frac{\partial a_{k_r}(x)}{\partial x} \\
& + \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2} U_{(n,j_r)}U_{(p,k_l)}b_n(r)b_p(r)a_{j_r}(x)\frac{\partial a_{k_l}(x)}{\partial x} + \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2} U_{(n,j_r)}U_{(p,k_r)}b_n(r)b_p(r)a_{j_r}(x)\frac{\partial a_{k_r}(x)}{\partial x} \\
& + \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2} V_{(n,j_l)}U_{(p,k_l)}b_n(r)\frac{\partial b_p(r)}{\partial r}a_{j_l}(x)a_{k_l}(x) + \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2} V_{(n,j_l)}U_{(p,k_r)}b_n(r)\frac{\partial b_p(r)}{\partial r}a_{j_l}(x)a_{k_r}(x) \\
& + \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2} V_{(n,j_r)}U_{(p,k_l)}b_n(r)\frac{\partial b_p(r)}{\partial r}a_{j_r}(x)a_{k_l}(x) + \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2} V_{(n,j_r)}U_{(p,k_r)}b_n(r)\frac{\partial b_p(r)}{\partial r}a_{j_r}(x)a_{k_r}(x)
\end{aligned}
\tag{3.43}
$$

The next step in the finite element procedure is applying the Galerkin projection onto the axial and radial basis functions $a_i(x)$ and $b_m(r)$ respectively. In the context of the marching scheme approach, it is decided to only project onto the right axial basis functions denoted by $a_{i_r}(x)$ to maintain square matrices, leading to the final expression for the non-linear advection term in Eq. (3.44).

$$
\begin{aligned}
\int_{X_l}^{X_r} LHS r\mathrm{d}r\mathrm{d}x = {} & \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2} U_{(n,j_l)}U_{(p,k_l)} \underbrace{\int_{0}^{R_{max}} b_m(r)b_n(r)b_p(r)r\mathrm{d}r}_{\text{Tensor A}} \underbrace{\int_{X_l}^{X_r} a_{i_r}(x)a_{j_l}(x)\frac{\partial a_{k_l}(x)}{\partial x}\mathrm{d}x}_{\text{Tensor C}_{l,l}} \\
& + \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2} U_{(n,j_l)}U_{(p,k_r)} \underbrace{\int_{0}^{R_{max}} b_m(r)b_n(r)b_p(r)r\mathrm{d}r}_{\text{Tensor A}} \underbrace{\int_{X_l}^{X_r} a_{i_r}(x)a_{j_l}(x)\frac{\partial a_{k_r}(x)}{\partial x}\mathrm{d}x}_{\text{Tensor C}_{l,r}} \\
& + \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2} U_{(n,j_r)}U_{(p,k_l)} \underbrace{\int_{0}^{R_{max}} b_m(r)b_n(r)b_p(r)r\mathrm{d}r}_{\text{Tensor A}} \underbrace{\int_{X_l}^{X_r} a_{i_r}(x)a_{j_r}(x)\frac{\partial a_{k_l}(x)}{\partial x}\mathrm{d}x}_{\text{Tensor C}_{r,l}} \\
& + \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2} U_{(n,j_r)}U_{(p,k_r)} \underbrace{\int_{0}^{R_{max}} b_m(r)b_n(r)b_p(r)r\mathrm{d}r}_{\text{Tensor A}} \underbrace{\int_{X_l}^{X_r} a_{i_r}(x)a_{j_r}(x)\frac{\partial a_{k_r}(x)}{\partial x}\mathrm{d}x}_{\text{Tensor C}_{r,r}} \\
& + \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2} V_{(n,j_l)}U_{(p,k_l)} \underbrace{\int_{0}^{R_{max}} b_m(r)b_n(r)\frac{\partial b_p(r)}{\partial r}r\mathrm{d}r}_{\text{Tensor B}} \underbrace{\int_{X_l}^{X_r} a_{i_r}(x)a_{j_l}(x)a_{k_l}(x)\mathrm{d}x}_{\text{Tensor B}_{l,l}} \\
& + \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2} V_{(n,j_l)}U_{(p,k_r)} \underbrace{\int_{0}^{R_{max}} b_m(r)b_n(r)\frac{\partial b_p(r)}{\partial r}r\mathrm{d}r}_{\text{Tensor B}} \underbrace{\int_{X_l}^{X_r} a_{i_r}(x)a_{j_l}(x)a_{k_r}(x)\mathrm{d}x}_{\text{Tensor B}_{l,r}} \\
& + \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2} V_{(n,j_r)}U_{(p,k_l)} \underbrace{\int_{0}^{R_{max}} b_m(r)b_n(r)\frac{\partial b_p(r)}{\partial r}r\mathrm{d}r}_{\text{Tensor B}} \underbrace{\int_{X_l}^{X_r} a_{i_r}(x)a_{j_r}(x)a_{k_l}(x)\mathrm{d}x}_{\text{Tensor B}_{r,l}} \\
& + \sum_{n,p=0}^{N_r}\sum_{j,k=0}^{N_x/2} V_{(n,j_r)}U_{(p,k_r)} \underbrace{\int_{0}^{R_{max}} b_m(r)b_n(r)\frac{\partial b_p(r)}{\partial r}r\mathrm{d}r}_{\text{Tensor B}} \underbrace{\int_{X_l}^{X_r} a_{i_r}(x)a_{j_r}(x)a_{k_r}(x)\mathrm{d}x}_{\text{Tensor B}_{r,r}}
\end{aligned} \tag{3.44}
$$

3.4. Solving Matrix and Tensor Multiplications

The diffusion and advection terms in both the stream function approach and the velocity component approach maintain distinct Galerkin projection matrices and tensors for the axial and radial dependent components. This section discusses the procedure of merging these matrices and tensors to form unified representations. However, it is important to mention that when employing piecewise linear basis functions in the axial direction with a marching scheme, the axial matrices and tensors are reduced to only one value, rendering the subsequent discussion unnecessary. However, the following discussion is provided in case higher-order axial basis functions are desired or a global system solution is sought.

For merging the matrices associated with the radial and axial Galerking projections a Kronecker tensor product is employed. This prioritises the axial indices first, ensuring that only non-zero elements are present along lines adjacent to the matrix diagonal.

Then, to merge the tensors associated with the radial and axial triple Galerkin projections, first, a projection index is selected for both the axial and radial tensors, yielding matrices. Following this, a

Kronecker tensor product is calculated between the matrices for each combination of projections. This process concludes with the generation of a Kronecker tensor product between the tensors.

3.5. Non-Linear Solver

Due to the system's nonlinear characteristics, employing a nonlinear solver is essential. Therefore, the Newton-Raphson method has been selected, which originates from the first-order Taylor series expansion of the governing equation formulated in its fixed-point form. The equation that governs this method, as detailed in Eq. (3.45), depends on the Jacobian matrix, J, to advance to the subsequent iteration step. Moreover, in this context, the index i represents the combined index of the axial and radial basis functions onto which the fixed-point form is projected, beginning from the axial index. The index n, on the other hand, denotes the current iteration step in the iterative process.

$$x_i^{n+1} = x_i^n - J(x_i^n)^{-1} f(x_i^n) \quad \text{with} \quad J(x_i^n) = \frac{\partial f_i(x_i^n)}{\partial x_i^n} \tag{3.45}$$

When the stream function approach is implemented, the Newton-Raphson method is governed by Eq. (3.46), and the Jacobian will resemble the matrix presented in Eq. (3.47). It solely contains the momentum equation. Moreover, N represents the combined number of unknowns, contingent on the order of the axial and radial basis functions.

$$\psi_i^{n+1} = \psi_i^n - J(\psi_i^n)^{-1} f(\psi_i^n) \quad \text{with} \quad J(\psi_i^n) = \frac{\partial f_i(\psi_i^n)}{\partial \psi_i^n} \tag{3.46}$$

$$J(\psi_i^n) = \begin{bmatrix} \frac{\partial f_1}{\partial \psi_1} & \frac{\partial f_1}{\partial \psi_2} & \cdots & \frac{\partial f_1}{\partial \psi_{N-1}} & \frac{\partial f_1}{\partial \psi_N} \\ \frac{\partial f_2}{\partial \psi_1} & \frac{\partial f_2}{\partial \psi_2} & \cdots & \frac{\partial f_2}{\partial \psi_{N-1}} & \frac{\partial f_2}{\partial \psi_N} \\ \vdots & & \ddots & & \vdots \\ \frac{\partial f_{N-1}}{\partial \psi_1} & \frac{\partial f_{N-1}}{\partial \psi_2} & \cdots & \frac{\partial f_{N-1}}{\partial \psi_{N-1}} & \frac{\partial f_{N-1}}{\partial \psi_N} \\ \frac{\partial f_N}{\partial \psi_1} & \frac{\partial f_N}{\partial \psi_2} & \cdots & \frac{\partial f_N}{\partial \psi_{N-1}} & \frac{\partial f_N}{\partial \psi_N} \end{bmatrix}_{[N \times N]} \tag{3.47}$$

In the velocity component approach, there are twice as many unknowns compared to the stream function approach. Hence, both the continuity and momentum equations are incorporated to solve for $2N$ unknowns. In this scenario, the governing equation for the Newton-Raphson method is provided by Eq. (3.48), and the Jacobian matrix will resemble the one presented in Eq. (3.49). Here, f is characterised by two indices, where the first index, either 1 or 2, signifies whether the function corresponds to the continuity equation or the momentum equation, respectively.

$$\psi_i^{n+1} = \psi_i^n - J(\psi_i^n)^{-1} f(\psi_i^n) \quad \text{with} \quad J(\psi_i^n) = \frac{\partial f_i(\psi_i^n)}{\partial \psi_i^n} \tag{3.48}$$

$$J(\psi_i^n) = \begin{bmatrix} \frac{\partial f_{1,1}}{\partial U_1} & \frac{\partial f_{1,1}}{\partial U_2} & \cdots & \frac{\partial f_{1,1}}{\partial U_{N-1}} & \frac{\partial f_{1,1}}{\partial U_N} & \frac{\partial f_{1,1}}{\partial V_1} & \frac{\partial f_{1,1}}{\partial V_2} & \cdots & \frac{\partial f_{1,1}}{\partial V_{N-1}} & \frac{\partial f_{1,1}}{\partial V_N} \\ \frac{\partial f_{1,2}}{\partial U_1} & \frac{\partial f_{1,2}}{\partial U_2} & \cdots & \frac{\partial f_{1,2}}{\partial U_{N-1}} & \frac{\partial f_{1,2}}{\partial U_N} & \frac{\partial f_{1,2}}{\partial V_1} & \frac{\partial f_{1,2}}{\partial V_2} & \cdots & \frac{\partial f_{1,2}}{\partial V_{N-1}} & \frac{\partial f_{1,2}}{\partial V_N} \\ \vdots & & \ddots & & \vdots & \vdots & & \ddots & & \vdots \\ \frac{\partial f_{1,N-1}}{\partial U_1} & \frac{\partial f_{1,N-1}}{\partial U_2} & \cdots & \frac{\partial f_{1,N-1}}{\partial U_{N-1}} & \frac{\partial f_{1,N-1}}{\partial U_N} & \frac{\partial f_{1,N-1}}{\partial V_1} & \frac{\partial f_{1,N-1}}{\partial V_2} & \cdots & \frac{\partial f_{1,N-1}}{\partial V_{N-1}} & \frac{\partial f_{1,N-1}}{\partial V_N} \\ \frac{\partial f_{1,N}}{\partial U_1} & \frac{\partial f_{1,N}}{\partial U_2} & \cdots & \frac{\partial f_{1,N}}{\partial U_{N-1}} & \frac{\partial f_{1,N}}{\partial U_N} & \frac{\partial f_{1,N}}{\partial V_1} & \frac{\partial f_{1,N}}{\partial V_2} & \cdots & \frac{\partial f_{1,N}}{\partial V_{N-1}} & \frac{\partial f_{1,N}}{\partial V_N} \\ \frac{\partial f_{0,1}}{\partial U_1} & \frac{\partial f_{0,1}}{\partial U_2} & \cdots & \frac{\partial f_{0,1}}{\partial U_{N-1}} & \frac{\partial f_{0,1}}{\partial U_N} & \frac{\partial f_{0,1}}{\partial V_1} & \frac{\partial f_{0,1}}{\partial V_2} & \cdots & \frac{\partial f_{0,1}}{\partial V_{N-1}} & \frac{\partial f_{0,1}}{\partial V_N} \\ \frac{\partial f_{0,2}}{\partial U_1} & \frac{\partial f_{0,2}}{\partial U_2} & \cdots & \frac{\partial f_{0,2}}{\partial U_{N-1}} & \frac{\partial f_{0,2}}{\partial U_N} & \frac{\partial f_{0,2}}{\partial V_1} & \frac{\partial f_{0,2}}{\partial V_2} & \cdots & \frac{\partial f_{0,2}}{\partial V_{N-1}} & \frac{\partial f_{0,2}}{\partial V_N} \\ \vdots & & \ddots & & \vdots & \vdots & & \ddots & & \vdots \\ \frac{\partial f_{0,N-1}}{\partial U_1} & \frac{\partial f_{0,N-1}}{\partial U_2} & \cdots & \frac{\partial f_{0,N-1}}{\partial U_{N-1}} & \frac{\partial f_{0,N-1}}{\partial U_N} & \frac{\partial f_{0,N-1}}{\partial V_1} & \frac{\partial f_{0,N-1}}{\partial V_2} & \cdots & \frac{\partial f_{0,N-1}}{\partial V_{N-1}} & \frac{\partial f_{0,N-1}}{\partial V_N} \\ \frac{\partial f_{0,N}}{\partial U_1} & \frac{\partial f_{0,N}}{\partial U_2} & \cdots & \frac{\partial f_{0,N}}{\partial U_{N-1}} & \frac{\partial f_{0,N}}{\partial U_N} & \frac{\partial f_{0,N}}{\partial V_1} & \frac{\partial f_{0,N}}{\partial V_2} & \cdots & \frac{\partial f_{0,N}}{\partial V_{N-1}} & \frac{\partial f_{0,N}}{\partial V_N} \end{bmatrix}_{[2N \times 2N]} \tag{3.49}$$

The ease of dealing with linear terms stems from their constant derivatives, whereas the derivatives of the non-linear terms remain dependent on the stream function coefficients or the coefficients of the velocity components. Finally, the iteration concludes once the norm of $J(\psi_i^n)^{-1} f(\psi_i^n)$ falls below 1e−6.

3.6. Implementation Boundary Conditions

There are two types of boundary conditions to be implemented Dirichlet boundary conditions and Neumann boundary conditions.

Incorporating Dirichlet boundary conditions entails specifying the solution values at particular points or along specific boundaries within the computational domain. One method involves solving solely for the unknown coefficients in the solution vector, followed by the addition of Dirichlet boundary conditions at the appropriate positions in the solution vector. Alternatively, the system of equations can be adjusted to incorporate the Dirichlet boundary conditions without the need to remove equations. This is achieved by modifying the solution vector to include the Dirichlet boundary conditions. Furthermore, to maintain the Dirichlet boundary condition rows at their prescribed values, the corresponding rows in the Newton update vector are enforced to be exactly zero. This is accomplished by setting non-diagonal entries of the relevant row in the Jacobian matrix to zero. Additionally, since the increment for the row corresponding to the Dirichlet boundary condition is zero, it becomes completely independent from all other rows and columns, allowing the corresponding column to be zeroed as well [81].

Then, incorporating Neumann boundary conditions is less straightforward. Depending on the order of the employed basis functions, two distinct approaches are employed. Firstly, when higher-order Hermite interpolation basis functions are utilised, Neumann boundary conditions can be directly applied. This is achieved by setting the coefficient multiplied by the basis function associated with the first derivative at the boundary equal to the required value of the derivative. Then, in the case of piecewise linear basis functions, the Neumann boundary condition, which essentially states that $U_{i,1} = U_{i,0}$, can be employed by enforcing this statement directly in the matrix and tensor formulations.

4

Verification

In this chapter, the implementation strategy is verified. This is a crucial step in the model development process, considering the likelihood of errors slipping into the implementation phase. The verification process consists of several parts. First, each element of the wake model is individually verified, which is followed by an overall verification of the entire model. However, before verifying the different aspects of the wake model, first, the accuracy of the radial discretisation will be discussed in Section 4.1. Additionally, it is worth noting that the verification process is exclusively conducted for the velocity component approach. This choice stems from the similarity between the velocity component approach and the stream function approach, with the velocity component approach requiring the implementation of one additional equation.

The first test involves confirming the accurate implementation of the diffusion term, as described in Section 4.2. This is followed by verifying the correct implementation of the continuity equation in Section 4.3. Next, the implementation of the Newton-Raphson non-linear solver is validated as detailed in Section 4.4. Finally, Section 4.5 discusses the observed issues encountered when implementing the full Ainslie wake model.

4.1. Accuracy Radial Discretisation

This section provides a brief discussion on the accuracy of radial discretization. Before delving into how the order of the Hermite interpolation polynomial basis functions and the number of degrees of freedom affect accuracy in Subsection 4.1.3, the method used to approximate a given function and the error measure used are discussed in Subsection 4.1.1 and Subsection 4.1.2 respectively. A representation of the Hermite basis functions for various orders is shown in Fig. 2.5.

4.1.1. Function Approximation with Basis Functions

The function to be approximated, denoted by $f(x)$, is assumed to be able to be expressed as a summation over the basis functions, $b_n(x)$, multiplied by unknown coefficients, α_n, as illustrated in Eq. (4.1).

$$f(r) \approx \sum_{n=1}^{N} \alpha_n b_n(r) \tag{4.1}$$

To address the non-orthogonality of the Hermite interpolation polynomials and solve for the unknown coefficients, α_n, the subsequent step involves applying the Ritz-Galerkin projection to Eq. (4.1), as shown in Eq. (4.2), with $b_m(r)$ acting as the test function.

$$\underbrace{\int_{\Omega} f(r) b_m(r) r \mathrm{d}r}_{f_m} = \sum_{n=1}^{N} \alpha_n \underbrace{\int_{\Omega} b_n(r) b_m(r) r \mathrm{d}r}_{\text{Global Mass Matrix A}} \tag{4.2}$$

The unknown coefficients are then calculated using Eq. (4.3), such that the function can be approximated using Eq. (4.1).

$$\alpha_n = A \backslash f_m \tag{4.3}$$

4.1.2. Error Metric for Radial Discretisation

The approximation error is evaluated by integrating the squared difference between the approximation and the Gaussian function across the entire domain, as described in Eq. (4.4). This integration is performed using symbolic methods in MATLAB.

$$\int_{\Omega} (f(r) - f_{ref}(r))^2 r \mathrm{d}r \tag{4.4}$$

4.1.3. Results Radial Discretisation Error

The graph shown in Figure 4.1 showcases the dependence of the absolute approximation error on the number of degrees-of-freedom, and the order of basis functions when approximating a Gaussian profile. Significantly, the graph reveals a pattern where the reduction in approximation error becomes more significant with higher orders of basis functions. Specifically, the slopes indicating the decrease in approximation error increase with the order of basis functions, aligning with the anticipated decrease, indicated by the dotted black lines. This finding suggests that when employing higher-order basis functions, achieving the same level of approximation error requires fewer degrees of freedom. In other words, the increased complexity introduced by higher-order basis functions enhances the efficiency of the system in achieving the desired approximation with a reduced number of degrees of freedom. Despite the rapid increase in computational time associated with higher-order basis functions, the reduced need for elements or degrees of freedom counterbalances this effect and can even result in lower computational time.

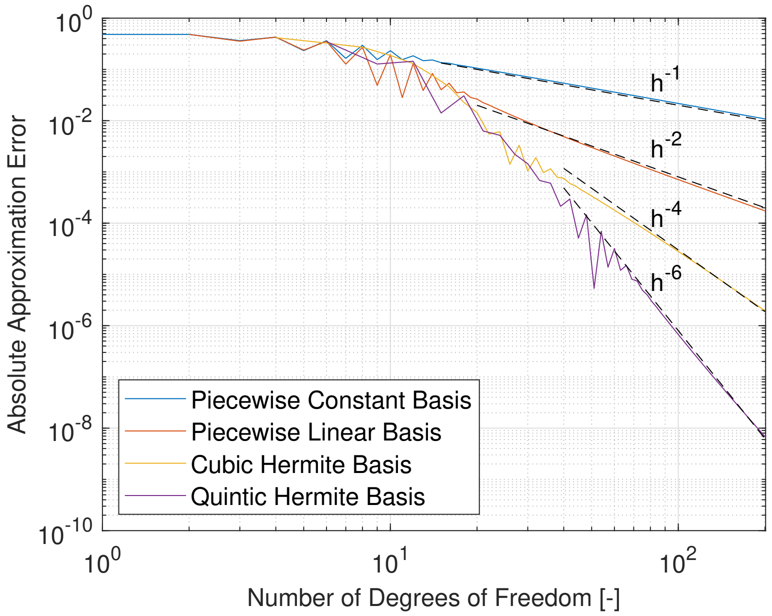


Figure 4.1: Absolute approximation errors for the Gaussian based on the number of degrees of freedom and the choice of basis functions.

4.2. Verification of Diffusion Term

The first term to be verified is the diffusion term within the momentum equation. The main objective of this test is to confirm the mass conservation characteristic of the diffusion equation.

The equation that will be used for this test case is provided in Eq. (4.5), and is closely related to the heat transfer equation. However, since the axial velocity is assumed to be independent of time, the left-hand side of Eq. (4.5) needs to be rewritten. For this purpose, one can either use a finite difference method or one can rewrite the equation in terms of the change in axial velocity in the downstream direction, as derived in Eq. (4.6). Additionally, Eq. (4.6) can also be obtained from linearising the advection terms in the Ainslie wake model, as outlined in Eq. (4.7).

$$\frac{\partial U}{\partial t} = \nu_t \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial U}{\partial r}\right) \tag{4.5}$$

$$\frac{\partial U}{\partial t} = \frac{\partial U}{\partial t}\frac{\partial x}{\partial x} = \frac{\partial U}{\partial x}U_\infty = \nu_t \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial U}{\partial r}\right) \tag{4.6}$$

$$U\frac{\partial U}{\partial x} + V\frac{\partial U}{\partial r} \approx U_\infty \frac{\partial U}{\partial x} \tag{4.7}$$

Considering the fact that the Ainslie wake model equation is dependent on both the axial and radial direction two distinct cases will be examined. In the first case, which is described and discussed in Subsection 4.2.1, the axial dependency is neglected and Eq. (4.5) is employed using the Crank-Nicolson method. This ensures that the verification of the radial basis functions and the integral-preserving characteristic occurs independently, without being influenced by the axial basis functions. In contrast, in the second case, outlined in Subsection 4.2.2, the axial velocity is composed of contributions from both axial and radial basis functions and Eq. (4.6) is employed to study the integral-preserving characteristic of the equation when having both radial and axial dependencies included.

4.2.1. One-Dimensional Case

For the one-dimensional case, the velocity is assumed to be solely a function of the radial dimension. Based on this assumption the velocity can be expressed as a linear combination of radial basis function, as presented in Eq. (4.8), which allows for the verification of the radial basis function implementation.

$$U = \sum_{n=0}^{N_r} U_n b_n(r) \tag{4.8}$$

Applying the Crank-Nicolson method to Eq. (4.5) and implementing Eq. (4.8), results in Eq. (4.9), where the superscripts i and $i+1$ refer to the known and unknown velocity coefficients respectively. The Crank-Nicolson finite difference scheme is chosen for time discretisation due to its simplicity and stability. Additionally, this allows the equation to be handled as a one-dimensional problem with discrete time steps, rather than in continuous time.

$$\begin{aligned} \frac{U_n^{i+1}b_n(r) - U_n^i b_n(r)}{\Delta t} &= \nu_t \frac{1}{2}\left(\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial b_n(r)}{\partial r}\right)U_n^{i+1} + \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial b_n(r)}{\partial r}\right)U_n^i\right) \\ U_n^{i+1}\left[b_n(r) - \frac{1}{2}\nu_t\Delta t\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial b_n(r)}{\partial r}\right)\right] &= U_n^i\left[b_n(r) + \frac{1}{2}\nu_t\Delta t\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial b_n(r)}{\partial r}\right)\right] \end{aligned} \tag{4.9}$$

The next step involves applying the Galerkin projection on the radial projection functions, denoted by $b_m(r)$, which results in Eq. (4.10).

$$\begin{aligned} &U_n^{i+1}\left[\int_0^{R_{max}} b_m(r)b_n(r)r\mathrm{d}r - \frac{1}{2}\nu_t\Delta t\int_0^{R_{max}} b_m(r)\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial b_n(r)}{\partial r}\right)r\mathrm{d}r\right] \\ &= U_n^i\left[\int_0^{R_{max}} b_m(r)b_n(r)r\mathrm{d}r + \frac{1}{2}\nu_t\Delta t\int_0^{R_{max}} b_m(r)\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial b_n(r)}{\partial r}\right)r\mathrm{d}r\right] \end{aligned} \tag{4.10}$$

Lastly, integration by parts is applied to the second order derivative in the diffusion term in Eq. (4.10), resulting in Eq. (4.11).

$$
\begin{aligned}
U_n^{i+1}&\left[\int_0^{R_{max}} b_m(r)b_n(r)r\mathrm{d}r-\frac{1}{2}\nu_t\Delta t\left(b_m(r)r\frac{\partial b_n(r)}{\partial r}\bigg|_0^{R_{max}}-\int_0^{R_{max}}\frac{\partial b_m(r)}{\partial r}\frac{\partial b_n(r)}{\partial r}r\mathrm{d}r\right)\right]\\
&=U_n^{i}\left[\int_0^{R_{max}} b_m(r)b_n(r)r\mathrm{d}r+\frac{1}{2}\nu_t\Delta t\left(b_m(r)r\frac{\partial b_n(r)}{\partial r}\bigg|_0^{R_{max}}-\int_0^{R_{max}}\frac{\partial b_m(r)}{\partial r}\frac{\partial b_n(r)}{\partial r}r\mathrm{d}r\right)\right]
\end{aligned}
\tag{4.11}
$$

Before delving into the discussion of the results of this test case, it important to acknowledge that the time discretisation also has its implications. However, as the focus is on the spatial discretisation, the exact implications of the time discretisation will not be explored here as these fall outside the scope and focus of this section.

After having established the equation for solving the linear diffusion equation, the subsequent step entails defining the error metric. For this purpose the integral-preserving property of the diffusion equation is utilised, relying on the conservation of mass flux. More specifically, the integral under the diffusing curve should remain constant regardless of the number of time steps applied. Based on this property, the definition for the error at time step t, denoted by ϵ_t, is provided in Eq. (4.12), where $\mathcal{I}$, represents the initial preserved integral quantity. Solving the integrals is done using symbolic calculations in Matlab, as the basis functions are also symbolically defined as a function of r.

$$
\begin{gathered}
\sum_{n=0}^{N_r} U_n^{t=0}\int_0^{R_{max}} b_n(r)r\mathrm{d}r=\mathcal{I}=\sum_{n=0}^{N_r} U_n^{t}\int_0^{R_{max}} b_n(r)r\mathrm{d}r\\
\epsilon_t=\left|\sum_{n=0}^{N_r} U_n^{t}\int_0^{R_{max}} b_n(r)r\mathrm{d}r-\mathcal{I}\right|
\end{gathered}
\tag{4.12}
$$

Before assessing the results, the input parameters will be presented. Starting with the initial inflow conditions, only the axial inflow velocity is required for the diffusion equation. For the test, a Gaussian profile is used as inflow profile, as specified in Eq. (4.13), with σ set to a value of 2. Furthermore the constant viscosity, ν_t, and the time step, Δt, are take to be equal to 0.05 m^2/s and 1.0 s respectively. The value for the constant viscosity is determined based on what was deemed as a reasonable value from solving the Ainslie wake model using and already existing finite volume solver, while the value of the time step was determined based on the value used for a similar verification test presented in [9]. Moreover, the outer domain boundary is determined by the standard deviation of the Gaussian inflow profile. To prevent any mass leakage over the outer domain boundary at $r = R_{max}$, the domain must be adequately large. Hence, the domain boundary extends to $R_{max} = 100\sigma$ for this purpose.

$$
U_{in}=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{r}{\sigma}\right)^2}
\tag{4.13}
$$

4.2.2. Two-Dimensional Case

The primary distinction from the one-dimensional case is that the velocity is no longer solely a function of the radial dimension; it now also becomes dependent on the axial dimension. Based on this the velocity is expressed as a linear combination of radial and axial basis function, as shown in Eq. (4.14), which allows the verification of the two-dimensional diffusion term implementation. Furthermore, a marching scheme is utilised for the test case, following the implementation of the wake model. This allows for the velocity components to be divided into known velocity components, associated with the left axial basis functions, and unknown velocity components, associated with the right axial basis functions. Rewritting Eq. (4.14) to fit the marching scheme approach results in Eq. (4.15), where subscripts l, and r indicate that the associated velocity components and basis functions are related to the known and unknown velocity components respectively.

$$U = \sum_{n=0}^{N_r} \sum_{k=0}^{N_x} U_{(n,k)} b_n(r) a_k(x) \tag{4.14}$$

$$U = \sum_{n=0}^{N_r} \sum_{k=0}^{N_x/2} U_{(n,k_l)} b_n(r) a_{k_l}(x) + U_{(n,k_r)} b_n(r) a_{k_r}(x) \tag{4.15}$$

For this test case Eq. (4.6) serves as the foundation for conducting the error analysis. The first step in deriving the linear diffusion equation that is used for the error analysis involves incorporating Eq. (4.15) into Eq. (4.6), yielding Eq. (4.16).

$$\begin{aligned}
&U_\infty \sum_{n=0}^{N_r} \sum_{k=0}^{N_x/2} U_{(n,k_l)} b_n(r) \frac{\partial a_{k_l}(x)}{\partial x} + U_\infty \sum_{n=0}^{N_r} \sum_{k=0}^{N_x/2} U_{(n,k_r)} b_n(r) \frac{\partial a_{k_r}(x)}{\partial x} \\
&= \nu_t \sum_{n=0}^{N_r} \sum_{k=0}^{N_x/2} U_{(n,k_l)} \frac{1}{r}\frac{\partial}{\partial r}\left(r \frac{\partial b_n(r)}{\partial r} \right) a_{k_l}(x) + \nu_t \sum_{n=0}^{N_r} \sum_{k=0}^{N_x/2} U_{(n,k_r)} \frac{1}{r}\frac{\partial}{\partial r}\left(r \frac{\partial b_n(r)}{\partial r} \right) a_{k_r}(x)
\end{aligned} \tag{4.16}$$

Implementing the Ritz-Galerkin method where $b_m(r)$ and $a_{i_r}(x)$ are the radial and axial test functions to project on respectively, results in Eq. (4.17).

$$\begin{aligned}
&U_\infty \sum_{n=0}^{N_r} \sum_{k=0}^{N_x/2} U_{(n,k_l)} \int_0^{R_{max}} b_m(r) b_n(r) r \mathrm{d}r \int_{X_l}^{X_r} a_{i_r}(x) \frac{\partial a_{k_l}(x)}{\partial x} \mathrm{d}x \\
&+ U_\infty \sum_{n=0}^{N_r} \sum_{k=0}^{N_x/2} U_{(n,k_r)} \int_0^{R_{max}} b_m(r) b_n(r) r \mathrm{d}r \int_{X_l}^{X_r} a_{i_r}(x) \frac{\partial a_{k_r}(x)}{\partial x} \\
&= \nu_t \sum_{n=0}^{N_r} \sum_{k=0}^{N_x/2} U_{(n,k_l)} \int_0^{R_{max}} b_m(r) \frac{1}{r}\frac{\partial}{\partial r}\left(r \frac{\partial b_n(r)}{\partial r} \right) r \mathrm{d}r \int_{X_l}^{X_r} a_{i_r}(x) a_{k_l}(x) \mathrm{d}x \\
&+ \nu_t \sum_{n=0}^{N_r} \sum_{k=0}^{N_x/2} U_{(n,k_r)} \int_0^{R_{max}} b_m(r) \frac{1}{r}\frac{\partial}{\partial r}\left(r \frac{\partial b_n(r)}{\partial r} \right) r \mathrm{d}r \int_{X_l}^{X_r} a_{i_r}(x) a_{k_r}(x) \mathrm{d}x
\end{aligned} \tag{4.17}$$

Lastly, the second-order derivatives in the diffusion terms, in Eq. (4.17), are reduced to first-order derivatives by applying integration by parts. Furthermore, the components associated with the unknown velocities are positioned on the right-hand side of the equation, whereas those linked with the known components are situated on the left-hand side. This results in Eq. (4.18).

$$\begin{aligned}
&\sum_{n=0}^{N_r} \sum_{k=0}^{N_x/2} U_{(n,k_r)} \left(U_\infty \int_0^{R_{max}} b_m(r) b_n(r) r \mathrm{d}r \int_{X_l}^{X_r} a_{i_r}(x) \frac{\partial a_{k_r}(x)}{\partial x} \mathrm{d}x \right) \\
&- \sum_{n=0}^{N_r} \sum_{k=0}^{N_x/2} U_{(n,k_r)} \left(\nu_t \left(b_m(r) r \frac{\partial b_n(r)}{\partial r} \Bigg|_0^{R_{max}} - \int_0^{R_{max}} \frac{\partial b_m(r)}{\partial r} \frac{\partial b_n(r)}{\partial r} r \mathrm{d}r \right) \int_{X_l}^{X_r} a_{i_r}(x) a_{k_r}(x) \mathrm{d}x \right) \\
&= - \sum_{n=0}^{N_r} \sum_{k=0}^{N_x/2} U_{(n,k_l)} \left(U_\infty \int_0^{R_{max}} b_m(r) b_n(r) r \mathrm{d}r \int_{X_l}^{X_r} a_{i_r}(x) \frac{\partial a_{k_l}(x)}{\partial x} \right) \\
&+ \sum_{n=0}^{N_r} \sum_{k=0}^{N_x/2} U_{(n,k_l)} \left(\nu_t \left(b_m(r) r \frac{\partial b_n(r)}{\partial r} \Bigg|_0^{R_{max}} - \int_0^{R_{max}} \frac{\partial b_m(r)}{\partial r} \frac{\partial b_n(r)}{\partial r} r \mathrm{d}r \right) \int_{X_l}^{X_r} a_{i_r}(x) a_{k_l}(x) \mathrm{d}x \right)
\end{aligned} \tag{4.18}$$

For the discussion of the results, the same error metric as in the one-dimensional case, as detailed in Eq. (4.12), along with the same parameter values, as detailed in Subsection 4.2.1, are utilised. The only exception is the value of the eddy viscosity, which has been changed to 0.5 m^2/s. The reason for this is to keep the constant, $\frac{\nu_t}{U_\infty}$, in front of the diffusion term unchanged.

Furthermore, the only variables that remain undefined are the step size of the marching step, Δx, which determines the integration bounds over the axial direction, and the free-stream inflow velocity, U_∞. The marching step size is set at 1.0 m, while the free-stream inflow velocity is fixed at 10 m/s. The value for U_∞ was chosen to keep the constant in front of the diffusion term unchanged. Additionally, a wind speed of 10 m/s corresponds to the annual average wind speed for Wind Class I, as defined in the IEC 61400-1 standard [42], which corresponds to the design specifications of the DTU 10 MW wind turbine used as the reference for the full simulations [82]. Then, the marching step size was determined through a visual inspection of the diffusion process. The primary goals were to maintain stability and ensure appropriate levels of diffusion. This involved balancing the step size. Specifically, it needed to be large enough to avoid having almost no diffusion and small enough to prevent numerical instability.

4.2.3. Results and Discussion on Verification Diffusion Term

This section will discuss and compare the results obtained for the one-dimensional and two-dimensional examination cases as outlined in Subsections 4.2.1 and 4.2.2 respectively. It is important to highlight that while the velocity component approach is utilised as the implementation method for verification of the diffusion term, the presentation of results will encompass not only those obtained using cubic Hermite splines but also those utilising quintic Hermite interpolation basis functions, which are of importance for the stream function approach. Additionally, this will offer insights into how the order of the basis functions impacts the results.

In presenting the results, a distinction is made between two different cases based on the radial mesh employed. Both cases use a radial mesh divided into two parts. More specifically the inner part of the mesh, corresponding to the inner part of the domain, is defined by the gradient of the Gaussian inflow profile, while the outer part employs an exponentially increasing mesh, as illustrated in Fig. 4.2, where the inner part extends up to the point where the Gaussian inflow profile falls below the threshold of 10^{-10} m/s. However, in the first case, the inner part comprises 20 radial nodes, while the outer part consists of 10 radial nodes. In contrast, in the second case, the inner part employs 30 nodes, and the outer part utilises 15 nodes. The results for the one-dimensional case when employing a mesh with 30 radial nodes and 45 radial nodes are presented in Fig. 4.3a, while the results from the two-dimensional case employing a mesh with radially 30 and 45 nodes are shown in Fig. 4.3b respectively.

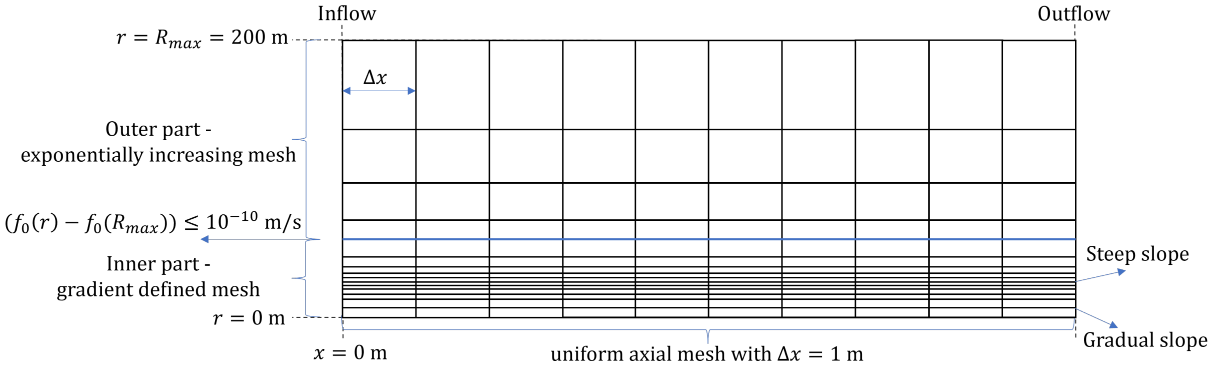
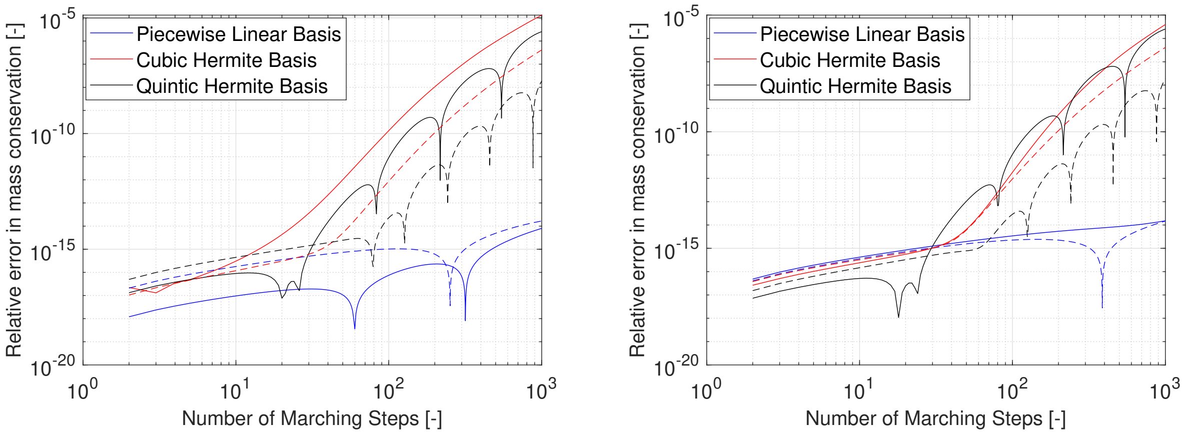


Figure 4.2: Mesh schematic of the initially used mesh.



(a) One-dimensional case.

(b) Two-dimensional case.

Figure 4.3: Relative error in mass conservation for different orders of Hermite interpolation basis functions and different number of radial nodes. The solid line represents results obtained using a mesh with 30 nodes, while the dashed line represents results obtained using a mesh with 45 nodes.

From a visual comparison between the one-dimensional and two-dimensional cases, similar trends can be observed when employing the same number of radial nodes and the same order of basis functions. Based on this observation it can be deduced that the implementation of the axial dimension has minimal influence on mass conservation, suggesting a correct implementation of the second dimension. Furthermore, it should be noted that the downward peaks in error, observed in each of the cases are zero crossings and can be ignored.

Then, comparing the results for the same order of basis functions but for different radial meshes, it can be observed that when employing 30 radial nodes, higher mass conservation errors are evident after 1000 diffusion steps. However, at the initial stages with a low number of diffusion steps, the errors are either lower or similar. From these observations, it can be concluded that a lower number of nodes results in a faster increase in mass conservation error. Moreover, this is closely related to the fact that, in general, the higher-order basis functions demonstrate a quicker divergence from the original slight increase in error compared to the case where a mesh of 45 nodes is utilised.

To determine the root cause of the rapid increase in relative mass conservation error, a more thorough investigation is conducted. This detailed investigation is essential, as such a rapid increase in error is highly undesirable, leading to error propagation when marching downstream and resulting in unreliable outcomes.

However, before delving into the discussion, it is important to address the undesirable observations by taking a step back. In response, piecewise linear basis functions are implemented in the radial direction to solve both the one-dimensional and two-dimensional diffusion equations. It is essential to acknowledge that linear basis functions are restricted to the velocity component approach since the highest-order derivative involved in the stream function approach is not compatible with piecewise linear basis functions. However, even within the framework velocity component approach the use of piecewise linear basis functions poses a challenge due to the discontinuity in the first derivative. This necessitates an approximation for the diffusion term, since after applying integration by parts to solve for the second derivative, the boundary evaluation term involves a first derivative, making it undefined. Hence, the boundary evaluation component of the diffusion term was disregarded.

The results for the one-dimensional and two-dimensional scenarios employing piecewise linear basis functions radially are also illustrated in Figs. 4.3a and 4.3b respectively. From visual inspection, it is evident that in the one-dimensional case, utilising 30 radial nodes leads to a more steep rise in mass conservation error compared to using 45 radial nodes, while for the two-dimensional case, the number of radial nodes employed has no significant consequences. Moreover, a notable contrast with the higher-order basis functions is that when employing piecewise linear basis functions, there is no significant increase in integral error after a certain number of diffusion steps. This indicates that the higher-order basis functions are subjected to certain limitations when it comes to accurately preserving the mass conservation error. Lastly, based on this comparison discarding the boundary evaluation term when employing piecewise linear basis functions, does not seem to affect the mass conservation compared to the cases implementing higher-order basis functions.

The first thing that has been considered to explain the discrepancy when implementing higher-order basis functions is the condition number of the inverted matrix, as high condition numbers might result in high numerical errors. The results presented in Table 4.1 for the one-dimensional case and in Table 4.2 for the two-dimensional case indicate that higher-order basis functions tend to exhibit considerably higher condition numbers, potentially inducing unwanted numerical errors. This could explain why higher-order basis functions display a sudden divergence from their initial slope after a certain number of diffusion steps, leading to elevated errors in mass conservation. Therefore, one might expect that the case employing quintic Hermite interpolation basis functions, which yields the highest condition number, would demonstrate the highest error and the most rapid increase when deviating from the initial slope. However, contrary to this expectation, it is observed that the case employing cubic Hermite interpolation basis functions yields the highest error and that the behaviour when using higher-order basis functions is quite similar. Based on this analysis it can not be concluded that the high condition numbers are the reason for the steep increase in mass conservation error when higher-order basis functions are employed.

	Piecewise linear basis	Cubic Hermite basis	Quintic Hermite basis
Condition number for 30 nodes	1.84e+5	8.84e+9	8.32e+14
Condition number for 45 nodes	2.47e+5	9.47e+9	9.24e+14

Table 4.1: Condition numbers for the one-dimensional case for the various orders of basis functions and meshes with different numbers of nodes.

	Piecewise linear basis	Cubic Hermite basis	Quintic Hermite basis
Condition number for 30 nodes	1.64e+5	6.97e+9	6.37e+14
Condition number for 45 nodes	2.12e+5	7.29e+9	7.00e+14

Table 4.2: Condition numbers for the two-dimensional case for the various orders of basis functions and meshes with different numbers of nodes.

Before proceeding with the discussion and analysis of the observed behavior, the reason behind the high condition numbers observed for high-order basis functions will be elaborated upon.

When expressing the velocity components or the stream function as a linear combination of basis functions and coefficients, the basis functions (represented by ξ) associated with the first- and second-order derivatives in the cubic and quintic cases incorporate scaling factors dependent on the cell size, as illustrated in a general context by Eq. (4.19) and Eq. (4.20), respectively. These scaling factors are incorporated to ensure the continuity across the cell boundaries of the first-order derivative when utilising cubic Hermite basis functions and the continuity over the cell boundaries of the second-order derivative when using quintic Hermite basis functions. Ensuring this continuity across the cell boundaries is essential when employing integration by parts, as without it, the boundary evaluation term becomes undefined. Nevertheless, because these scaling factors are dependent on the mesh and an exponentially increasing mesh is utilised outside the wake region for computational efficiency, a significant gap exists between the lowest and highest values within the matrix, leading to ill-conditioned matrices.

$$f(x) = f(x_l)\xi_0(x) + \left.\frac{\mathrm{d}f}{\mathrm{d}x}\right|_{x=x_l}(x_r - x_l)\xi_1(x) + f(x_r)\xi_2(x) + \left.\frac{\mathrm{d}f}{\mathrm{d}x}\right|_{x=x_r}(x_r - x_l)\xi_3(x) \tag{4.19}$$

$$\begin{aligned} f(x) = f(x_l)\xi_0(x) + \left.\frac{\mathrm{d}f}{\mathrm{d}x}\right|_{x=x_l}(x_r - x_l)\xi_1(x) + \left.\frac{\mathrm{d}^2 f}{\mathrm{d}x^2}\right|_{x=x_l}\frac{(x_r - x_l)^2}{2}\xi_2(x) \\ + f(x_r)\xi_3(x) + \left.\frac{\mathrm{d}f}{\mathrm{d}x}\right|_{x=x_r}(x_r - x_l)\xi_4(x) + \left.\frac{\mathrm{d}^2 f}{\mathrm{d}x^2}\right|_{x=x_r}\frac{(x_r - x_l)^2}{2}\xi_5(x) \end{aligned} \tag{4.20}$$

Subsequently, a detailed examination was conducted on the values at the boundary $r = R_{max}$ to ensure that there is no significant numerical error accumulation at the outer boundary when employing higher-order basis functions due to the high condition numbers observed. It is important to emphasise that this is purely to check for numerical error accumulation, as mass leakage over the boundary is unlikely. This is due to the domain being taken sufficiently large to prevent such occurrences. Additionally, the observed increase occurs only for the higher-order basis functions, while all simulation cases share the same parameters. Therefore, if mass leakage were present, each of the simulations would show a sudden increase in mass-preserving error.

Upon observing the final Gaussian profile after 1000 diffusion steps, it becomes evident that when higher-order basis functions are employed, minor oscillations arise due to the mesh not being tailored to the diffused Gaussian profiles, as illustrated in Fig. 4.4. These oscillations result in both positive and negative values in the tail of the Gaussian profile, leading to a loss of integral preservation and the emergence of significant errors. This phenomenon also clarifies why using a mesh of 30 nodes leads to an earlier departure from the expected shallow slope, as the lower number of nodes results in the earlier onset of these oscillations. However, it is crucial to acknowledge that even in the case where piecewise linear basis functions are employed and no oscillations are observed, there is an undershoot evident where the Gaussian profile becomes flat. Nevertheless, it appears that this undershoot has a less significant impact on the integral preserving characteristic of the diffusion equation.

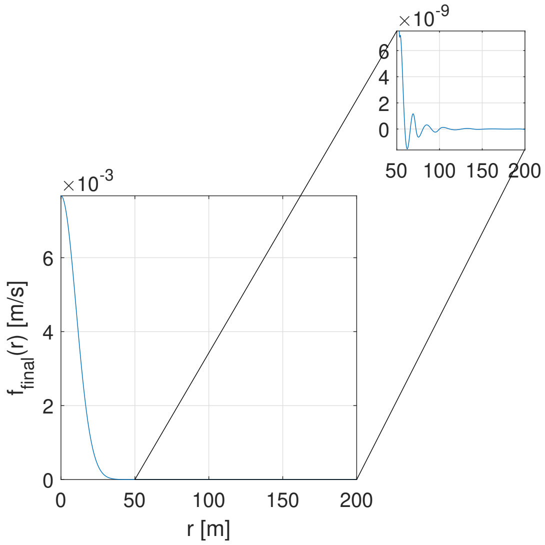


Figure 4.4: Zoomed-in final Gaussian profile illustrating the present oscillations.

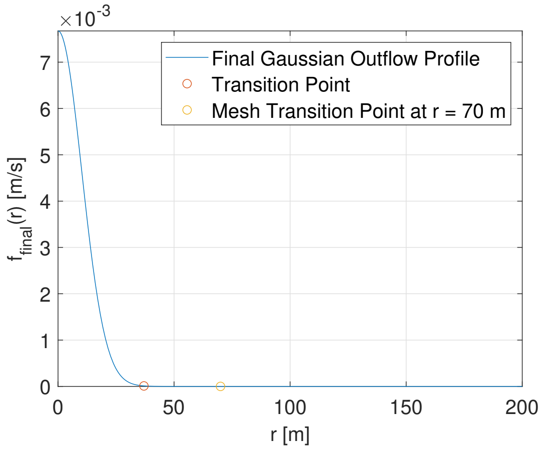


Figure 4.5: Illustration of the Gaussian transition point and mesh transition point.

Based on this analysis, a new radial mesh was developed to accommodate the diffusive nature of the profile. The creation of the mesh involved closely examining the final diffusion profile. It was determined that the mesh needed to be sufficiently fine, extending up to a point slightly beyond the transition point of the final Gaussian profile, where the profile becomes flat, as indicated in Fig. 4.5. Beyond this point, specifically at the radial position where the value of the final Gaussian outflow profile, $f_{final}(r)$, falls below 10^{-10} m/s, an exponentially increasing mesh was reintroduced for computational efficiency. Moreover, for the inner part of the mesh, which extends from $r = 0$ m to $r = 70$ m, a uniform mesh with 100 nodes was used, while the outer part exponentially increased with 5 nodes, as illustrated in Fig. 4.6.

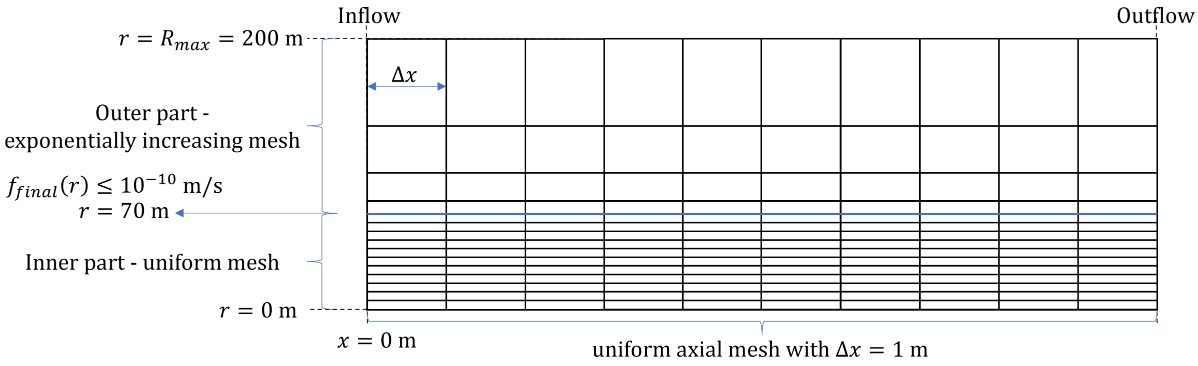
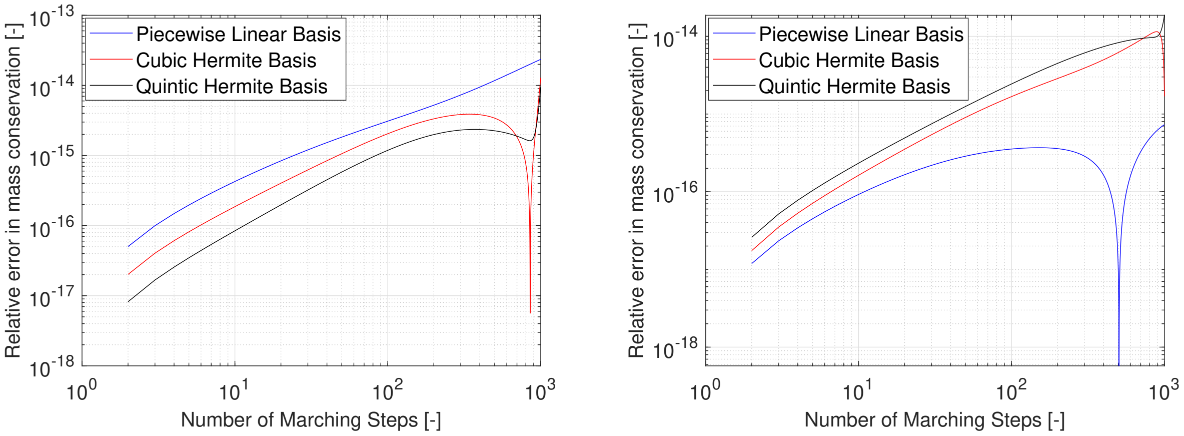


Figure 4.6: Mesh schematic of the adapted mesh.



(a) One-Dimensional case.

(b) Two-dimensional case.

Figure 4.7: Relative error in mass conservation for different orders of Hermite interpolation basis functions utilising a mesh with 105 nodes.

The results obtained from this mesh implementation are presented in Figs. 4.7a and 4.7b for the one-dimensional and two-dimensional cases respectively. Upon visual inspection, it is immediately evident that the cases implementing higher-order basis functions in the radial direction no longer exhibit a sudden, significant increase in mass conservation error. This validates the hypothesis that the mesh was accountable for this behaviour and verifies the correct implementation of the diffusion equation using a higher-order basis function. Additionally, it indicates that a greater number of radial nodes are required to accurately model the diffusion for a higher number of diffusion steps when employing higher-order basis functions. Moreover, piecewise linear basis functions have proven to be more resilient against ill-defined meshes.

The higher mesh sensitivity observed when employing higher-order basis functions has also been discussed in literature [83, 84, 85]. Higher-order basis functions, such as Hermite splines, exhibit greater sensitivity to mesh distortion and poor quality compared to lower-order elements like piecewise linear functions. This increased sensitivity arises because the mapping between physical and computational spaces becomes more complex with higher-order elements. Consequently, poorly shaped or distorted elements can significantly reduce accuracy and cause ill-conditioning of the system matrices. In contrast, piecewise linear basis functions, with their simpler mapping and fewer interpolation requirements, are less affected by coarse or distorted meshes, resulting in greater stability and less accuracy loss on such meshes [83, 84, 85].

Additionally, the condition numbers for each of the six simulations are presented to investigate if the higher-resolution mesh has a significant impact on their order. From comparing Table 4.3 with Tables 4.1 and 4.2, it can be noted that the condition numbers using piecewise linear basis functions are of the same order and even slightly decreased, while for the higher-order basis functions, an increase is observed. However, this increase remains within one order of magnitude, indicating that the condition number is relatively insensitive to the employed mesh.

	Piecewise linear basis	Cubic Hermite basis	Quintic Hermite basis
One-dimensional case	1.76e+5	1.38e+10	1.84e+15
Two-dimensional case	1.58e+5	1.09e+10	1.41e+15

Table 4.3: Condition numbers for different orders of basis functions using a higher resolution mesh.

4.2.4. Limitations Imposed by the Higher-Order Finite Element Method

Based on the verification of the diffusion term and the assessment of the integral-preserving characteristic of a Gaussian function during pure diffusion, several conclusions can be drawn regarding the limitations introduced when employing higher-order basis functions in the radial direction compared to using piecewise linear basis functions. The first limitation relates to the observed condition numbers, while the second limitation involves the sensitivity of the mesh.

In terms of the condition number, it has been illustrated in Subsection 4.2.3, that as the order of the basis functions increases, so does the condition number. While no immediate conclusion about the implication of this observation has been drawn, it remains pertinent. High condition numbers are undesirable and should be mitigated, as they may introduce unforeseen numerical errors, that persist and accumulate as the simulation progresses, when marching further downstream, leading to unacceptable levels of uncertainty in the results. Moreover, extremely high condition numbers can result in instability problems. Furthermore, it has been observed that although the condition number changes with the mesh, it remains within reasonable bounds, keeping the order roughly the same. The latter observation shows that the condition number is relatively insensitive to the employed mesh. It should be noted, however, that in this case, the condition number did not pose any problems since the initial condition is extremely smooth and does not contain higher eigenfunctions of the operator. In the case of a general diffusion problem starting from an arbitrary initial condition, instability would likely occur as the condition numbers are expected become excessively high.

The main limitation that is introduced by employing radially higher-order basis functions concerns the mesh sensitivity. Even though less degrees of freedom are required to accurately represent a function using higher-order basis functions, as shown in Subsection 4.1.1, it has been observed that when dealing with a mesh that is excessively coarse, oscillations may arise, highly impacting the integral-preserving property. In comparison, when using piecewise linear basis functions with an equivalent number of nodes, the undershoots observed in this case have demonstrated greater robustness in terms of mass conservation. In order to determine the exact impact of the mesh onto the mass conservation a detailed mesh sensitivity study should be performed.

Lastly, simulation times are crucial, especially for commercial codes that require a constant balance between computational time and accuracy. Therefore, the simulation times are compared for the different cases. The simulation time is divided into two parts: the time required to project the basis functions onto the inflow profile to obtain the coefficients for the initial condition, and the actual simulation time, which involves generating the matrices and computing the coefficients for each of the 1000 downstream diffusion steps.

	Piecewise linear basis	Cubic Hermite basis	Quintic Hermite basis
Inflow generation time - 30 nodes	12 seconds	46 seconds	1 minute 36 seconds
Inflow generation time - 45 nodes	20 seconds	1 minute 8 seconds	2 minutes 36 seconds
Inflow generation time - 105 nodes	44 seconds	5 minutes 6 seconds	9 minutes 45 seconds
Simulation time - 30 nodes	3 minutes 37 seconds	7 minutes 15 seconds	12 minutes 13 seconds
Simulation time - 45 nodes	5 minutes 6 seconds	9 minutes 12 seconds	18 minutes 49 seconds
Simulation time - 105 nodes	9 minutes 12 seconds	23 minutes 23 seconds	47 minutes 39 seconds

Table 4.4: Computational times for one-dimensional cases.

	Piecewise linear basis	Cubic Hermite basis	Quintic Hermite basis
Inflow generation time - 30 nodes	15 seconds	46 seconds	1 minute 43 seconds
Inflow generation time - 45 nodes	20 seconds	1 minute 10 seconds	2 minutes 33 seconds
Inflow generation time - 105 nodes	46 seconds	5 minutes 4 seconds	9 minutes 45 seconds
Simulation time - 30 nodes	5 minutes 20 seconds	12 minutes 51 seconds	19 minutes 29 seconds
Simulation time - 45 nodes	10 minutes 28 seconds	17 minutes 42 seconds	31 minutes 13 seconds
Simulation time - 105 nodes	19 minutes 35 seconds	51 minutes 47 seconds	1 hour 39 minutes 8 seconds

Table 4.5: Computational times for two-dimensional cases.

The computational times for the one-dimensional and two-dimensional cases are provided in Tables 4.4 and 4.5 respectively. Notably, the computational times for generating inflow coefficients are very similar for both cases, as these coefficients are independent of the axial dimension. However, there is a significant increase in simulation times when implementing the second dimension.

Additionally, the computational times for different orders of basis functions can be compared. This comparison reveals a significant increase in computational time when higher-order basis functions are employed. Combining this increase in computational time with the necessity for a higher resolution mesh for the higher-order basis functions due to mesh sensitivity to achieve desired accuracy makes the higher-order basis functions less suitable for commercial codes.

It should be noted, however, that the model used for running these simulations has not yet been optimised for computational efficiency. Therefore, these computational times can likely be significantly reduced, making higher-order basis functions a feasible option once again, in terms of computational time. Consequently, at this stage, these computational times should also not be used for comparison with other commercial codes.

4.3. Verification Continuity Equation

To verify the accurate implementation of the continuity equation, its mass-conserving property was utilised. Specifically, different inflow and outflow profiles with equivalent integrals were generated. Subsequently, these inflow and outflow velocity profiles were incorporated into the continuity equation, and the results were examined. First, in Subsection 4.3.1, a comprehensive overview will be provided on how the continuity equation is implemented and verified. This is followed by a short discussion on the results in Subsection 4.3.2.

4.3.1. Verification Strategy Continuity Equation

To verify the accurate implementation of the continuity equation, the discretised formulation of the continuity equation provided in Eq. (3.37) is implemented using different inflow and outflow profiles with equivalent integrals. However, the generation of several inflow and outflow profiles for axial and radial velocities can be quite challenging when using cylindrical coordinates, as the radial position influences the values of the integrals. Therefore, the verification strategy is slightly simplified by employing Cartesian coordinates, where x still refers to the streamwise direction, while y refers to the direction along one of the wind turbine blades. This only slightly alters Eq. (3.37), resulting in Eq. (4.21).

$$\begin{aligned}\int_{X_l}^{X_r}\int_{0}^{Y} LHS\ \mathrm{d}y\mathrm{d}x =& \sum_{n=0}^{N_y}\sum_{k=0}^{N_x/2} U_{(n,k_l)} \int_{0}^{Y} b_m(y)b_n(y)\mathrm{d}y \int_{X_l}^{X_r} a_{i_r}(x)\frac{\partial a_{k_l}(x)}{\partial x}\mathrm{d}x \\ &+ \sum_{n=0}^{N_y}\sum_{k=0}^{N_x/2} U_{(n,k_r)} \int_{0}^{Y} b_m(y)b_n(y)\mathrm{d}y \int_{X_l}^{X_r} a_{i_r}(x)\frac{\partial a_{k_r}(x)}{\partial x}\mathrm{d}x \\ &+ \sum_{n=0}^{N_y}\sum_{k=0}^{N_x/2} V_{(n,k_l)} \int_{0}^{Y} \left(b_m(y)b_n(y) + b_m(y)\frac{\partial b_n(y)}{\partial y}\right)\mathrm{d}y \int_{X_l}^{X_r} a_{i_r}(x)a_{k_l}(x)\mathrm{d}x \\ &+ \sum_{n=0}^{N_y}\sum_{k=0}^{N_x/2} V_{(n,k_r)} \int_{0}^{Y} \left(b_m(y)b_n(y) + b_m(y)\frac{\partial b_n(r)}{\partial y}\right)\mathrm{d}y \int_{X_l}^{X_r} a_{i_r}(x)a_{k_r}(x)\mathrm{d}x = 0\end{aligned} \tag{4.21}$$

The mesh employed for this verification test consists of one cell with a width of 1 m in the axial direction, and a uniform mesh with 10 cells also each with a width of 1 m in the radial direction, as presented in Fig. 4.8. It should be noted, however, that this verification test has also been performed using different non-uniform meshes, making sure that the test is mesh-independent. Furthermore, the inflow and outflow conditions for the axial and transverse velocities are determined by defining the values of inflow and outflow velocity coefficients such that the integrals under the curves are preserved. This guarantees an accurate representation of the velocity profiles by the basis functions. For the axial velocity, different inflow and outflow velocity profile pairs were generated for each order of the basis functions. Figs. 4.9 to 4.11 provides three different inflow and outflow pairs defined for each order of the basis functions. Furthermore, since a marching scheme is used to solve for the continuity equation, the transverse velocity profile is only defined within one cell. Therefore, the options for different transverse velocity profiles are more restricted, with the simplest choice being a constant velocity profile along the entire transverse direction.

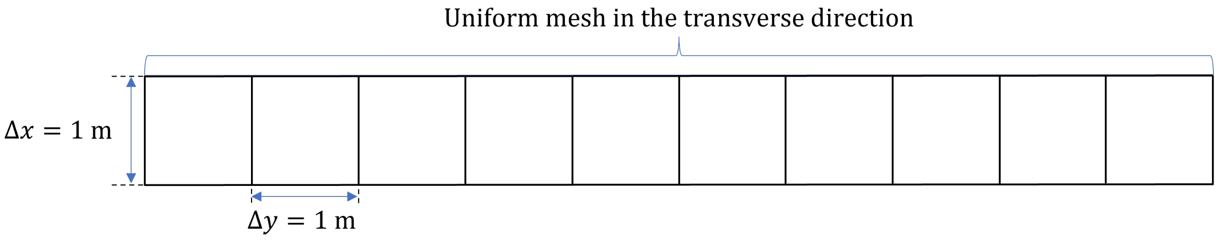
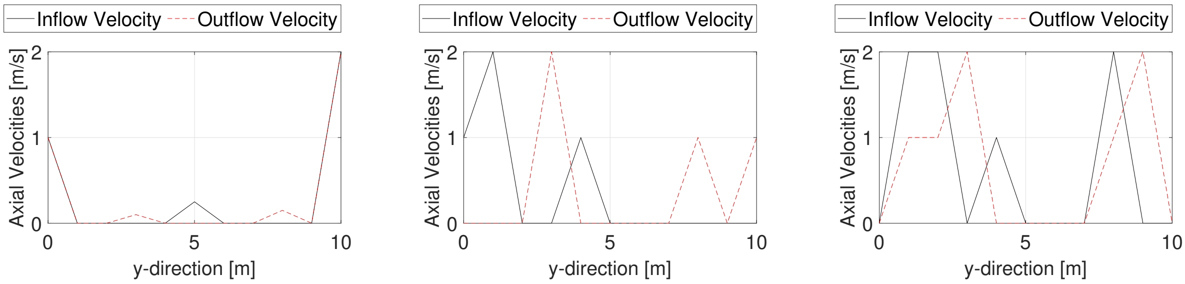


Figure 4.8: Illustration mesh used for verifying the continuity equation implementation

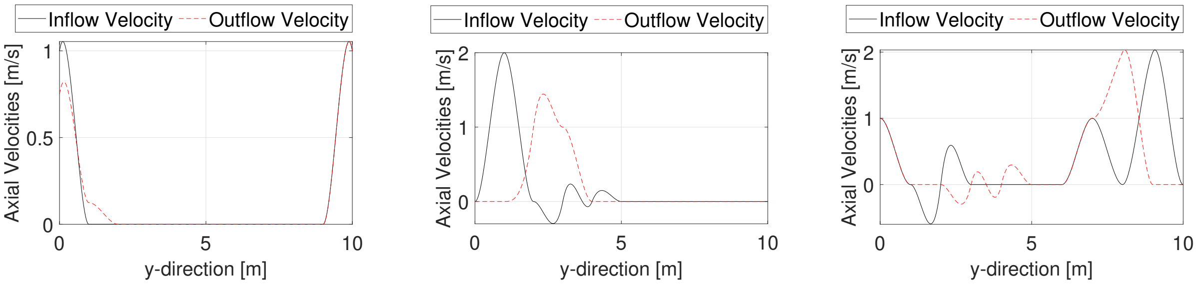


(a) First axial inflow and outflow velocity pair.

(b) Second axial inflow and outflow velocity pair.

(c) Third axial inflow and outflow velocity pair.

Figure 4.9: Various axial integral preserving inflow and outflow velocity pair using piecewise linear basis functions.

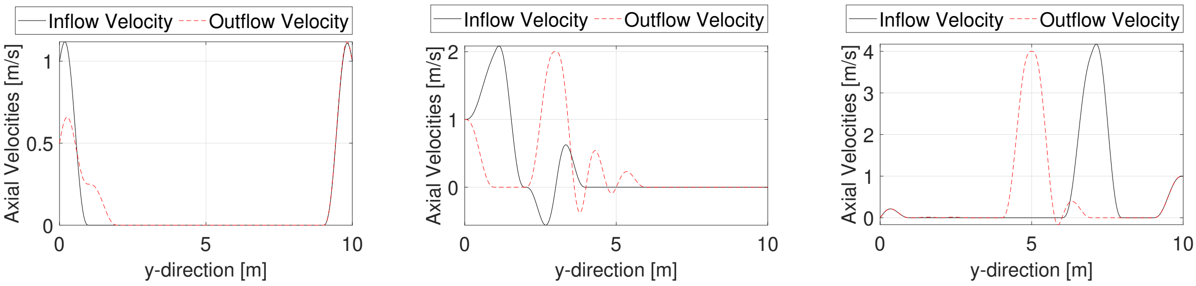


(a) First axial inflow and outflow velocity pair.

(b) Second axial inflow and outflow velocity pair.

(c) Third axial inflow and outflow velocity pair.

Figure 4.10: Various axial integral preserving inflow and outflow velocity pair using cubic Hermite basis functions.



(a) First axial inflow and outflow velocity pair.

(b) Second axial inflow and outflow velocity pair.

(c) Third axial inflow and outflow velocity pair.

Figure 4.11: Various axial integral preserving inflow and outflow velocity pair using quintic Hermite basis functions.

4.3.2. Results and Discussion on Verification of the Continuity Equation

Solving Eq. (4.21) under the prescribed inflow and outflow conditions detailed in Subsection 4.3.1, and incorporating Dirichlet boundary conditions for the transverse velocity, as well as Neumann and Dirichlet boundary conditions for the axial velocity at the inner and outer boundaries, in line with the boundary conditions specified in Table 3.1, consistently yielded zero for nearly all combinations of inflow and outflow velocities, as outlined in Table 4.6.

	Result Implementation Continuity Equation		
	Case 1	Case 2	Case 3
Piecewise linear basis	0.0	0.0	0.0
Cubic Hermite interpolation	-0.00268	$-3.0444e{-16}$	$-3.0444e{-16}$
Quintic Hermite interpolation	-0.00896	$-3.7470e{-16}$	$-3.7470e{-16}$

Table 4.6: Resulting values from the implementation of the continuity equation with known mass conserving inflow and outflow velocities.

However, exceptions were observed for the cases employing higher-order basis functions. Particularly, these exceptions occurred in situations where there were changes in velocity coefficients at the boundary, not governed by the boundary conditions. Specifically, these exceptions occurred in the scenarios depicted in Figs. 4.10a and 4.11a, where changes were applied to the boundary value of the axial velocity at $y = 0$ and to the second-order derivatives of the axial velocity at $y = 0$ and $y = Y$, in case quintic Hermite interpolation basis functions were employed, between consecutive steps. This observation affirms that the continuity equation is accurately discretised for the purpose of the Ainslie wake model when utilising piecewise linear basis functions. However, there seems to be a discrepancy in the implementation when higher-order basis functions are employed, particularly concerning the boundary conditions, since only the cases with completely fixed boundary values do yield zero.

To explain the reason behind this discrepancy a detailed examination was undertaken of the matrices constituting Eq. (4.21), as outlined in Appendix A. Starting with the matrices associated with the term $\frac{\partial U}{\partial x}$, it becomes apparent that when employing piecewise linear basis functions, the vectors resulting from the summation of these matrices along the projection dimension, associated with the known inflow profile and the unknown outflow profile as presented in Eqs. (A.2) and (A.3) respectively, exhibit opposite signs. This is desirable, as it indicates that equal inflow and outflow axial velocity profiles result in $\frac{\partial U(x,y)}{\partial x}$ being equal to zero. Moreover, these vectors demonstrate symmetry, and their values align with the area defined under the projection functions. Specifically, the projection functions in the middle of the domain encompass twice the area compared to those at the boundaries, and correspondingly, the coefficients in the middle are also twice as large as those at the boundaries in the vector. Consequently, as long as the area beneath the inflow and outflow velocity profiles remains consistent, the addition of these vectors yields the expected outcome of zero, ensuring mass conservation.

Subsequently, analysing the matrices associated with the term $\frac{\partial U(x,y)}{\partial x}$ for cases where higher-order basis functions are employed, a significant observation emerges. The vectors resulting from the summation along the projection index, which contain the coefficients corresponding to the values, the first-order derivatives, and the second-order derivatives (in the case of quintic Hermite interpolation basis functions) in an alternating manner, exhibit non-symmetrical patterns at the boundaries. This asymmetry arises from the odd nature of the basis functions defining the value of the first-order derivatives. While this characteristic does not impact the middle of the domain, it affects the boundaries where only the positive or negative counterpart of the odd function is present. More precisely, the multiplication between even and odd functions at these boundaries leads to sign changes in the boundary columns on the rows where even functions are multiplied by the odd function, resulting in slightly different values in the resulting vector. For example, in Eq. (A.10), the first value in the vector is 0.2607 instead of 0.25, and the previous to last value in the vector is 0.2393 instead of 0.25. While the overall sum remains 0.5, this asymmetry causes coefficients in the matrix to no longer align with the areas they represent, resulting in non-zero outcomes for the continuity equation in certain scenarios. However, the opposing signs between the vectors related to the inflow and outflow conditions ensure that when all boundary values are fixed, the continuity equation is satisfied. Nevertheless, fixing each of the boundary conditions is not in line with the Ainslie wake model as there is diffusion present resulting in a change of the boundary value at $r = 0$.

An initially considered solution to satisfy the continuity equation in cases where not all boundary conditions are fixed was to eliminate the basis functions at the boundaries that are not fixed or are fixed to zero if they have an opposite even or odd property to the fixed boundary conditions. This approach was intended to result in symmetric vectors or vectors that add up to zero due to the fixed coefficients at the boundaries, thereby resolving the issue. However, upon further examination, it was found that this method alters the matrices in such a way that it does not effectively work. Consequently, this solution is not viable, indicating that higher-order basis functions do not satisfy the continuity equation for the purpose of the Ainslie wake model. From this, it can be concluded that the highest-order basis functions that remain viable for solving the Ainslie wake model are quadratic basis functions or higher-order basis functions that exhibit symmetry. Higher-order Hermite basis functions that do not maintain symmetry introduce complexities that render them unsuitable for this model.

Lastly, upon closer examination of the matrices associated with the term $\frac{\partial V(x,y)}{\partial y}$, it is observed that the values in the vectors resulting from the summation of these matrices along the projection dimension, associated with the known inflow profile at $y = 0$ and the unknown outflow profile at $y = Y$ respectively, exhibit opposite signs. This is again desirable, as it indicates that equal inflow and outflow transverse velocity profiles result in $\frac{\partial U}{\partial x}$ being equal to zero, ensuring mass conservation. However, symmetry is not present at the boundaries along the axial direction, as the vector values on the right side of the cell are twice those on the left side. This occurs because both basis functions in the axial direction are projected onto the right axial basis function, resulting in twice the integral value when the same basis functions are projected onto each other compared to when opposite basis functions are projected onto each other. The drawback from this observation is that not every transverse inflow and outflow profile with a preserved integral yields a zero solution. Nonetheless, this does not pose any problems for the Ainslie wake model due to the imposed Dirichlet boundary condition, which fixes the values of $V(x, y)$ at the radial boundaries. However, if this were not the case, projecting onto piecewise constant basis functions in the axial direction could be an option to avoid mass conservation issues arising from the observed asymmetry.

Even though Cartesian coordinates were used for simplicity in verifying the continuity equation, it is important to consider the changes that occur when employing cylindrical coordinates. Starting with the term $\frac{\partial U(x,y)}{x}$, the transition to cylindrical coordinates, resulting in $\frac{\partial U(x,r)}{\partial x}$, does not significantly affect this term. The primary change occurs when applying the Galerkin projection. More precisely, the integral in the radial direction includes an additional factor of r in the integrand. Consequently, applying the same coefficients to the basis functions with initially equal areas does not yield conserved areas due to this multiplication by r. Therefore, it is necessary to account for factors that adjust the area under the curve to ensure that the defined inflow and outflow axial velocity profiles have the same area. However, making sure the areas under the inflow and outflow axial velocity profiles are perserved yields the same satisfactory results as for the case using Cartesian coordinates.

However, rewriting the term $\frac{\partial V(x,y)}{\partial y}$ in cylindrical coordinates yields $\frac{\partial V(x,r)}{\partial r} + \frac{1}{r} V(x, r)$, which has quite different characteristics. Specifically, when applying the Galerkin projection, the first term will include an extra factor r in the integrand, while the second term results in the extra term r being canceled out due to the factor $\frac{1}{r}$ in the original term, as detailed in Eq. (4.22). This leads to the observation that the values in the vectors resulting from the summation of the divergence matrices along the projection dimension, associated with the known inflow profile at $r = 0$ and the unknown outflow profile at $r = R_{max}$ respectively, no longer align with the area under the inflow and outflow profiles and therefore no longer cancel each other out, compromising mass conservation in most cases. However, due to the Dirichlet boundary condition $V(x, r) = 0$ applied at the boundaries $r = 0$ and $r = R_{max}$ for the Ainslie wake model, the non-zero values at the boundaries cancel out, mitigating the observed issues.

$$\int_{X_l}^{X_r} \int_{0}^{R_{max}} \left(\frac{\partial V(x,r)}{\partial r} + \frac{1}{r} V(x,r) \right) r \mathrm{d}r \mathrm{d}x = \int_{X_l}^{X_r} \int_{0}^{R_{max}} \frac{\partial V(x,r)}{\partial r} r \mathrm{d}r \mathrm{d}x + \int_{X_l}^{X_r} \int_{0}^{R_{max}} V(x,r) \mathrm{d}r \mathrm{d}x \tag{4.22}$$

The main conclusion drawn from the discussion is that for the velocity component approach, the implementation of higher-order basis functions becomes unfeasible due to the odd basis functions defining the values of the first-order derivatives. However, the use of piecewise linear basis functions in the radial direction has shown to yield satisfactory results, particularly in its effective handling of two mass-conserving velocity profiles. It should be noted, however, that for scenarios applying different boundary conditions to the radial velocity mass conservation might be compromised when employing cylindrical coordinates, meaning that the outer radial bound of the domain should be far enough from the disk such that $V(x, R_{max}) = 0$ can be imposed.

Moreover, it should be noted that since the implementation of the stream function ensures global mass conservation, the higher-order basis functions necessary due to the order reduction inherent in the stream function formulation might still be viable for this approach. Thus, while higher-order basis functions are not applicable in the velocity component approach, they might still be suitable for the stream function approach. Therefore, this method should not be completely discarded.

4.4. Verification Non-Linear Solver Implementation

To verify the correct implementation of the Newton-Raphson method for solving the non-linear Ainslie wake model, the non-linear solver was applied to the linear diffusion equation solved in Section 4.2. The outcome of applying the non-linear solver, which converges in one step, and solving the equation without the non-linear solver yielded the same results, confirming the correct functioning of the non-linear solver.

4.5. Verification Ainslie Wake Model

In light of the limitations posed by the higher-order FEM implementation, as identified in Subsection 4.2.4 and based on their unsuitability to the application of the continuity equation when using a velocity component approach as discussed in Subsection 4.3.2, the next course of action involved taking a step back and implementing the velocity component approach utilising piecewise linear basis functions in the radial direction to solve the transport equation. However, despite the anticipation that this approach would yield satisfactory results, as the individual transport equation components for the implementation of the velocity component approach with piecewise linear basis functions have been verified, stability issues have been observed. Therefore, an additional verification step entailed analytically computing each matrix entry and visually confirming that they corresponded with the matrices calculated in the Ainslie wake model. Following this examination, it was concluded that no apparent errors were observed in the matrix entries.

The fact that each component performs adequately on its own, yet instability arises when combined, suggests that the chosen implementation strategy may overlook crucial considerations for stability. However, there is limited knowledge available about applying the finite element method to the Ainslie wake model. Therefore, examining the stability issues encountered when solving the steady incompressible Navier-Stokes equations and the Stokes equations with finite element methods can provide valuable insights into identifying the sources of instability and potential solutions, as these equations are the closest related to the Ainslie wake model equation with available stability information. However, before discussing stability considerations for the incompressible Navier-Stokes equations and the Stokes equations, it is important to first highlight the differences with the Ainslie wake model. The main difference lies in the fact that the Ainslie wake model neglects pressure gradients. Even though this is a valid assumption when applied to the far wake, this simplification can lead to stability issues not present in the Navier-Stokes or Stokes equations, where pressure plays a crucial role in maintaining stability and ensuring accurate flow representation.

To ensure stability when solving the steady incompressible Navier-Stokes equations or the Stokes equations, it is essential to satisfy the Ladyzhenskaya-Babuka-Brezzi (LBB) condition. However, assessing compliance with this condition can be challenging due to its abstract nature. The essence of the LBB condition lies in recognising that the continuity equation in the incompressible Navier-Stokes equations is fully determined by the pressure [86]. Therefore, since the Ainslie wake model explicitly neglects pressure, enforcing only global mass conservation may be insufficient, leading to stability issues. Consequently, local mass conservation might become necessary to stabilise the model. Moreover, with the ultimate goal of extending the Ainslie wake model to include a forcing term and a pressure Poisson solver, adhering to the LBB condition becomes crucial. Failing to meet this condition can lead to an inconsistent system of equations, causing the matrices to become singular or highly ill-conditioned [86]. This adherence is vital to ensure the stability and accuracy of the extended model. While ensuring local mass conservation might yield a stable Ainslie wake model, the extended model might still face stability issues. Therefore, to prevent this preliminary measures should be taken to avoid limitations in future model developments.

To guarantee conformity with the LBB condition, limitations are imposed on the number of applicable elements used. Typically, a widely accepted guideline is that the order of pressure approximation should be one level lower than that of velocity. For instance, if velocity is approximated using a linear polynomial, pressure is approximated by a constant per element [87, 86]. However, relying solely on this guideline may not suffice. Another critical criterion involves ensuring element admissibility, where the number of velocity unknowns exceeds the number of pressure unknowns. Achieving admissibility often involves placing velocity unknowns at midside points rather than vertices. While this maintains conformity with the incompressible continuity equation, it introduces challenges by disrupting velocity continuity across element boundaries and yielding non-conforming elements. Nevertheless, these elements still fulfill the requirements of the continuity equation. Although this approach aids in identifying non-admissible elements, it does not directly explain why certain elements meet admissibility criteria. The exact conditions for admissibility are disclosed under the LBB condition [86].

Based on this, it can be concluded that the current implementation of the finite element method is not suitable for solving the extended Ainslie wake model, and special attention is needed to satisfy the LBB condition. One potential approach to address these challenges is to utilise mixed finite elements, such as Taylor-Hood elements, which contribute to enforcing the divergence-free constraint and maintaining stability.

A mixed finite element approach is a numerical technique used in finite element analysis that involves solving a problem by simultaneously approximating multiple field variables, typically using different types of finite elements for each variable. This method is often employed to overcome stability issues, such as those arising from the violation of the LBB condition, which can occur in standard finite element methods when approximating solutions to certain partial differential equations [88, 89].

The Taylor-Hood element is a popular choice in mixed finite element methods. In a typical Taylor-Hood scheme, the polynomial degree of the pressure basis functions is one lower than that used for the velocities. Specifically, it uses quadratic basis functions for the velocity and linear basis functions for the pressure. This can be implemented by creating two grids one related to the pressure and one to the velocities. The velocity grid contains all pressure nodes but also involves points at the midpoint of each element edge [90, 88, 89].

Another approach to circumvent the complications arising from the absence of pressure in the incompressibility constraint is to implement a penalty function method. This method involves perturbing the continuity equation with a small term containing the pressure [86]. However, since the current stage of model development does not include pressure in the transport equation, this approach is not deemed applicable [86, 89].

5

Extending Ainslie Wake Model with Forcing

The Ainslie wake model, which neglects pressure gradients by assuming their insignificance outside the near wake region, can be enhanced to yield more accurate outcomes within this region by incorporating the pressure gradient. This enhancement involves employing a Poisson solver to solve the pressure Poisson equation provided in Eq. (5.2). The pressure Poisson equation is derived by taking the divergence of the momentum equation, outlined in Eq. (5.1), and enforcing the incompressible continuity equation. The divergence of the pressure gradient can be separated into two contributions one from the flow field (∇p_u) and one from the applied forcing (∇p_f), as indicated in Eq. (5.2). Since the contribution from the forcing can be precomputed, this chapter will focus on the implementation of the forcing term as an initial step in developing the full pressure Poisson solver, thereby simplifying the pressure Poisson equation to Eq. (5.3), ignoring the contribution from the velocity field.

$$(\vec{u} \cdot \nabla)\vec{u} = -\frac{1}{\rho}\nabla p + \nu \Delta \vec{u} + \frac{1}{\rho}\vec{f} \tag{5.1}$$

$$\frac{1}{\rho}\Delta p = -\underbrace{\nabla \cdot ((\vec{u} \cdot \nabla)\vec{u})}_{\Delta p_u} + \frac{1}{\rho}\underbrace{(\nabla \cdot \vec{f})}_{\Delta p_f} \tag{5.2}$$

$$\Delta p_f = \nabla \cdot \vec{f} \tag{5.3}$$

Before delving into the discussion on adding the forcing term to the right-hand side of the momentum equation as a first step in adding a full pressure Poisson solver, it is crucial to acknowledge that due to the challenges encountered with the FEM-based solver for the Ainslie wake model, as outlined in Chapter 4, the decision has been made to apply the extension to an existing solver that employs the FVM as a numerical method for solving the Ainslie wake model.

Section 5.1 begins by examining various representations of the forcing term. Next, Section 5.2 will verify the chosen forcing formulation. Following this, Section 5.3 discusses different strategies for implementing the selected forcing. Finally, Section 5.4 presents the results and limitations of the chosen implementation strategy.

5.1. Actuator Disk Forcing

In the simplified scenario of assuming a constant thrust coefficient, the wind turbine is modelled as an actuator disc. This representation visualises the wind turbine as an infinitely thin disk that exerts a uniform axial force density, denoted as f_x, onto the flow. Additionally, the axial force density f_x mimics the behaviour of a Dirac delta function positioned at the disc, as described in Eq. (5.4) [91].

$$f_x = F(r)\delta(x) \tag{5.4}$$

For ease of implementation, and given that the pressure Poisson equation is elliptic, it is essential to establish a continuous representation of the initially discontinuous actuator disk forcing. This continuous representation of the actuator disk force distribution can be achieved through several approaches, two of which will be presented in this section.

The first approach is based on the idea that the forcing can be defined either as the curl of a vector potential or equivalently as the gradient of a scalar potential due to their equivalence to a constant-strength double element. Here, the forcing is chosen to be represented by the curl of a vector potential, as detailed in Subsection 5.1.1.

Subsequently, the second approach, presented in Subsection 5.1.2, involves implementing the forcing by numerically solving the pressure Poisson equation outlined in Eq. (5.3), where the force density is represented as a Dirac delta function.

5.1.1. Forcing as the Curl of a Vector Potential

In the initial approach, the divergence of the force density is assumed to be zero within the domain, Ω, and singular on the boundary, $\partial\Omega$, as outlined in Eq. (5.5). In this context, it is important to clarify that the term boundary encompasses both the outer boundary of the domain and the inner boundary located at the actuator disk, where the singularities occur. Thus, the actuator disk is considered part of the boundary rather than the domain itself, and it introduces singularities along it. Based on this, the force can be represented as the curl of a vector potential, ϕ, such that Eq. (5.6) holds. Consequently, with these assumptions established, the Biot-Savart law can be applied to construct a continuous representation of the force density, given it is modelled as a constant-strength dipole/doublet element or, equivalently, as a vortex ring.

$$\nabla \cdot \vec{f} = 0, \text{ in } \Omega, \text{ and singular on } \partial\Omega \tag{5.5}$$

$$\vec{f} = \nabla \times \phi \tag{5.6}$$

The force experienced at a specific point, denoted as $\vec{P}$, due to a vortex ring with a radius, R, is determined by integrating the Biot-Savart law, which describes the induced force by a vortex filament of strength, Γ, across a single rotation along the azimuth angle, θ [92, 93, 94]. The resulting expression for $\vec{f}$ is given in Eq. (5.7), where $\vec{r}$ is the distance between the vortex ring and the point at which $\vec{f}$ is calculated, which is defined as outlined in Eq. (5.8). Furthermore, the vortex filament can be expressed as provided in Eq. (5.9).

$$\vec{f} = \frac{\Gamma}{4\pi} \oint_C \frac{\vec{r} \times \mathrm{d}\vec{l}}{4\pi |\vec{r}|^3} \tag{5.7}$$

$$\vec{r} = \vec{V} - \vec{P} = [R\cos\theta - r, R\sin\theta, -x] \tag{5.8}$$

$$\mathrm{d}\vec{l} = [-R\sin\theta, R\cos\theta, 0]\mathrm{d}\theta \tag{5.9}$$

However, when the evaluation point coincides with the ring itself, the solution becomes singular, particularly when $z = z_0$ and $r = R$. To address this singularity, a regularisation parameter δ can be introduced to represent a vortex with a finite core thickness. This leads to a modified version of Eq. (5.7), described in Eq. (5.10) [93, 94].

$$\vec{f} = \frac{\Gamma}{4\pi} \oint \frac{\vec{r} \times \mathrm{d}\vec{l}}{(|\vec{r}|^2 + \delta^2)^{3/2}} \tag{5.10}$$

In conclusion, following certain manipulations, the equation referenced as Eq. (5.10) can be transformed to yield de-singularised expressions for the radial and axial induced forces, as outlined in Eq. (5.11) [93, 94]. These expressions are given in terms of the complete elliptic integrals of the first ($K(m)$) and second ($E(m)$) kind, which are detailed in Eq. (5.13) [95].

$$
\begin{aligned}
f_r^\delta(r,x) =& \frac{\Gamma}{2\pi a}\frac{x}{r}\left[\frac{(r^2+R^2+x^2+\delta^2)}{A}E(m) - K(m)\right] \\
f_x^\delta(r,x) =& \frac{-\Gamma}{2\pi a}\left[\frac{(r^2-R^2+x^2+\delta^2)}{A}E(m) + K(m)\right]
\end{aligned}
\tag{5.11}
$$

with

$$
A = (r-R)^2 + x^2 + \delta^2, \quad a = \sqrt{(r+R)^2+x^2+\delta^2}, \quad \text{and} \quad m = \frac{4rR}{a^2} \tag{5.12}
$$

$$
\begin{aligned}
K(m) &= \int_0^1 [(1-t^2)(1-mt^2)]^{-\frac{1}{2}}\,\mathrm{d}t \\
E(m) &= \int_0^1 (1-t^2)^{-\frac{1}{2}}(1-mt^2)^{\frac{1}{2}}\,\mathrm{d}t
\end{aligned}
\tag{5.13}
$$

With the mathematical representations for the forcing terms established, their connection to the pressure can be inferred from Eq. (5.3), as demonstrated in Eq. (5.14), where p_f indicates the pressure in the domain resulting from the applied force.

$$
\begin{aligned}
\frac{\partial p_f}{\partial x} &= f_x^\delta(r,x) \\
\frac{\partial p_f}{\partial r} &= f_r^\delta(r,x)
\end{aligned}
\tag{5.14}
$$

Considering that the axial momentum equation solely concerns the pressure gradient in the axial direction and referring to the relationship presented in Eq. (5.14), only the axial component of the forcing is needed and it can be directly integrated into the momentum equation.

Furthermore, the vector potential corresponding to the induced forces is provided in Eq. (5.15). Moreover, the vector potential is closely related to the Stokes stream function as shown in Eq. (5.16), and can also be used as a means to implement the forcing [96].

$$
\phi = \frac{\Gamma\sqrt{rR}}{2\pi r}\left[\left(\frac{2}{k}-k\right)K(k^2) - \frac{2}{k}E(k^2)\right] \qquad \text{with} \qquad k^2 = m \tag{5.15}
$$

$$
\psi = r\phi = \frac{\Gamma\sqrt{rR}}{2\pi}\left[\left(\frac{2}{k}-k\right)K(k^2) - \frac{2}{k}E(k^2)\right] \tag{5.16}
$$

5.1.2. Forcing as Dirac Delta Function

The forcing can be incorporated either directly into the momentum equation or through the pressure Poisson equation. The previous method, outlined in Subsection 5.1.1, directly implements the forcing in the momentum equation. In contrast, this approach uses the latter implementation strategy. More specifically, the expression for the pressure due to the applied force by the actuator disk can be derived by solving the Poisson equation, provided in Eq. (5.3). This is achieved by utilising the fact that the solution to the Poisson equation can be obtained through convolution of its right-hand side with the Green's function intrinsic to the pressure Poisson equation.

It should be noted that this approach was briefly investigated in the beginning but discarded due to the complexity of the derivation. However, due to the limitations encountered with the first approach, this approach was revisited and a forcing term was eventually derived based on insights from the derivation of the first approach, presented in Subsection 5.1.1. Unfortunately, due to time constraints, there was no opportunity to implement the second approach, and only the derivation of the forcing formulation is presented.

Starting from the Poisson equation, outlined in Eq. (5.3) and applying convolution with the Green's function intrinsic to the Poisson equation, presented in Eq. (5.17), yields Eq. (5.18).

$$G(r, x, \theta; r', x', \theta') = \frac{1}{4\pi} \frac{1}{\sqrt{(x - x')^2 + r^2 + r'^2 - 2rr' \cos(\theta - \theta')}} \tag{5.17}$$

$$p(r, x) = \int_0^{2\pi} \int_{-\infty}^{\infty} \int_0^{\infty} (\nabla \cdot \vec{f}(r', x')) G(r, x, \theta; r', x', \theta') r' \mathrm{d}r' \mathrm{d}x' \mathrm{d}\theta' \tag{5.18}$$

Then, the forcing is modelled as a Dirac delta function in the axial direction and a Heaviside function in the radial direction, as presented in Eq. (5.19), to ensure the total force is contained within the bounds of a truncated domain. To check this the axial forcing, outlined in Eq. (5.19) is integrated over a domain with finite bounds, as presented in Eq. (5.20). This derivation indicates two key points. Firstly, it shows that the total force is confined within the domain provided the outer radial boundary is larger than the radius of the actuator disk. Secondly, it reveals that the value of f_0 corresponds to the total thrust force.

$$f_x(r, x) = \frac{f_0}{\pi R^2} \left(1 - H(r - R)\right) \delta(x); \qquad f_r(r, x) = 0; \qquad f_\theta(r, x) = 0 \tag{5.19}$$

$$\begin{aligned}
\int_0^{2\pi} \int_{-X_{max}}^{X_{max}} \int_0^{R_{max}} \frac{f_0}{\pi R^2} \left(1 - H(r - R)\right) \delta(x) r \mathrm{d}r \mathrm{d}x \mathrm{d}\theta &= \frac{f_0}{\pi R^2} 2\pi \int_{-X_{max}}^{X_{max}} \int_0^{R_{max}} \left(1 - H(r - R)\right) \delta(x) r \mathrm{d}r \mathrm{d}x \\
&= \frac{2f_0}{R^2} \int_0^{R_{max}} \left(1 - H(r' - R)\right) r \mathrm{d}r \\
&= \frac{2f_0}{R^2} \int_0^{R} r \mathrm{d}r = \frac{2f_0}{R^2} \frac{1}{2} R^2 = f_0
\end{aligned} \tag{5.20}$$

Filling in the expression for the axial forcing into Eq. (5.18) and employing the definition of the Heaviside function results in Eq. (5.21).

$$\begin{aligned}
p(r, x) &= \frac{f_0}{\pi R^2} \int_0^{2\pi} \int_{-\infty}^{\infty} \int_0^{\infty} \left(1 - H(r' - R)\right) \frac{\partial \delta(x')}{\partial x'} \frac{1}{4\pi} \frac{1}{\sqrt{(x - x')^2 + r^2 + r'^2 - 2rr' \cos(\theta - \theta')}} r' \mathrm{d}r' \mathrm{d}x' \mathrm{d}\theta' \\
&= \frac{f_0}{\pi R^2} \int_0^{2\pi} \int_{-\infty}^{\infty} \int_0^{R} \frac{\partial \delta(x')}{\partial x'} \frac{1}{4\pi} \frac{1}{\sqrt{(x - x')^2 + r^2 + r'^2 - 2rr' \cos(\theta - \theta')}} r' \mathrm{d}r' \mathrm{d}x' \mathrm{d}\theta'
\end{aligned} \tag{5.21}$$

The subsequent step involves taking the integral over θ'. However, before proceeding $(\theta - \theta')$ is redefined. Specifically, $(\theta' - \theta) = \tilde{\theta}$ such that $\cos(\theta - \theta') = \cos(\theta' - \theta) = \cos(\tilde{\theta})$, indicating that $\mathrm{d}\theta' = \mathrm{d}\tilde{\theta}$. Additionally, exploiting the symmetry of the cosine function allows for converting the integration bounds from 0 to 2π, to 0 to π by multiplying the integral by 2. This adjustment is crucial when integrating over $\tilde{\theta}$ for substitution with elliptic integrals. The adjusted expression for the pressure before taking the integral over $\tilde{\theta}$ is outlined in Eq. (5.22).

Then, the derivation of the integral over $\tilde{\theta}$, is presented in Eq. (5.23), where the resulting integral involves a complete elliptical integral of the first kind K, as defined in [95].

$$p(r,x) = \frac{f_0}{2\pi^2 R^2} \int\limits_0^{\pi} \int\limits_{-\infty}^{\infty} \int\limits_0^{R} \frac{\partial \delta(x')}{\partial x'} \frac{1}{\sqrt{(x-x')^2 + r^2 + r'^2 - 2rr'\cos\bigl(\tilde{\theta}\bigr)}} r' \mathrm{d}r' \mathrm{d}x' \mathrm{d}\tilde{\theta} \tag{5.22}$$

$$\begin{aligned}
p(r,x) &= \frac{f_0}{2\pi^2 R^2} \int\limits_{-\infty}^{\infty} \int\limits_0^{R} \frac{\partial \delta(x')}{\partial x'} r' \int\limits_0^{\pi} \frac{1}{\sqrt{a - b\cos\bigl(\tilde{\theta}\bigr)}} \mathrm{d}\tilde{\theta} \mathrm{d}r' \mathrm{d}x' \text{ with } \quad a = (x-x')^2 + r^2 + r'^2, \text{ and } b = 2rr' \\
&= \frac{f_0}{2\pi^2 R^2} \int\limits_{-\infty}^{\infty} \int\limits_0^{R} \frac{\partial \delta(x')}{\partial x'} r' \int\limits_0^{\pi} \frac{1}{\sqrt{a - b\left(1 - 2\sin^2\left(\frac{\tilde{\theta}}{2}\right)\right)}} \mathrm{d}\tilde{\theta} \mathrm{d}r' \mathrm{d}x' \\
&= \frac{f_0}{2\pi^2 R^2} \int\limits_{-\infty}^{\infty} \int\limits_0^{R} \frac{\partial \delta(x')}{\partial x'} r' \int\limits_0^{\pi} \frac{1}{\sqrt{(a-b) + 2b\sin^2\left(\frac{\tilde{\theta}}{2}\right)}} \mathrm{d}\tilde{\theta} \mathrm{d}r' \mathrm{d}x' \\
&= \frac{f_0}{2\pi^2 R^2} \int\limits_{-\infty}^{\infty} \int\limits_0^{R} \frac{\partial \delta(x')}{\partial x'} \frac{r'}{\sqrt{a-b}} \int\limits_0^{\pi} \frac{1}{\sqrt{1 - \frac{2b}{b-a}\sin^2\left(\frac{\tilde{\theta}}{2}\right)}} \mathrm{d}\tilde{\theta} \mathrm{d}r' \mathrm{d}x' \\
&= \frac{f_0}{\pi^2 R^2} \int\limits_{-\infty}^{\infty} \int\limits_0^{R} \frac{\partial \delta(x')}{\partial x'} \frac{r'}{\sqrt{a-b}} \int\limits_0^{\pi/2} \frac{1}{\sqrt{1 - \frac{2b}{b-a}\sin^2(u)}} \mathrm{d}u \mathrm{d}r' \mathrm{d}x' \text{ with } u = \frac{\tilde{\theta}}{2}, \text{ and } \mathrm{d}\tilde{\theta} = 2\mathrm{d}u \\
&= \frac{f_0}{\pi^2 R^2} \int\limits_{-\infty}^{\infty} \int\limits_0^{R} \frac{\partial \delta(x')}{\partial x'} \frac{r'}{\sqrt{a-b}} K\left(\frac{2b}{b-a}\right) \mathrm{d}r' \mathrm{d}x' \\
&= \frac{f_0}{\pi^2 R^2} \int\limits_{-\infty}^{\infty} \int\limits_0^{R} \frac{\partial \delta(x')}{\partial x'} \frac{r'}{\sqrt{(x-x')^2 + r^2 + r'^2 - 2rr'}} K\left(\frac{-4rr'}{(x-x')^2 + r^2 + r'^2 - 2rr'}\right) \mathrm{d}r' \mathrm{d}x' \\
&= \frac{f_0}{\pi^2 R^2} \int\limits_{-\infty}^{\infty} \int\limits_0^{R} \frac{\partial \delta(x')}{\partial x'} \frac{r'}{\sqrt{(x-x')^2 + (r-r')^2}} K\left(\frac{-4rr'}{(x-x')^2 + (r-r')^2}\right) \mathrm{d}r' \mathrm{d}x'
\end{aligned} \tag{5.23}$$

The next step involves applying integration by parts to shift the gradient of the Dirac delta function onto the rest of the integrand, in order to solve for the integral over x'. This process utilises the definition of the derivative of the complete elliptic integral of the first kind, K, as given in Eq. (5.24), where E represents the complete elliptic integral of the second kind. Additionally, the definition of the integral of a function multiplied by a Dirac delta function is also employed. The resulting expression for the pressure is given in Eq. (5.25).

$$\frac{\partial K(m)}{\partial m} = \frac{(m-1)K(m) + E(m)}{2m - 2m^2} \tag{5.24}$$

$$\begin{aligned}
p(r,x) = {} & \underbrace{\frac{f_0}{\pi^2 R^2}\int\limits_{-\infty}^{\infty}\int\limits_{0}^{R}\frac{\partial}{\partial x'}\left[\delta(x')\frac{r'}{\sqrt{(x-x')^2+(r-r')^2}}K\left(\frac{-4rr'}{(x-x')^2+(r-r')^2}\right)\right]\mathrm{d}r'\mathrm{d}x'}_{=0} \\
& -\frac{f_0}{\pi^2 R^2}\int\limits_{-\infty}^{\infty}\int\limits_{0}^{R}\delta(x')\frac{\partial}{\partial x'}\left[\frac{r'}{\sqrt{(x-x')^2+(r-r')^2}}K\left(\frac{-4rr'}{(x-x')^2+(r-r')^2}\right)\right]\mathrm{d}r'\mathrm{d}x' \\
= {} & -\frac{f_0}{\pi^2 R^2}\int\limits_{-\infty}^{\infty}\int\limits_{0}^{R}\delta(x')\frac{r'(x-x')}{\sqrt{(x-x')^2+(r-r')^2}^3}K\left(\frac{-4rr'}{(x-x')^2+(r-r')^2}\right)\mathrm{d}r'\mathrm{d}x' \\
& -\frac{f_0}{\pi^2 R^2}\int\limits_{-\infty}^{\infty}\int\limits_{0}^{R}\delta(x')\frac{r'}{\sqrt{(x-x')^2+(r-r')^2}}\frac{\left(\frac{-4rr'}{(x-x')^2+(r-r')^2}-1\right)K\left(\frac{-4rr'}{(x-x')^2+(r-r')^2}\right)+E\left(\frac{-4rr'}{(x-x')^2+(r-r')^2}\right)}{2\left(\frac{-4rr'}{(x-x')^2+(r-r')^2}\right)-2\left(\frac{-4rr'}{(x-x')^2+(r-r')^2}\right)^2}\mathrm{d}r'\mathrm{d}x' \\
= {} & -\frac{f_0}{\pi^2 R^2}\int\limits_{0}^{R}\frac{r'}{\sqrt{x^2+(r-r')^2}^3}\left(xK\left(\frac{-4rr'}{x^2+(r-r')^2}\right)+\frac{\left(\frac{-4rr'}{x^2+(r-r')^2}-1\right)K\left(\frac{-4rr'}{x^2+(r-r')^2}\right)+E\left(\frac{-4rr'}{x^2+(r-r')^2}\right)}{2\left(\frac{-4rr'}{x^2+(r-r')^2}\right)-2\left(\frac{-4rr'}{x^2+(r-r')^2}\right)^2}\right)\mathrm{d}r'
\end{aligned} \tag{5.25}$$

To simplify the expression in Eq. (5.25), the variable m is introduced, which is defined in Eq. (5.26). Additionally, also the desingularised formulation is presented in Eq. (5.27).

$$m = \frac{-4rr'}{x^2+(r-r')^2} \tag{5.26}$$

$$m_\delta = \frac{-4rr'}{x^2+(r-r')^2+\delta^2} \tag{5.27}$$

Applying the definition for m to Eq. (5.25) yields Eq. (5.28).

$$p(r,x) = -\frac{f_0}{\pi^2 R^2}\int\limits_{0}^{R}\frac{r'}{\sqrt{x^2+(r-r')^2}^3}\left(xK(m)+\frac{(m-1)K(m)+E(m)}{2m-2m^2}\right)\mathrm{d}r' \tag{5.28}$$

The expression in Eq. (5.28) still contains an integral over r'. However, no analytical solution exists for this integral, as confirmed using both Wolfram Alpha and Mathematica. Consequently, this integral must be solved using numerical integration techniques.

Lastly, to obtain the total forcing over the entire domain due to the pressure, the gradient of the pressure in the axial direction must be integrated over the full domain, as shown in Eq. (5.29).

$$
\begin{aligned}
F_{total}(r,x) =& \int_0^{R_{max}} \int_{X_{min}}^{X_{max}} \int_0^{2\pi} \frac{\partial p(r,x)}{\partial x} r \mathrm{d}\theta \mathrm{d}x \mathrm{d}r \\
=& -\frac{f_0}{\pi^2 R^2} \int_0^{R_{max}} \int_{X_{min}}^{X_{max}} \int_0^{2\pi} \frac{\partial}{\partial x} \left[\int_0^{R} \frac{r'}{\sqrt{x^2+(r-r')^2}^3} \left(xK(m) + \frac{(m-1)K(m)+E(m)}{2m-2m^2} \right) \mathrm{d}r' \right] r \mathrm{d}\theta \mathrm{d}x \mathrm{d}r \\
=& \frac{f_0}{4\pi^2 R^2} \int_0^{R_{max}} \int_{X_{min}}^{X_{max}} \int_0^{2\pi} \frac{\partial}{\partial x} \left[\int_0^{R} \frac{-4rr'}{\sqrt{x^2+(r-r')^2}^3} \left(xK(m) + \frac{(m-1)K(m)+E(m)}{2m-2m^2} \right) \mathrm{d}r' \right] \mathrm{d}\theta \mathrm{d}x \mathrm{d}r \\
=& \frac{f_0}{4\pi^2 R^2} \int_0^{R_{max}} \int_{X_{min}}^{X_{max}} \int_0^{2\pi} \frac{\partial}{\partial x} \left[\int_0^{R} \frac{1}{\sqrt{x^2+(r-r')^2}} \left(\left(mx - \frac{1}{2} \right) K(m) + \frac{E(m)}{2(1-m)} \right) \mathrm{d}r' \right] \mathrm{d}\theta \mathrm{d}x \mathrm{d}r \\
=& \frac{f_0}{2\pi R^2} \int_0^{R_{max}} \int_{X_{min}}^{X_{max}} \frac{\partial}{\partial x} \left[\int_0^{R} \frac{1}{\sqrt{x^2+(r-r')^2}} \left(\left(mx - \frac{1}{2} \right) K(m) + \frac{E(m)}{2(1-m)} \right) \mathrm{d}r' \right] \mathrm{d}x \mathrm{d}r \\
=& \frac{f_0}{2\pi R^2} \int_0^{R_{max}} \int_0^{R} \left[\frac{1}{\sqrt{x^2+(r-r')^2}} \left(\left(mx - \frac{1}{2} \right) K(m) + \frac{E(m)}{2(1-m)} \right) \right] \Bigg|_{X_{min}}^{X_{max}} \mathrm{d}r' \mathrm{d}r
\end{aligned}
\tag{5.29}
$$

The expression derived in Eq. (5.29) introduces an additional integral over r that cannot be solved analytically, necessitating numerical integration. Consequently, to determine the total force within the domain, numerical integration over both r, and r' is required.

To ensure accurate results, it is essential to employ a suitable numerical integration method for these integrals. Analysing these numerical results will provide valuable insights into the behavior of the total axial force, allowing for a comparison with the previous forcing formulation to assess its effectiveness. However, due to current time constraints, further investigation of this aspect will be postponed to future studies.

For completion, the desingularised expression for the total axial force is presented in Eq. (5.30)

$$
F_{total}(r,x) = \frac{f_0}{2\pi R^2} \int_0^{R_{max}} \int_0^{R} \left[\frac{1}{\sqrt{x^2+(r-r')^2+\delta^2}} \left(\left(m_\delta x - \frac{1}{2} \right) K(m_\delta) + \frac{E(m_\delta)}{2(1-m_\delta)} \right) \right] \Bigg|_{X_{min}}^{X_{max}} \mathrm{d}r' \mathrm{d}r
\tag{5.30}
$$

5.2. Verification Induced Forcing Formulation

To verify the correct implementation of the induced forcing, the induced forcing obtained using Eq. (5.11) is compared with the induced forcing obtained from the equivalent quadrilateral doublet element, using the functions from an existing three-dimensional boundary element solver. Specifically, a circle with the same radius as the vortex ring was formed consisting of quadrilateral doublet elements, as illustrated in Fig. 5.1. 5,000 equidistant points were used along the circle's circumference to achieve an approximation that closely resembles a perfect circle. Then, applying the same dipole strength the induced axial and radial forcing were compared at several points. It should be noted, however, that since the quadrilateral doublet element induced forcing calculation does not implement a de-regularisation constant, the comparison is only valid for very small δ. Therefore, δ is set equal to 1e−10, as further increasing δ has shown not to have any more effect on the resulting calculated difference in induced forcing.

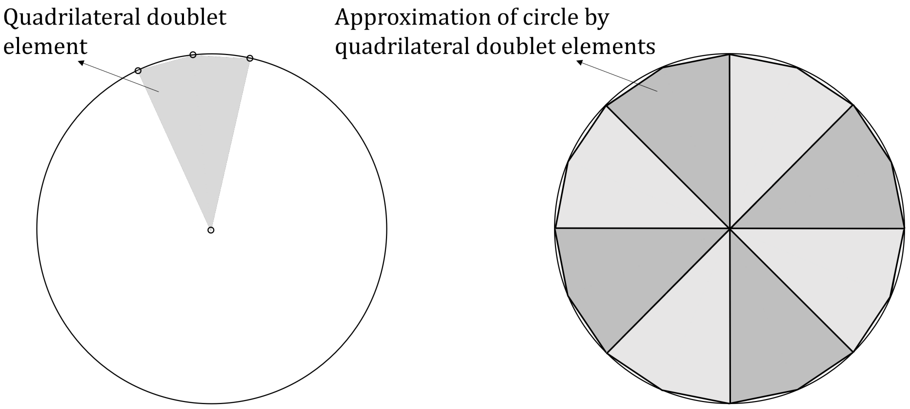


Figure 5.1: Illustration of quadrilateral doublet element (left) and of the approximation of a circle with constant doublet strength using constant-strength quadrilateral doublet elements (right).

For this verification case, the errors in the induced total force, axial force, and radial force, when evaluated at points relatively close to the disk, are of the order of $\mathcal{O}(1\text{e}{-8})$ and $\mathcal{O}(1\text{e}{-9})$. This error decreases further for points located farther away from the disk and also when increasing the number of points on the circle's circumference, thereby more accurately approximating a real circle with quadrilateral doublet elements. Specifically, as the number of quadrilateral doublet elements in increased the spacing between the circle circumference and the edges of the quadrilateral doublet elements as can be seen from Fig. 5.1 decreases matching the circle more closely.

5.3. Forcing Term Implementation into the Momentum Equation

In this section, the implementation strategy to incorporate the forcing term into the momentum equation will be laid out. For this purpose, two different approaches will be presented. The first approach directly integrates the forcing term over the volume of each marching step, ensuring an accurate representation of the forces within the computational domain, as discussed in Subsection 5.3.1. The second approach utilises the vector potential, which offers an alternative way to introduce the forcing term into the momentum equation. This approach is further elaborated in Subsection 5.3.2. By using the vector potential, the computational complexity of the forcing term can be reduced, potentially reducing the computational cost.

5.3.1. Implementation of Forcing Term using the Induced Forcing

To include the gradient of the pressure component due to the forcing, ∇p_f, in the momentum equation for the finite volume framework, the forcing needs to be integrated over each cell volume in the domain. Specifically, for each marching step, which corresponds to one axial cell, the integral of the induced forcing is calculated for each radial mesh cell. These integrals are then combined into a vector and added to the right-hand side of the momentum equation. The vector added to the right-hand side of the momentum equation is represented in Eq. (5.31), where X_l and X_r refer to the left and right bounds of the specific marching step respectively, and the indices of the radial integration bounds indicate the radial mesh nodes.

$$\begin{bmatrix} \int\limits_{0}^{2\pi}\int\limits_{X_l}^{X_r}\int\limits_{r_0}^{r_1} f_x(r,x)r\,\mathrm{d}r\,\mathrm{d}x\,\mathrm{d}\theta \\ \int\limits_{0}^{2\pi}\int\limits_{X_l}^{X_r}\int\limits_{r_1}^{r_2} f_x(r,x)r\,\mathrm{d}r\,\mathrm{d}x\,\mathrm{d}\theta \\ \vdots \\ \int\limits_{0}^{2\pi}\int\limits_{X_l}^{X_r}\int\limits_{r_{N_r-2}}^{r_{N_r-1}} f_x(r,x)r\,\mathrm{d}r\,\mathrm{d}x\,\mathrm{d}\theta \\ \int\limits_{0}^{2\pi}\int\limits_{X_l}^{X_r}\int\limits_{r_{N_r-1}}^{r_{N_r}} f_x(r,x)r\,\mathrm{d}r\,\mathrm{d}x\,\mathrm{d}\theta \end{bmatrix} = \begin{bmatrix} 2\pi\int\limits_{X_l}^{X_r}\int\limits_{r_0}^{r_1} f_x(r,x)r\,\mathrm{d}r\,\mathrm{d}x \\ 2\pi\int\limits_{X_l}^{X_r}\int\limits_{r_1}^{r_2} f_x(r,x)r\,\mathrm{d}r\,\mathrm{d}x \\ \vdots \\ 2\pi\int\limits_{X_l}^{X_r}\int\limits_{r_{N_r-2}}^{r_{N_r-1}} f_x(r,x)r\,\mathrm{d}r\,\mathrm{d}x \\ 2\pi\int\limits_{X_l}^{X_r}\int\limits_{r_{N_r-1}}^{r_{N_r}} f_x(r,x)r\,\mathrm{d}r\,\mathrm{d}x \end{bmatrix} \tag{5.31}$$

Even though the forcing is symbolically represented in Matlab, the presence of complete elliptic integrals within the integrand, makes symbolic evaluation of the integrals unfeasible. Hence, the integrals are resolved using the Gauss-Legendre quadrature rule, as outlined in Appendix B.

To verify the correct implementation of the Gauss-Legendre quadrature rule, numerical integration is applied to a function with known integrals. The result of the numerical integration is then compared with the analytically calculated integral to ensure accuracy.

Applying the Gauss-Legendre quadrature rule to Eq. (5.31) results in Eq. (5.32). In this solution, the Gauss-Legendre quadrature weights and nodes in the axial direction remain consistent across each vector entry and only vary per marching step, while they differ in the radial direction, as evident from Eq. (5.32).

$$\begin{bmatrix} 2\pi \left(\sum_{i=1}^{N} w_{x,i} \sum_{j=1}^{N} w_{r,j} f_x(r_j, x_i) r_j \right) \\ 2\pi \left(\sum_{i=1}^{N} w_{x,i} \sum_{j=1}^{N} w_{r,j} f_x(r_j, x_i) r_j \right) \\ \vdots \\ 2\pi \left(\sum_{i=1}^{N} w_{x,i} \sum_{j=1}^{N} w_{r,j} f_x(r_j, x_i) r_j \right) \\ 2\pi \left(\sum_{i=1}^{N} w_{x,i} \sum_{j=1}^{N} w_{r,j} f_x(r_j, x_i) r_j \right) \end{bmatrix} \tag{5.32}$$

Furthermore, the Gauss-Legendre theorem has relatively poor convergence when it comes to the approximation of the integral of singular functions. In principle, a specific quadrature rule for the approximation of the integral of singular functions should yield better results. However, since a higher regularisation parameter smoothens out the singularity, one can play around with this parameter to improve the convergence. Specifically, it has been observed that when the regularisation parameter is of the same order as the largest cell boundary close to the actuator disk over which the integral is calculated, the error between using 10 quadrature points and 20 quadrature points is of the order of $\mathcal{O}(1\text{e}{-14})$. This indicates that the accuracy of the numerical integration is relatively insensitive to the number of quadrature points within this range, suggesting that 10 quadrature points may be sufficient for practical purposes in this context. Therefore, δ is set equal to the larger of Δx and Δr near the actuator disk, as reducing the number of quadrature points further reduces the computational cost. Furthermore, it should be noted that the number of quadrature points can be adjusted based on cell size and distance from the disk. Smaller cells require fewer quadrature points for accurate results, and cells located farther from the disk also need fewer points. By employing an exponential mesh that increases with distance from the disk, using as few as 10 quadrature points for the large cells further away from the disk is sufficient. The optimisation of the number of quadrature points to achieve a balance between accuracy and computational cost, along with exploration of more suitable numerical integration methods, however, fall outside the scope of this thesis and are left for future work.

5.3.2. Implementation of Forcing Term using the Vector Potential

Under the axisymmetric assumption of the Ainslie wake model, the flow can be considered two-dimensional in the sense that the azimuthal velocity component around the axis is identically zero, and that the velocity gradient in the azimuthal direction is also zero [97].

Assuming axisymmetric flow, the Stokes stream function, ψ, as outlined in Eq. (5.16) can be employed. Furthermore, the Stokes stream function can be related to the induced forcing vector field as presented in Eq. (5.33), where $\vec{i}_\theta$ is the unit vector in the azimuthal direction [97].

$$\vec{f} = -\frac{1}{r}(\vec{i}_\theta \times \nabla\psi) \quad \text{with} \quad f_x = \frac{1}{r}\frac{\partial\psi}{\partial r}, \quad \text{and} \quad f_r = -\frac{1}{r}\frac{\partial\psi}{\partial x} \tag{5.33}$$

Then, the formulation for the induced force vector field can be integrated over the cell volume, which provides the forcing term added to the right-hand side of the momentum equation when employing the finite volume method. The corresponding derivation is presented in Eq. (5.34), where the relation between the Stokes stream function and the vector potential, presented in Eq. (5.16), is employed. Furthermore, the first row of the vector contains the integrated induced forcing in the radial direction, while the last row contains the integrated induced forcing in the axial direction. As discussed in Subsection 5.1.1 only the axial component of the forcing is required, making the first element therefore redundant.

$$\begin{aligned}
&\int_0^{2\pi}\int_{X_l}^{X_r}\int_{R_d}^{R_u} -\frac{1}{r}(\vec{i}_\theta \times \nabla\psi(r,x))r\mathrm{d}r\mathrm{d}x = -2\pi\vec{i}_\theta \times \int_{X_l}^{X_r}\int_{R_d}^{R_u} \nabla\psi(r,x)\mathrm{d}r\mathrm{d}x = -2\pi\vec{i}_\theta \times \int_{X_l}^{X_r}\int_{R_d}^{R_u} \begin{pmatrix} \frac{\partial\psi(r,x)}{\partial r} \\ \frac{1}{r}\frac{\partial\psi(r,x)}{\partial\theta} = 0 \\ \frac{\partial\psi(r,x)}{\partial x} \end{pmatrix} \mathrm{d}r\mathrm{d}x \\
&= -2\pi \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \times \begin{pmatrix} \int_{X_l}^{X_r} \left(\psi(R_u,x) - \psi(R_d,x)\right)\mathrm{d}x \\ 0 \\ \int_{R_d}^{R_u} \psi(r,X_r) - \psi(r,X_l)\mathrm{d}r \end{pmatrix} = 2\pi \begin{pmatrix} -\int_{R_d}^{R_u} \psi(r,X_r) - \psi(r,X_l)\mathrm{d}r \\ 0 \\ \int_{X_l}^{X_r} \left(\psi(R_u,x) - \psi(R_d,x)\right)\mathrm{d}x \end{pmatrix} \\
&= 2\pi \begin{pmatrix} -\int_{R_d}^{R_u} r\phi(r,X_r) - r\phi(r,X_l)\mathrm{d}r \\ 0 \\ \int_{X_l}^{X_r} \left(R_u\phi(R_u,x) - R_d\phi(R_d,x)\right)\mathrm{d}x \end{pmatrix}
\end{aligned} \tag{5.34}$$

Moreover, also in this approach the Gauss-Legendre quadrature rule is employed to determine the integral, leading to the expression for the forcing term vector in Eq. (5.35).

$$\begin{bmatrix} 2\pi\int_{X_l}^{X_r} \left(r_{(1)}\phi(r_{(1)},x) - r_{(0)}\phi(r_{(0)},x)\right)\mathrm{d}x \\ 2\pi\int_{X_l}^{X_r} \left(r_{(2)}\phi(r_{(2)},x) - r_{(1)}\phi(r_{(1)},x)\right)\mathrm{d}x \\ \vdots \\ 2\pi\int_{X_l}^{X_r} \left(r_{(N_r-1)}\phi(r_{(N_r-1)},x) - r_{(N_r-2)}\phi(r_{(N_r-2)},x)\right)\mathrm{d}x \\ 2\pi\int_{X_l}^{X_r} \left(r_{(N_r-1)}\phi(r_{(N_r-1)},x) - r_{(N_r)}\phi(r_{(N_r)},x)\right)\mathrm{d}x \end{bmatrix} = \begin{bmatrix} 2\pi\left(r_{(1)}\sum_{i=1}^{N} w_i\phi(r_{(1)},x_i) - r_{(0)}\sum_{i=1}^{N} w_i\phi(r_{(0)},x_i)\right) \\ 2\pi\left(r_{(2)}\sum_{i=1}^{N} w_i\phi(r_{(2)},x_i) - r_{(1)}\sum_{i=1}^{N} w_i\phi(r_{(1)},x_i)\right) \\ \vdots \\ 2\pi\left(r_{(N_r-1)}\sum_{i=1}^{N} w_i\phi(r_{(N_r-1)},x_i) - r_{(N_r-2)}\sum_{i=1}^{N} w_i\phi(r_{(N_r-2)},x_i)\right) \\ 2\pi\left(r_{(N_r-1)}\sum_{i=1}^{N} w_i\phi(r_{(N_r-1)},x_i) - r_{(N_r)}\sum_{i=1}^{N} w_i\phi(r_{(N_r)},x_i)\right) \end{bmatrix} \tag{5.35}$$

To verify the accurate implementation of this approach, a comparison was conducted between the resulting integrated induced vector field using this method and the one obtained using the previously verified approach outlined in Subsection 5.3.1, employing a fixed set of parameters. Upon comparison, it was observed that the error between the two was of the order of $\mathcal{O}(1\mathrm{e}{-17})$. This indicates that the approach discussed in this section has been accurately implemented.

5.3.3. Comparison Forcing Implementation Approaches

To identify the most advantageous implementation strategy, the computational time required to obtain the integrated forcing is compared across the two different methods. Using a uniform radial mesh from 0 m to 5 m with 41 nodes and a uniform axial mesh running from -5 m to 5 m with 21 nodes, resulting in 800 mesh cells, the elapsed time using the implementation strategy presented in Subsection 5.3.1 is 389.8 seconds, whereas the elapsed time using the implementation strategy in Subsection 5.3.2 is 8.2 seconds.

This significant difference in computational time arises because the Gauss-Legendre quadrature rule is evaluated only 20 times in the latter method, compared to 800 times in the former method. Essentially, this comparison highlights the inefficiency of two-dimensional quadrature over one-dimensional quadrature, as the former requires significantly more points to achieve the same accuracy. Additionally, the ratio between the number of evaluations and the computational time is of the same order, reinforcing this point.

Based on the observations it can be concluded that the approach employing the vector potential performs significantly better in terms of computational cost, making it the preferred implementation approach for calculating the forcing term. Moreover, to further reduce the computational cost of the force term calculation, one can make use of the symmetry across the actuator disk observed in the axially induced forcing, such that only half of the forcing field needs to be calculated.

5.3.4. Determination of Γ for a given Thrust Coefficient

The strength of the vortex ring, Γ, is a constant that linearly scales the axial force exerted by the vortex ring at a specific location in the domain, as can be deducted from Eq. (5.11).

The total axial force exerted within the computational domain should equal the total thrust force exerted by the actuator disk. To achieve this, one can first calculate the total axial force within the domain for a vortex strength of one. Then, the actual value of Γ is determined by ensuring that the total force in the domain exerted by the vortex ring matches the total thrust force, as outlined in Eqs. (5.36) and (5.37). This shows that Γ is directly related to the thrust force and, consequently, the thrust coefficient generated by the actuator disk. It should be noted that this calculation is feasible because the total axial force exerted by the actuator disk is assumed to converge quickly as the domain size increases to infinity.

$$T = \Gamma \underbrace{\int_0^{2\pi}\int_{X_{min}}^{X_{max}}\int_0^{R_{max}} f_x(x,r)\bigg|_{\Gamma=1} r\mathrm{d}r\mathrm{d}x\mathrm{d}\theta}_{F_{x,total}|_{\Gamma=1}} \quad \text{or} \quad C_T = \frac{2\Gamma}{\rho V_\infty^2 \pi R^2}\underbrace{\int_0^{2\pi}\int_{X_{min}}^{X_{max}}\int_0^{R_{max}} f_x(x,r)\bigg|_{\Gamma=1} r\mathrm{d}r\mathrm{d}x\mathrm{d}\theta}_{F_{x,total}|_{\Gamma=1}} \tag{5.36}$$

$$\Gamma = \frac{T}{F_{x,total}|_{\Gamma=1}} = \frac{\frac{1}{2}\rho V_\infty^2 \pi R^2 C_T}{F_{x,total}|_{\Gamma=1}} \tag{5.37}$$

An alternative method to relate the thrust force T to the vortex strength Γ utilises the Kutta-Joukowski theorem, as defined in Eq. (5.38). Here $\vec{U}$, represents a vector comprising the radial, azimuthal, and axial velocity components at the actuator disk respectively, while $\vec{\Gamma}$ denotes a vector containing the vortex strength. Furthermore, a denotes the axial induction factor, and Ω wind turbine's rotational speed.

$$L = \rho(\vec{U}\times\vec{\Gamma}) \quad \text{with} \quad \vec{U} = \begin{bmatrix} 0 \\ \Omega r \\ U_\infty(1-a) \end{bmatrix}, \text{ and } \vec{\Gamma} = \begin{bmatrix} \Gamma \\ 0 \\ 0 \end{bmatrix} \tag{5.38}$$

Applying this theorem, the force on an elemental length dr along the wind turbine blade can be related to the bound circulation, as expressed in Eq. (5.39) [93]. From this derivation the thrust force on an elemental length dr along the wind turbine blade is given by $dT = -\rho\Gamma r$.

$$F = \rho \begin{bmatrix} 0 \\ \Omega r \\ U_\infty(1-a) \end{bmatrix} \times \begin{bmatrix} \Gamma \\ 0 \\ 0 \end{bmatrix} = \rho \begin{bmatrix} 0 \\ U_\infty(1-a)\Gamma \\ -\Gamma\Omega r \end{bmatrix} \tag{5.39}$$

Assuming Γ is constant along a blade, the expression for the local thrust force dT can be integrated across the rotor radius, and summation over all B wind turbine blades provides the total thrust force as depicted in Eq. (5.40) [93].

$$T = \frac{1}{2}\rho U_\infty^2 \pi R^2 C_T = \frac{1}{2}\rho\Omega R^2 \Gamma B \tag{5.40}$$

Subsequently, from Eq. (5.40) an expression for the vortex strength can be derived which is outlined in Eq. (5.41), where λ denotes the tip speed ratio [93].

$$\Gamma = \frac{\pi U_\infty^2 C_T}{\Omega B} = \frac{2T}{\rho \Gamma R^2 B} = \frac{2T}{\rho \lambda U_\infty R B} \tag{5.41}$$

For this thesis, the former approach is preferred due to its requirement for less specific operational wind turbine information. Additionally, it links Γ more closely to the exact forcing applied by the vortex ring.

Furthermore, in reality, the actuator disk does not exert a constant thrust force on the flow, as the thrust coefficient varies along the blade length. To achieve a more realistic forcing, blade element theory can be employed. Specifically, the actuator disk is divided into several independent annuli, each with its own thrust coefficient. Therefore, instead of having a single vortex ring, a superposition of vortex rings is used to achieve a varying loading in the radial direction. More specifically, the outermost vortex ring will have a strength corresponding to the thrust coefficient of the outermost radial annulus. Each subsequent vortex ring will have a strength related to the change in the thrust coefficient, similar to the concept behind lifting line theory [98].

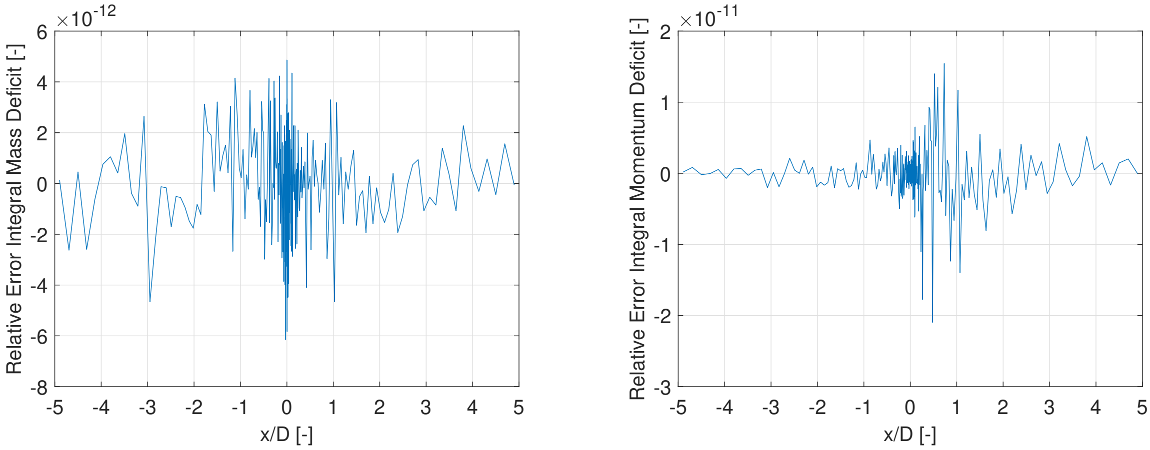
5.4. Results Forcing Implementation

This section will present the results obtained from extending the Ainslie wake model with the inclusion of the forcing term. The forcing term implementation must not compromise mass and momentum conservation. Therefore, Subsection 5.4.1 will present the observed errors in the mass and momentum conservation when the forcing term is implemented. Furthermore, Subsection 5.4.2 will present the axial velocity profile at the actuator disk and compare it against the profile obtained in [91] to verify the accuracy of the implementation. Lastly, Subsection 5.4.4 will discuss the observed limitations due to the employed formulation of the forcing term itself.

5.4.1. Mass and Momentum Conservation

The incorporation of the forcing term into the Ainslie wake model should preserve mass and momentum conservation. Therefore, the integral mass and momentum deficits will be outlined in this section. However, before delving into the results, the simulation parameters are presented.

The inflow velocity for the simulation is set to a constant 10 m/s, while the turbine radius, taken from the DTU 10MW wind turbine, measures 89.15 m. The computational domain extends from -5D to 5D in the axial direction and from 0D to 2D in the radial direction, with D representing the turbine diameter. The regularisation parameter, δ, is set to 1, based on the largest step size near the actuator disk, which is 0.7146 m in the radial direction. This slightly larger value for the regularisation constant ensures a more smeared-out effect of the forcing near the actuator disk, contributing to the stability of the model. The thrust coefficient for this scenario is set to 0.838, corresponding to the total thrust coefficient for the DTU 10MW turbine with a uniform inflow velocity of 10 m/s, as outlined in [82]. The strength of the vortex ring, Γ, was then calculated using the procedure outlined in Subsection 5.3.4.



(a) Integral mass deficit.

(b) Integral momentum deficit.

Figure 5.2: Relative errors of integral mass and momentum deficits.

The relative errors of the integral mass and momentum deficits are provided Fig. 5.2. From these results, it can be observed that the corresponding integral mass and momentum errors are of the order of $\mathcal{O}(10^{-12})$ and $\mathcal{O}(10^{-11})$ respectively. These errors are considered acceptable and confirm that the implementation of the forcing term does not compromise mass and momentum conservation.

5.4.2. Axial Velocity Profile at Actuator Disk

In this section, the normalised axial velocity profile at the actuator disk will be presented and compared with the results from [91], using mostly the same simulation parameters as outlined in Subsection 5.4.1. The primary difference is a slightly higher thrust coefficient, adjusted to match the parameters used in [91]. Specifically, the thrust coefficient was set to 8/9. Additionally, since the paper generates the profile for an inviscid flow, the constant viscosity was set to 0.0 m^2/s. Lastly, the regularisation constant, δ, was decreased to 0.1. This adjustment was made because, based on visual comparison, the profile in the literature showed a sharper peak than what was initially observed with $\delta = 1$. It should be noted that since the paper does not specify the exact value of the regularisation constant used, there may still be some discrepancy. Moreover, the computational domain was extended to achieve a more converged value for Γ. Specifically, the domain extends from $-40D$ to $40D$ in the axial direction and from $0D$ to $30D$ in the radial direction.

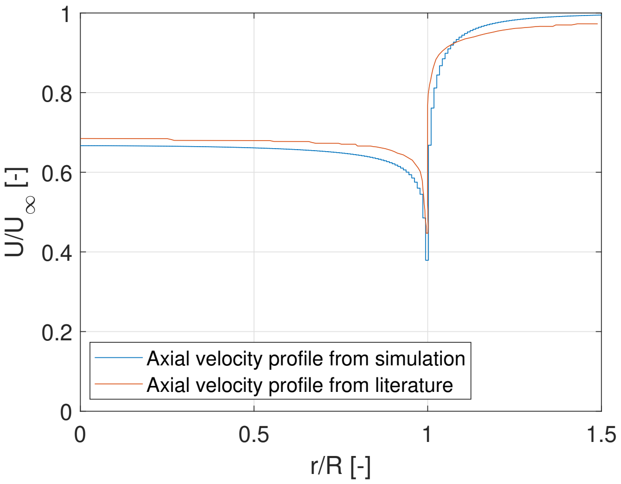


Figure 5.3: Comparison of normalised axial velocity profile at x = 0 m for a thrust coefficient C_T equal to 8/9 with literature [91]

Comparing the normalised axial velocity profile at the actuator disk obtained from the Ainslie wake model with the profile described in the literature, shown in Fig. 5.3, reveals a close agreement. Specifically, from the axis of symmetry to the edge of the actuator disk, the normalised axial velocity values exhibit only a minor offset upon visual inspection. However, Ainslie wake model simulation shows a lower peak value at the edge of the actuator disk. Additionally, there is a notable difference in the rate of velocity increase just beyond the actuator disk. In the Ainslie wake model, the velocity reaches the inflow velocity by $r/R = 1.5$, whereas the literature profile shows a slower convergence towards the inflow velocity beyond this point.

The offset from $r/R = 0$ to $r/R = 1.0$ might be due to a slightly higher vortex strength. Using a slightly lower vortex strength would shift the curve upwards and reduce the peak value slightly, resulting in a closer match. However, based on the discussion presented in Subsection 5.4.4, it was concluded that the converged value of Γ varies depending on the ratio between X_{max} and R_{max}. Consequently, the accuracy of the velocity profile simulated with the Ainslie wake model should not be taken as absolute. Rather, the focus should be on analysing the velocity profile's shape.

In general, it can be concluded that based on the comparison between the two normalised axial velocity profiles the implementation of the forcing term gives satisfactory results for the axial velocity profile.

5.4.3. Comparison with Results without Forcing Term Implementation

This section will compare the velocity profile obtained using the Ainslie wake model with a forcing implementation against the velocity profile obtained using the Ainslie wake model with the specified inflow profile at the actuator disk following the recommendations of the IEC 61400 standard [42].

The IEC 61400 recommendation prescribes the initial wake radius as specified in Eq. (5.42). Furthermore, at the inlet, a uniform velocity profile is applied according to Eq. (5.43), where a_m denotes the rotor-averaged induction. The relationship between the thrust coefficient and the induction is expressed in Eq. (5.44).

It is important to note that, for the simulation, instead of using the eddy viscosity prescribed by the IEC 61400 standard, a constant eddy viscosity of 1 m^2/s is employed, which is applicable for the near wake, based on the evolution $\nu_t(x)$ presented in [99].

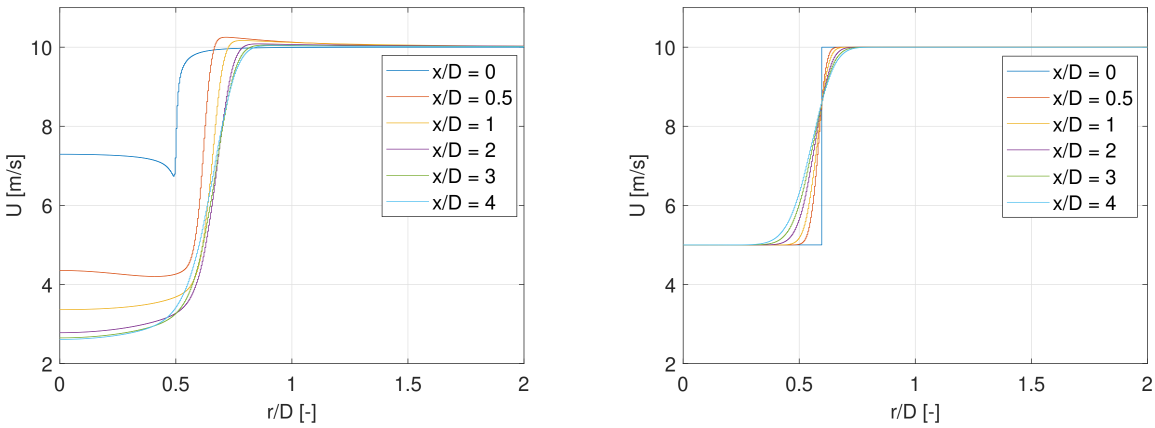
$$R_W(0) = 2R(1 - 0.45a_m^2)\sqrt{\frac{1+m}{8}} \quad \text{with} \quad m = \frac{1}{\sqrt{1 - C_T}} \tag{5.42}$$

$$U_W(0) = \begin{cases} U_\infty(1 - 2a_m), & r < R_W \\ U_\infty, & r > R_W \end{cases} \tag{5.43}$$

$$C_T = 4a_m(1 - a_m) \tag{5.44}$$

For the simulation, the turbine radius was set equal to the one of the DTU 10MW wind turbine [82]. Furthermore, an inflow wind speed of 10 m/s was used, and a thrust coefficient, C_T, of 0.75 was chosen, resulting in an average axial induction factor, a_m, of 0.25, using Eq. (5.44). Moreover, the computational domain was identical to that used in the comparison presented in Subsection 5.4.2, specifically extending from $-40D$ to $40D$ in the axial direction and from $0D$ to $30D$ in the radial direction. However, the velocity profile was only computed up to $4D$ behind the wind turbine, as the use of a constant eddy viscosity of 1 m^2/s is not valid beyond this point, as derived from [99] following the IEC 64110 standard. Lastly, for the regularisation parameter again $\delta = 1$ was used, based on the largest step size near the actuator disk, which is 0.7146 m in the radial direction.

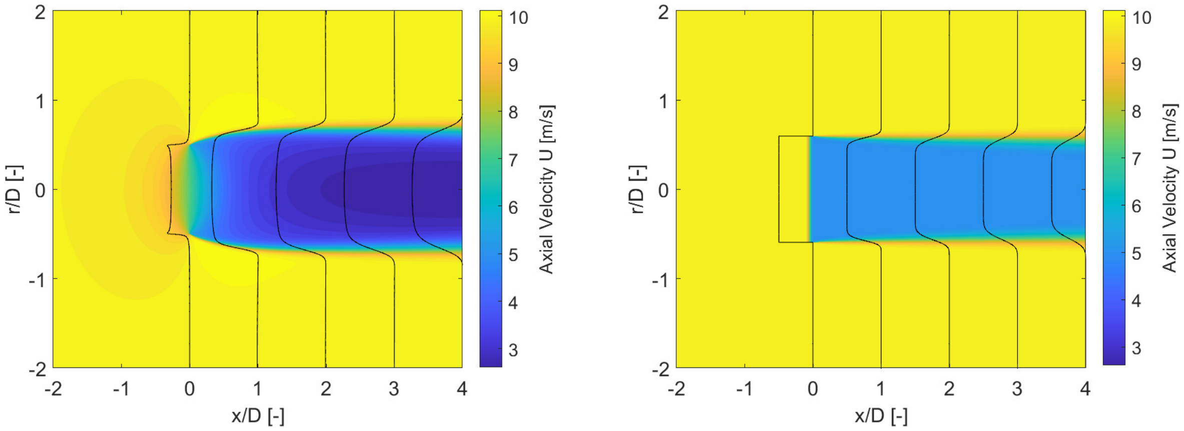
It is important to note that while the domain results in a converged value of Γ, the precise value of convergence is uncertain due to its dependency on the ratio between R_{max} and X_{max}, as discussed in Subsection 5.4.4. Therefore, the comparison is based on qualitative rather than quantitative results.



(a) Velocity profiles including forcing term.

(b) Velocity profiles excluding forcing term.

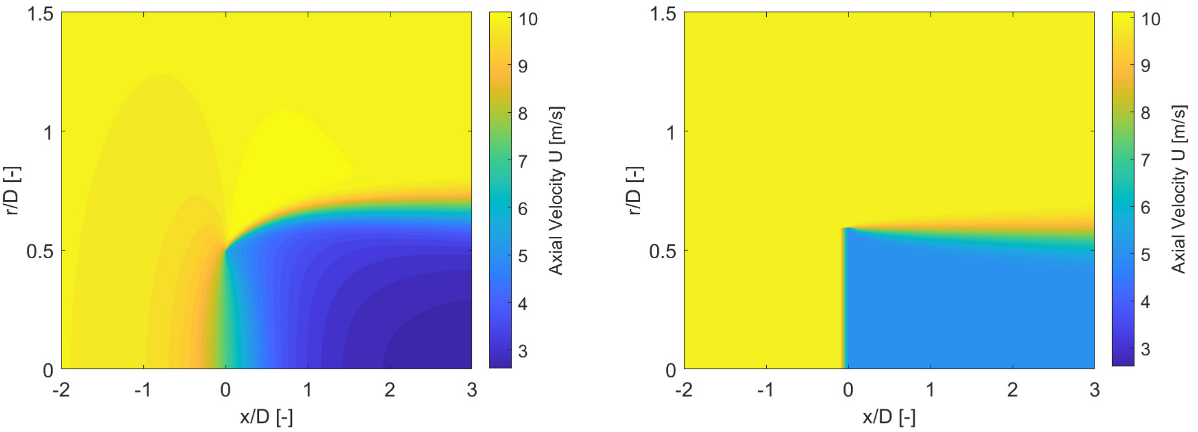
Figure 5.4: Comparison of the downstream axial velocity profiles from the forcing term implementation with the velocity profiles from the IEC 61400 standard.



(a) Velocity field and profiles including forcing term.

(b) Velocity field and profiles excluding forcing term.

Figure 5.5: Comparison of the downstream axial velocity field from the forcing term implementation with the velocity field from the IEC 61400 standard, where the black lines indicate the velocity deficit profiles at several downstream locations.



(a) Velocity field with forcing.

(b) Velocity field without forcing.

Figure 5.6: Comparison of the full axial velocity field from the forcing term implementation with the velocity field from the IEC 61400 standard.

Based on the results presented in Figs. 5.4 to 5.6, several key observations can be made. Starting with a comparison of the downstream velocity profiles, presented in Fig. 5.4, a significant difference can be observed. First, the axial velocity at the rotor disk, at $x/D = 0$, is significantly higher for the case including the forcing term. This is expected as in reality a velocity of $U_\infty(1 - a_m)$ is expected at the actuator disk rather than a velocity of $U_\infty(1-2a_m)$ as prescribed by the IEC 61400 standard. In contrast, the velocity in the near wake is significantly lower when the forcing term is implemented compared to when no forcing term is used, despite the wake expansion being very similar. This discrepancy suggests a notable difference in the momentum deficit, highlighting significant variations in the applied forcing, even though the applied thrust coefficient, C_T, is the same in both scenarios. However, the uncertainty in Γ complicates the determination of whether the applied forcing is truly identical in both cases.

Then, immediately downstream of the actuator disk, near the edge, the axial velocity increases, when the forcing term is included as observed from Figs. 5.4a and 5.6a. This phenomenon, known as the edge force effect, is caused by the pressure difference between the upstream and downstream sides of the actuator disk. This difference in pressure induces a radial flow toward the edge of the disk, leading to an acceleration of the flow [100].

The wake radius at the actuator disk observed in Figs. 5.5a and 5.6a is slightly smaller and closer to the actual wind turbine radius compared to the wake radius at the actuator observed in Figs. 5.5b and 5.6b, which again results in a better representation of reality. This indicates that the wake radius can be more accurately predicted at the actuator disk when a forcing term is implemented rather than using a prescribed formulation of the wake radius.

Lastly, it can be observed from Fig. 5.6a that the flow already starts to slow down before reaching the wind turbine. This indicates that the model also functions as a blockage model when the forcing is included. Specifically, the potential formulation of the forcing term provides insights into the effects not only at the rotor disk but also upstream and downstream of the rotor disk due to its widespread influence in the domain. The current formulation of the model following the IEC 61400 standard in contrast does not consider the effects upstream of the wind turbine as can be seen from Fig. 5.6b. This demonstrates a significant advantage of including the forcing term, as this behaviour more accurately reflects the flow dynamics observed in reality. Additionally, the extent of the blockage effect aligns with expectations based on literature [101, 102].

5.4.4. Limitations Forcing Formulation

While generating the results, it was observed that altering the domain size significantly affected the total force within the domain. This is undesirable, as the total force should remain constant, or converge when domain dimensions are extended to infinity.

Furthermore, based on this observation, it is evident that the chosen approach to determine the value of Γ, as presented in Subsection 5.3.4, will result in variations of Γ depending on the specific domain employed, introducing inaccuracies. This highlights a drawback of this method. In contrast, the alternative approach, which is independent of the total force within the domain, is expected to exhibit greater robustness against variations in the domain boundaries, ensuring more consistent calculations of the vortex strength. However, due to time constraints, this is not explored further and is left for future work.

To address this issue, first, the convergence of the integrated axial forcing term will be examined in Subsection 5.4.4.1. Furthermore, in order to understand and explain the observations, the asymptotic behaviour of the integrated axial forcing term will be analysed, starting from the mathematical formulation. Subsection 5.4.4.2 will focus on the asymptotic behaviour for changes in the domain size in the radial direction. This is followed by a discussion on the asymptotic behaviour for a change in the axial dimension of the domain, as presented in Subsection 5.4.4.3.

5.4.4.1. Convergence Forcing Term

To examine the convergence of the total axial force within the domain, the total axial force across the domain was calculated for exponentially increasing radial and axial outer bounds. Specifically, the outer bounds were doubled for each total force calculation, with $X_{max} = -X_{min} = R_{max}$.

Furthermore, to reduce the computational cost, advantage can be taken of the symmetry of the axial force across the actuator disk ($f_x(r,x) = f_x(r,-x)$). Specifically, for equal absolute values of X_{max} and X_{min}, this symmetry means the integral only needs to be calculated for half the domain and then multiplied by two, as illustrated in Fig. 5.7.

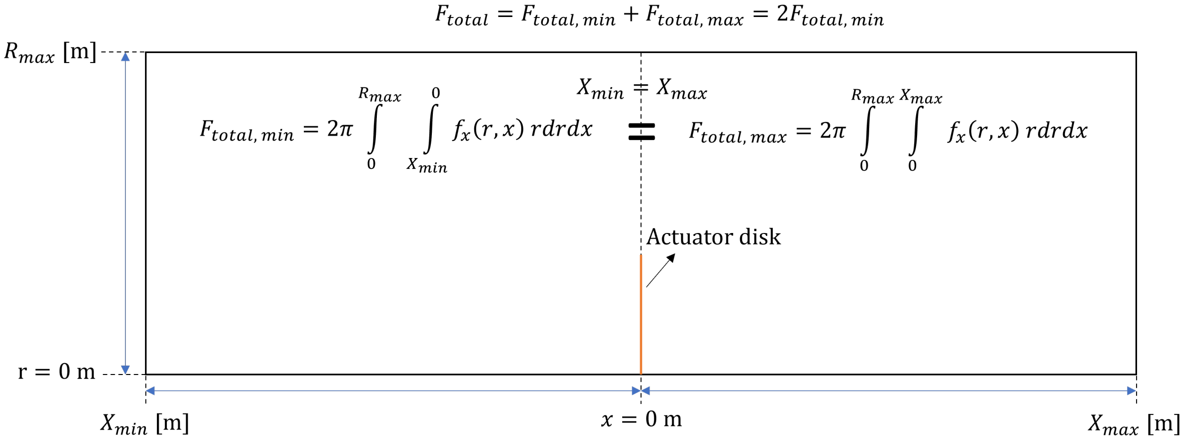
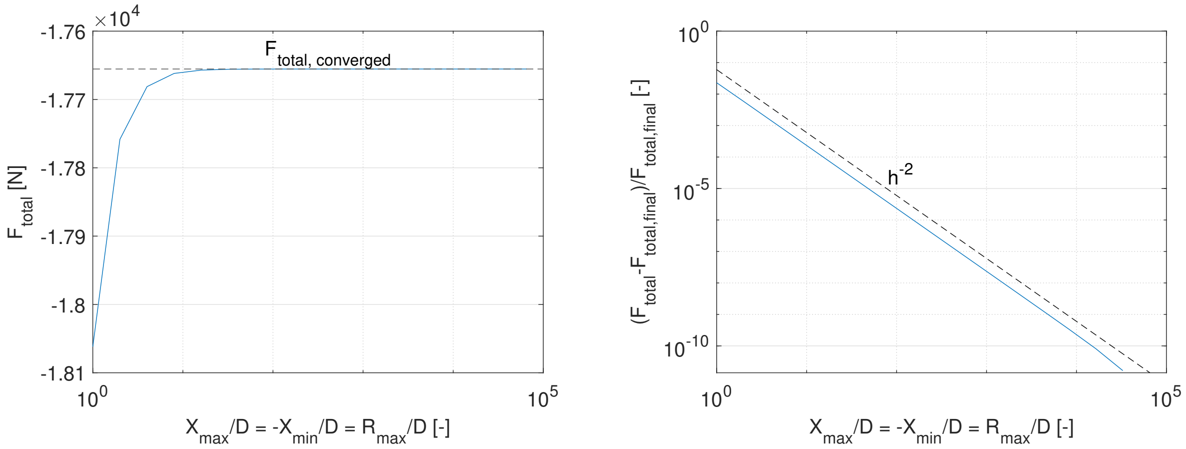


Figure 5.7: Representation of the domain, including definitions for the outer radial and axial bounds, and the equations used to calculate the total axial force.

The results, presented in Fig. 5.8a, are obtained for a vortex ring with a radius $R = 89.15$ m, based on the specifications for the DTU 10MW turbine. Additionally, the regularisation constant, δ, was fixed to 1. These results demonstrate that convergence is achieved when the domain bounds are increased to infinity, as required. Furthermore, the rate of convergence, shown in Fig. 5.8b, indicates that convergence occurs with an order of $\mathcal{O}(h^{-2})$, meaning that as the domain bounds are expanded, the error in the forcing decreases at a rate proportional to h^{-2}.

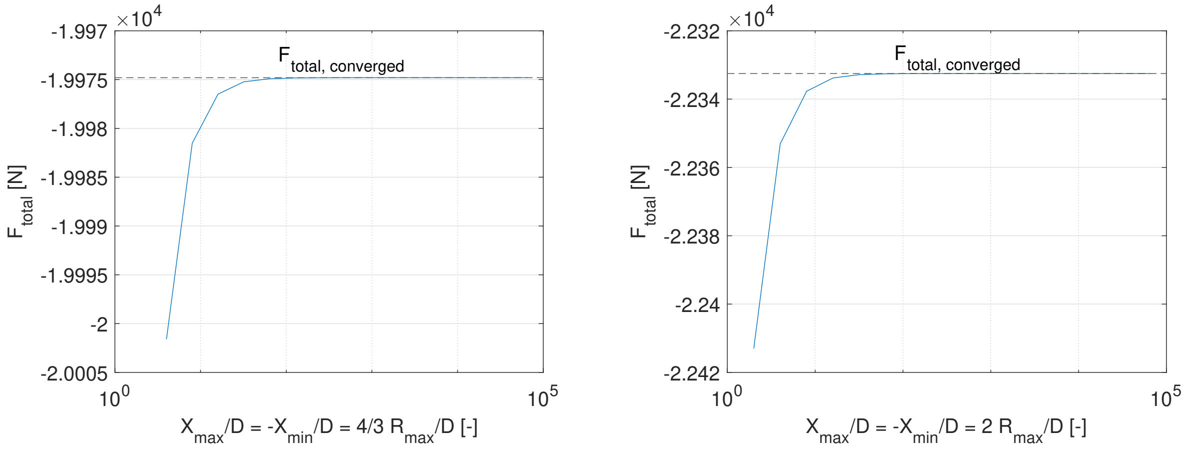


(a) Total axial force for $X_{max} = R_{max}$.

(b) Convegence Rate.

Figure 5.8: Change of the total axial force and the convergence rate for an increasing domain size.

Additionally, it has been observed that the convergence varies depending on the ratio between R_{max} and X_{max}, even for significantly large domains. To gain deeper insights into this behavior, the force convergence has been calculated for multiple ratios of the domain bounds.



(a) Total axial force for $X_{max} = 4/3R_{max}$.

(b) Total axial force for $X_{max} = 2R_{max}$.

Figure 5.9: Variation of the total axial force with increasing domain size when X_{max} is not equal to R_{max}.

The results presented in Fig. 5.9 demonstrate that altering the ratio between X_{max} and R_{max} affects the value to which the total force converges. Specifically, a lower R_{max}/X_{max} ratio results in a higher converged total force, and subsequently a lower value of Γ, as can be deducted from Eq. (5.37).

5.4.4.2. Asymptotic Behaviour of the Forcing for a Change in Radial Domain Dimension

To assess the asymptotic behaviour for an increase in the radial domain size, the value of the axial coordinate x, in the axial forcing term, is fixed at a constant value C. Consequently, the contribution of x can be neglected, allowing the analysis to focus solely on the effect of r. Furthermore, both the regularisation constant, δ, and the turbine radius, R, are neglected. Applying these assumptions to the axial force component outlined in Eq. (5.11) results in Eq. (5.45).

$$
\begin{aligned}
&f_x(r, C) = \frac{\Gamma}{2\pi r}\left[\frac{-r^2}{r^2}E\left(\frac{4r}{r^2}\right) + K\left(\frac{4r}{r^2}\right)\right] \\
&\text{where} \quad \frac{\Gamma}{2\pi r} \sim \frac{1}{r}, \ \frac{-r^2}{r^2} \sim 1, \ \text{and } m = \frac{4r}{r^2} \sim \frac{1}{r}
\end{aligned} \tag{5.45}
$$

To further analyse the dependency of the axial force on r, the next step involves applying a Taylor series expansion to $-E(m) + K(m)$ to assess its dependency on r. The resulting expression is presented in Eq. (5.46), where the higher order terms are neglected as these decay faster with increasing r. Based on the resulting Taylor expansion expansion it is deduced that the axial force component changes with $\frac{1}{r^2}$.

$$
-E(m) + K(m) \approx \frac{\pi}{4}m + \mathcal{O}(m^2) \sim \frac{\pi}{4}\frac{1}{r} + \mathcal{O}\left(\frac{1}{r^2}\right) \tag{5.46}
$$

Then, the resulting integral over the radial dimension is examined to understand the asymptotic behaviour of the total axial force for r increasing to infinity for a fixed value of x. Based on the integral presented in Eq. (5.47), it is concluded that the total axial force at a specific upstream or downstream position never converges, as $\ln(R)$ diverges when the radial domain size increases to infinity.

$$
F_{total} = \int_0^{R_{max}} f_x(r, 0) r \mathrm{d}r \sim \int_0^{R_{max}} \frac{1}{r^2} r \mathrm{d}r \sim \ln(R_{max}) \tag{5.47}
$$

To illustrate this asymptotic behaviour, the axial induced forcing was integrated over the radial direction for increasing values of the radial outer bound, R_{max}, at various fixed values of the axial coordinate x. This is indicated in Fig. 5.10 by the blue lines and the corresponding equations.

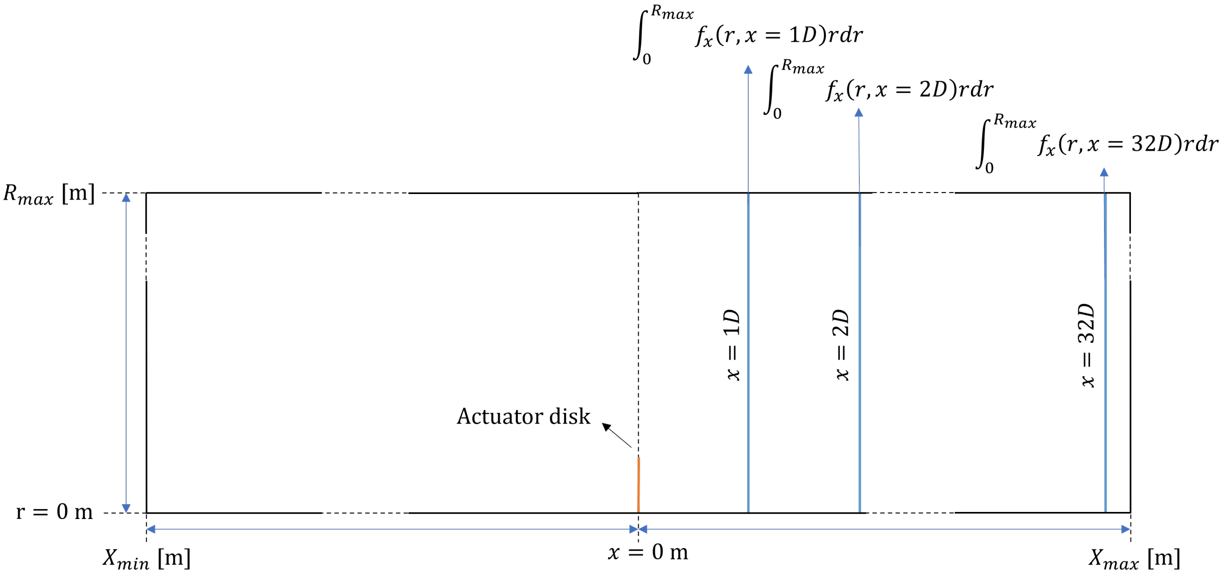
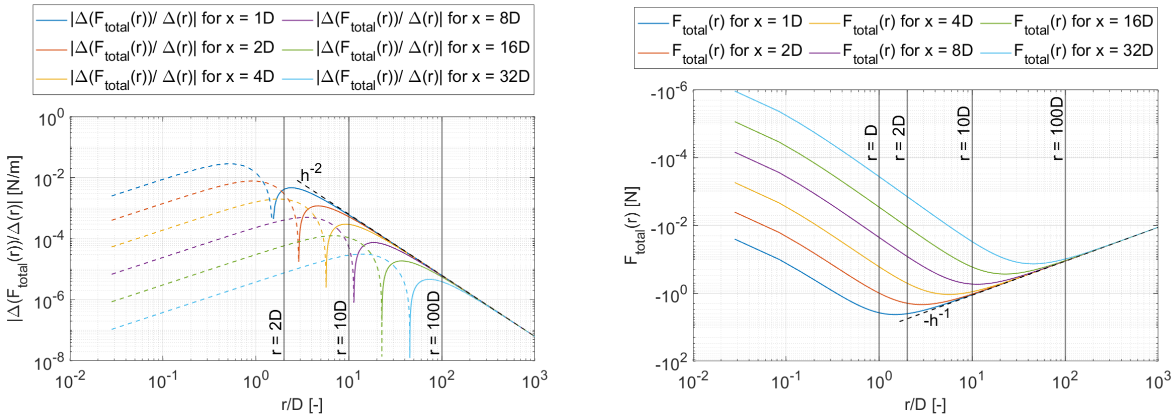


Figure 5.10: Representation of the domain with the outer radial and axial bounds, indicating the lines over which the axial force is integrated, along with the corresponding equations for the total axial force calculation to investigate the asymptotic behaviour in the radial direction.



(a) Effect on absolute change in total axial force. **(b)** Effect on the total axial force.

Figure 5.11: Effect of a change in the domain size in the radial direction on the total axial force and the absolute change in axial force. For effect on the absolute change in total axial force the dotted lines represent the original negative values and solid lines represent the positive values.

The resulting absolute change in total axial force per unit increase in radial domain size, $\Delta F_{total}(r)/\Delta r$, and the total axial force for fixed values of x are presented in Fig. 5.11. Based on these results two types of behaviour can be distinguished. First, from Fig. 5.11a an increase in the absolute change of the total axial force near the disk is observed, meaning the total axial force will diverge for increasing radial outer bound. This diverging behaviour is undesirable, and it can be observed that the range over which divergence occurs increases for larger values of x. Then, beyond this initially divergent region the absolute change in the total axial force decreases, meaning that the total axial force starts to converge, as desired. However, the total axial force only converges with the order of $\mathcal{O}(r^{-1})$, which results in a relatively slow convergence rate. From this, it can be concluded that even though the expected divergence is present, slow convergence is obtained for larger radial outer bounds, and is obtained fastest for low values of x.

Furthermore, it should be noted that the total axial force and the absolute change in total axial force presented here are not equal to the total axial force and the absolute change in total axial force within the full domain and that these calculations are only valid to investigate the asymptotic behaviour.

5.4.4.3. Asymptotic Behaviour of the Forcing for a Change in Axial Domain Dimension

In addition to examining the absolute change in total axial force due to a variation in the radial domain dimension, the effect of increasing the domain in the axial direction will also be studied. For this purpose, both the absolute change in total axial force per unit increase in axial domain size, $|\Delta F_{total}(x)/\Delta x|$, and the total axial force for varying axial domain sizes will be presented, for fixed values of r, as indicated by the green lines in Fig. 5.12 and the corresponding equations for calculating the total axial force.

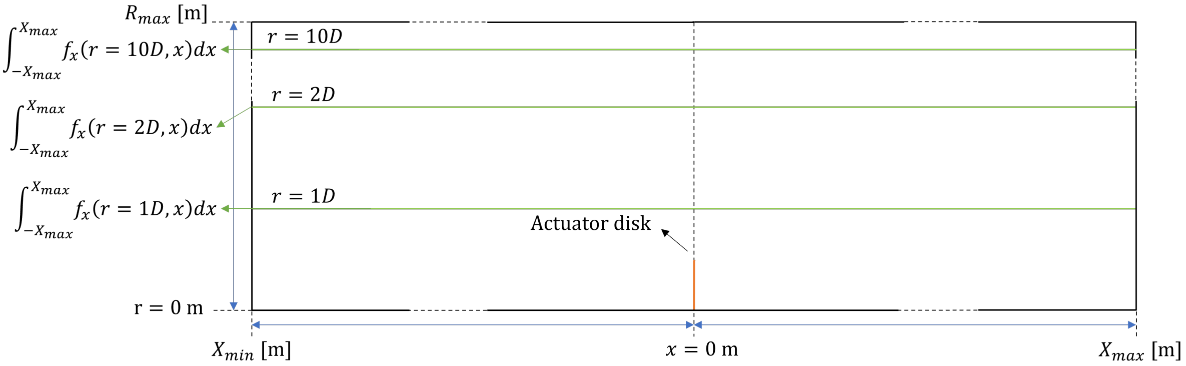


Figure 5.12: Representation of the domain with the outer radial and axial bounds, indicating the lines over which the axial force is integrated, along with the corresponding equations for the total axial force calculation to investigate the asymptotic behaviour in the axial direction.

Before presenting the results for this case, the expected asymptotic behaviour is derived. The effect of altering the axial domain size is assessed using a similar approach as for changing the radial direction, but with the distinction that the value of the radial coordinate r is held constant, $r = C$ m, and can therefore be neglected for this analysis. This assumption, combined with the previously outlined ones in Subsection 5.4.4.2, leads to the formulation of the axial force component described in Eq. (5.48).

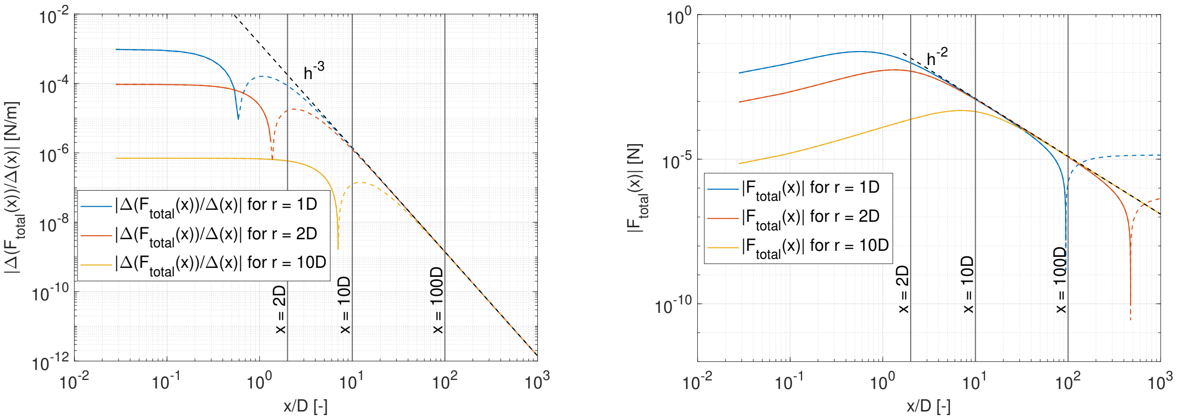
$$\begin{aligned} f_x(C,x) &= \frac{\Gamma}{2\pi x^2}\left[\frac{-x^2}{x^2}E\left(\frac{4}{x}\right)+K\left(\frac{4}{x^2}\right)\right] \\ \text{where}\quad &\frac{\Gamma}{2\pi x}\sim\frac{1}{x},\ \frac{-x^2}{x^2}\sim 1,\ \text{and } m=\frac{4}{x^2}\sim\frac{1}{x^2} \end{aligned} \tag{5.48}$$

To further analyse the asymptotic behaviour of the axial force on x, the next step involves employing the Taylor series expansion of $-E(m)+K(m)$ to assess its dependency on x. Based on the Taylor expansion outlined in Eq. (5.49), it is deduced that the axial force component changes with $\frac{1}{x^2}$. Furthermore, the higher order terms in the Taylor series expansion are neglected for this analysis as these decay faster with increasing x.

$$-E(m)+K(m)\approx\frac{\pi}{4}m+\mathcal{O}(m^2)\sim\frac{\pi}{4}\frac{1}{x^2}+\mathcal{O}\left(\frac{1}{x^4}\right) \tag{5.49}$$

Subsequently, the outcome of integrating the axial forcing term over the axial direction to compute the total axial force is depicted in Eq. (5.50). This result indicates that the total axial force converges with the order of $\mathcal{O}(x^{-2})$ as the axial domain size increases towards infinity. Furthermore, the change itself is expected to converge with the order of $\mathcal{O}(x^{-3})$.

$$F_{total}=\int_0^{X_{max}} f_x(C,x)\mathrm{d}x\sim\int_0^{X_{max}}\frac{1}{x^3}\mathrm{d}x\sim-\frac{1}{2X_{max}^2} \tag{5.50}$$



(a) Effect on absolute change in total axial force. **(b)** Effect on the total axial force.

Figure 5.13: Effect of a change in the domain size in the axial direction on the total axial force and the absolute change in axial force. The dotted lines represent the negative values and solid lines represent the positive values.

The results shown in Fig. 5.13 illustrate the expected trends for both the absolute change in axial force and the total axial force across various radial positions as the axial domain size increases. Furthermore, the absence of any increase in the absolute change in axial force indicates that there is no divergence present.

Then, depending on the radial position, notable differences are observed near the disk at $x = 0$. These differences are due to the localised effect of the actuator disk, which varies with radial distance. Specifically, the sign of the induced axial force changes at a certain position upstream of the wind turbine (and downstream due to the symmetry of the axial induced force across the actuator disk). The distance from the disk at which this change occurs varies with radial position, leading to discrepancies across different radial directions.

Additionally, for larger values of x, discrepancies in the total axial force are observed between different lines. This is because zero crossings, as indicated by the dotted lines, which can not be accurately represented in a logarithmic plot, causing variations in the total axial force across different radial positions.

5.4.5. Conclusion Implementation Forcing Term

Based on the results obtained from implementing the forcing term, it can be concluded that using potential theory provides a strong initial framework for representing the actuator disk's effect on the flow, as discussed in Subsection 5.4.2. However, the current formulation for axially induced force converges rather slowly because the potential formulation spreads the influence of the axial force across the entire domain. Therefore, it is recommended to explore alternative formulations for the forcing that ensure faster convergence of the forcing or provide a more localised effect, as these are expected to yield more satisfactory results.

Given the limitations of the initial forcing term derived from the Biot-Savart law, as discussed in Subsection 5.4.4, it is advised to implement the alternative forcing representation, presented in Subsection 5.1.2. This allows for a straightforward comparison of their performance. Such an evaluation will offer insights into their respective strengths and weaknesses, informing potential refinements to the formulation and implementation of the forcing term within the study's framework.

6

Conclusions and Recommendations

The objective of this thesis was to establish the groundwork for a computationally efficient wake model by solving the axisymmetric and stationary Navier-Stokes equations using a finite element method with the ultimate goal to utilise the outputs from this model to conduct wind turbine load calculations.

This chapter addresses the research questions outlined in Section 1.1, provides general conclusions drawn from the research, and offers recommendations based on these findings. It is structured into two sections. Firstly, Section 6.1 delves into answering the research questions and presenting an overall conclusion. Subsequently, in Section 6.2, suggestions and considerations for future work are discussed.

6.1. Conclusions

The primary objective of this research was to develop a faster and more stable numerical method for axisymmetric wake modelling. This aims to accelerate current wake profile calculations and enhance their robustness across various parameter settings, with the future goal of using the obtained wake profiles as input for wind turbine load calculations. For this purpose, the Ainslie wake model was chosen as the foundation. To solve the governing nonlinear partial differential equations of the Ainslie wake model, the finite element method was utilised with Hermite interpolation polynomials as basis functions in both the radial and axial directions. Furthermore, a marching scheme was employed in the axial direction instead of building a global solver.

The finite element implementation method explored in this thesis has been found to be unstable for solving the Ainslie wake model. As a consequence of the instability observed in the finite element implementation method for solving the Ainslie wake model, the research questions originally posed in this thesis cannot be satisfactorily answered. The conclusion of the study will therefore focus on discussing the underlying reasons for the issues encountered during the FEM implementation and on the conclusions that can be drawn from the implementation of the forcing term in the finite volume method Ainlie wake model.

Implementing Hermite interpolation basis functions has presented unique challenges. Increasing the order of Hermite interpolation basis functions has revealed significant issues of ill-conditioning, diminishing their suitability for this specific application. Additionally, higher-order Hermite interpolation basis functions have shown increased sensitivity to the employed mesh. This emphasises the importance of mesh quality when increasing the order of the basis functions. Moreover, the presence of both even and odd functions when using a higher-order basis has rendered them ineffective for solving the continuity equation, as the coefficients in the matrices no longer align with the area under the functions after summation along the projection index. In contrast, the piecewise linear basis functions have proven to be compatible with the continuity equation and have shown more robustness when solving the diffusion term, particularly with respect to the employed mesh and in terms of the condition number.

However, despite the expectation that the approach using piecewise linear basis functions would yield satisfactory results, stability issues were encountered. The exact cause of these stability issues remains unknown. It is anticipated, though, that they may be related to the LBB condition not being satisfied. The LBB condition is crucial for ensuring stability when solving the steady incompressible Navier-Stokes equations or Stokes equations, which are closely related to the governing equation of the Ainslie wake model.

While computational times for the Ainslie wake model using the finite element method remain uncertain due to instability, insights can be drawn from computational times for the diffusion equation presented in Subsection 4.2.4. These times indicate that higher-order basis functions require more computational resources due to the increased matrix sizes involved in simulations. This suggests a trade-off between the accuracy gained from higher-order basis functions, contingent on their suitability for the application and mesh quality, and the corresponding increase in computational costs. Moreover, these times notably exceed those required for running the Ainslie wake model with a finite volume method. It is important to note, however, that the model used for simulating the diffusion equation has not yet been optimised for computational efficiency, unlike the fully optimised finite volume Ainslie wake model. Therefore, these computational times should not be used for direct comparisons with other commercial codes at this stage.

The implementation of the forcing in the finite volume Ainslie wake model has demonstrated conservation of mass and momentum. Additionally, the axial velocity profile aligns with those presented in literature using similar force formulations, verifying the correctness of the implementation strategy. However, due to the limitations observed in the forcing formulations, no definitive conclusions can be drawn about the quantitative quality of the implementation. Specifically, the value of the vortex strength was shown to be highly dependent on the domain size and the ratio between the axial and radial outer bounds, complicating the determination of the correct converged value for Γ.

Finally, based on comparisons of the velocity field and profiles with current IEC 61400 standard, the implementation of the forcing term has been demonstrated to more accurately reflect the flow dynamics observed in reality. Specifically, the forcing implementation has shown to represent the flow dynamics upstream of the rotor by capturing the decrease in axial velocity known as the blockage effect and the extent of this blockage effect is consistent with expectations based on the literature [101, 102]. Additionally, it has demonstrated to capture the edge force effect caused by pressure differences before and after the actuator disk.

6.2. Recommendations

Increasing the order of the Hermite interpolation basis functions has revealed significant ill-conditioning, indicating that higher-order Hermite interpolation basis functions are not well-suited for this specific application. However, this does not imply that these basis functions are unsuitable for all finite element applications. In fact, these functions can be effective when the equations form the steady-state solution to the problem, as demonstrated in [103]. However, it is recommended to explore alternative basis functions that demonstrate improved performance for higher-order finite element methods for this particular application. These new basis functions should be even in nature, as uneven basis functions have been shown to introduce difficulties near boundaries when solving the incompressible continuity equation.

Furthermore, when employing higher-order basis functions, the quality of the mesh becomes increasingly critical. The diffusive nature of the axial velocity profile means that the initially fitted mesh, which is suitable for the inflow condition at the actuator disk, becomes inadequate as the velocity profile diffuses. To address this, it is recommended to implement an adaptive mesh refinement strategy. This approach will ensure that the mesh adapts dynamically to the evolving wake structure. Alternatively, one can ensure that the mesh is sufficiently refined up to the expected wake radius.

The current implementation strategy, which uses piecewise linear basis functions, has exhibited instability issues, likely stemming from the violation of the LBB condition. To address this, adopting a mixed finite element approach is recommended, which might offer a stable solver. Specifically, due to similarities between the governing equations of the Ainslie wake model and incompressible Navier-Stokes equations, implementing Taylor-Hood elements is suggested. In a typical Taylor-Hood scheme, quadratic basis functions are used for velocity, while linear basis functions are used for pressure. While the original Ainslie wake model does not initially account for pressure, incorporating this aspect represents a significant step towards achieving the broader objective.

The current approach to define Γ has shown to be highly dependent on the chosen domain size due to the extent of the vortex rings's influence domain. Additionally, the ratio between the outer domain boundaries has been shown to significantly affect the value to which Γ converges. Consequently, it is recommended to explore the alternative approach outlined in Subsection 5.3.4 for defining Γ, as it is independent of the domain size. Furthermore, it is advisable to investigate whether the alternative force approximation presented in Subsection 5.1.2 provides a more localised force field. If successful, this approach would become the preferred choice.

In reality, the actuator disk does not have constant loading. Therefore, it is recommended to enhance the force formulation to accommodate an actuator disk with multiple annuli, each characterised by its own local thrust coefficient. If using the vortex ring force formulation, this can be achieved by placing a vortex ring at each annulus, adjusting the value of Γ based on the change in C_T. One should however not forget the limitations in defining the value for Γ based on a given C_T. Furthermore, implementing a non-uniform actuator disk with the second force formulation presented in Subsection 5.1.2 is more complex.

Additionally, it is essential to implement a complete Poisson solver that incorporates the component of the Laplacian of the pressure due to velocity to assess the influence of this term. Near the disk, it is anticipated that this term will be overshadowed by the significantly larger forcing component. However, at greater distances from the disk, this term may exert a more noticeable influence. This implementation represents a step toward achieving greater realism in wake modelling, as the Ainslie model without the addition of a forcing term or a full Poisson solver is essentially a non-linear diffusion equation, sacrificing detailed dynamics for computational speed.

Finally, it is recommended to compare the velocity profiles obtained from current engineering models with those obtained from the Ainslie wake model incorporating a forcing term, or ideally incorporating a Poisson solver, at greater downstream distances. This comparison is crucial because current engineering models are less accurate in the near wake due to their underlying assumptions. However, for an accurate comparison, the non-constant eddy viscosity formulation as presented in the IEC 61400 standard should be implemented. Following this comparison, a further step towards reality would involve a comparison with a non-constant loaded actuator disk model.

References

[1] V. Z. Castillo et al. "Future global electricity demand load curves". In: *Energy* 258.124741 (2022). DOI: `https://doi.org/10.1016/j.energy.2022.124741`. URL: `https://www.scopus.com/inward/record.uri?eid=2-s2.0-85135930644&doi=10.1016%2fj.energy.2022.124741&partnerID=40&md5=fcd43a096849d8095d955a4b9eecb607`.

[2] Wind Europe. *Wind energy in Europe: Scenarios for 2030.* Sept. 2017.

[3] B. Knopf et al. "The European renewable energy target for 2030 An impact assessment of the electricity sector". In: *Energy Policy* 85 (2015), pp. 50–60. DOI: `https://doi.org/10.1016/j.enpol.2015.05.010`. URL: `https://www.sciencedirect.com/science/article/pii/S0301421515002037`.

[4] Y. Hu et al. "Computing turbulent far-wake development behind a wind turbine with and without swirl". In: *Wind and Structures, An International Journal* 15.1 (2012), pp. 17–26. DOI: `10.12989/was.2012.15.1.017`. URL: `https://www.scopus.com/inward/record.uri?eid=2-s2.0-84862953633&doi=10.12989%2fwas.2012.15.1.017&partnerID=40&md5=55bfc6dae089e09e96c5b36f9325e8c4`.

[5] Wind Europe. *Wind energy in Europe: 2021 Statistics and the outlook for 2022-2026.* Feb. 2022.

[6] L. J. Vermeer et al. "Wind turbine wake aerodynamics". In: *Progress in aerospace sciences* 39.6-7 (2003), pp. 467–510.

[7] M. Méchali et al. "Wake effects at Horns Rev and their influence on energy production". In: *European wind energy conference and exhibition.* Vol. 1. Citeseer Princeton, NJ, USA. 2006, pp. 10–20.

[8] K. Thomsen et al. "Fatigue loads for wind turbines operating in wakes". In: *Journal of Wind Engineering and Industrial Aerodynamics* 80.1-2 (1999), pp. 121–136. DOI: `10.1016/S0167-6105(98)00194-9`. URL: `https://www.scopus.com/inward/record.uri?eid=2-s2.0-0033086886&doi=10.1016%2fS0167-6105%2898%2900194-9&partnerID=40&md5=732e4970a0939818d716c33c9c3d60fe`.

[9] T. S. Lokken. "Semi-Infinite Laguerre Functions for Wind Turbine Wake Modeling". en. Delft: Delft University of Technology, Mar. 2023. URL: `http://resolver.tudelft.nl/uuid:2d489831-9099-442d-b236-e03ed63088b4`.

[10] J. Annoni et al. "Wind farm flow modeling using an input-output reduced-order model". In: *2016 american control conference (acc).* IEEE. 2016, pp. 506–512.

[11] J. A. Frederik et al. "The helix approach: Using dynamic individual pitch control to enhance wake mixing in wind farms". In: *Wind Energy* 23.8 (2020), pp. 1739–1751.

[12] C. H. Whiting et al. "A stabilized finite element method for the incompressible Navier–Stokes equations using a hierarchical basis". In: *International Journal for Numerical Methods in Fluids* 35.1 (2001), pp. 93–116.

[13] B. Sanderse et al. "Review of computational fluid dynamics for wind turbine wake aerodynamics". In: *Wind energy* 14.7 (2011), pp. 799–819.

[14] J. N. Sørensen. "2.08 - Aerodynamic Analysis of Wind Turbines". In: *Comprehensive Renewable Energy.* Ed. by Ali Sayigh. Oxford: Elsevier, 2012, pp. 225–241. DOI: `https://doi.org/10.1016/B978-0-08-087872-0.00209-2`. URL: `https://www.sciencedirect.com/science/article/pii/B9780080878720002092`.

[15] M. O. L. Hansen et al. "State of the art in wind turbine aerodynamics and aeroelasticity". In: *Progress in aerospace sciences* 42.4 (2006), pp. 285–330.

[16] M. O. L. Hansen. *Aerodynamics of wind turbines: Third edition*. English. Aerodynamics of Wind Turbines: Third Edition. Journal Abbreviation: Aerodyn. of Wind Turbines: Third Ed. Pages: 173 Publication Title: Aerodyn. of Wind Turbines: Third Ed. Taylor and Francis Inc., 2015. DOI: `10.4324/9781315769981`. URL: `https://www.scopus.com/inward/record.uri?eid=2-s2.0-84942810624&doi=10.4324%2f9781315769981&partnerID=40&md5=c08ad530b6986a344970decae66f6f5f`.

[17] Y.-T. Wu et al. “Simulation of turbulent flow inside and above wind farms: model validation and layout effects”. In: *Boundary-layer meteorology* 146 (2013), pp. 181–205.

[18] L. Martinez et al. “A comparison of actuator disk and actuator line wind turbine models and best practices for their use”. In: *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. 2012, p. 900.

[19] L. A. Martínez-Tossas et al. “Filtered lifting line theory and application to the actuator line model”. In: *Journal of Fluid Mechanics* 863 (2019), pp. 269–292.

[20] S. Schmitz et al. “Modeling the wakes of wind turbines and rotorcraft using the actuator-line method in an openfoam-les solver”. In: *Proceedings of the 69th AHS Forum, Phoenix, AZ, USA*. 2013, pp. 21–23.

[21] M. J. Churchfield et al. “An advanced actuator line method for wind energy applications and beyond”. In: *35th Wind Energy Symposium*. 2017, p. 1998.

[22] W. Z. Shen et al. “Actuator surface model for wind turbine flow computations”. In: *Proceedings of European Wind Energy Conference and Exhibition*. Vol. 7. 10. 2007.

[23] C. Sibuet Watters et al. “Application of the actuator surface concept to wind turbine rotor aerodynamics”. In: *Wind Energy* 13.5 (2010), pp. 433–447.

[24] G. Tuhfe et al. “Wind turbine wake models developed at the technical university of Denmark: A review”. In: *Renewable and Sustainable Energy Reviews* 60 (2016), pp. 752–769. DOI: `https://doi.org/10.1016/j.rser.2016.01.113`. URL: `https://www.sciencedirect.com/science/article/pii/S136403211600143X`.

[25] E. M. de Boer. “Analyzing the wake of a wind turbine actuated with Dynamic Induction Control”. en. Delft: Delft University of Technology, Aug. 2021. URL: `http://resolver.tudelft.nl/uuid:6470bdc1-d9c2-4ed2-b3f2-6854a7daa5f0`.

[26] F. Porté-Agel et al. “Wind-Turbine and Wind-Farm Flows: A Review”. en. In: *Boundary-Layer Meteorology* 174.1 (Jan. 2020), pp. 1–59. DOI: `10.1007/s10546-019-00473-0`. URL: `http://link.springer.com/10.1007/s10546-019-00473-0` (visited on 11/05/2023).

[27] J. K. Kaldellis et al. “Critical evaluation of Wind Turbines analytical wake models”. In: *Renewable and Sustainable Energy Reviews* 144 (2021), p. 110991.

[28] H. Snel. “Review of the present status of rotor aerodynamics”. In: *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology* 1.S1 (1998), pp. 46–69.

[29] H. Glauert. “Airplane propellers”. In: *Aerodynamic theory* (1935).

[30] W. Z. Shen et al. “Tip loss corrections for wind turbine computations”. In: *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology* 8.4 (2005), pp. 457–475.

[31] R. E. Wilson et al. “Applied aerodynamics of wind power machines”. In: *National Science Foundation* (1974).

[32] Y. El Khchine et al. “Tip Loss Factor Effects on Aerodynamic Performances of Horizontal Axis Wind Turbine”. en. In: *Energy Procedia* 118 (Aug. 2017), pp. 136–140. DOI: `10.1016/j.egypro.2017.07.028`. URL: `https://linkinghub.elsevier.com/retrieve/pii/S1876610217325547` (visited on 11/12/2023).

[33] D. Micallef et al. “An investigation of radial velocities for a horizontal axis wind turbine in axial and yawed flows”. In: *Wind Energy* 16.4 (2013), pp. 529–544.

[34] N. Charhouni et al. “Qualification of Three Analytical Wake Models”. en. In: *Journal of Mechanics Engineering and Automation* 6.4 (Apr. 2016). DOI: `10.17265/2159-5275/2016.04.006`. URL: `http://www.davidpublisher.com/index.php/Home/Article/index?id=27362.html` (visited on 11/09/2023).

[35] J. W. Lin et al. “New engineering wake model for wind farm applications”. In: *Renewable Energy* 198 (2022), pp. 1354–1363. DOI: `https://doi.org/10.1016/j.renene.2022.08.116`. URL: `https://www.sciencedirect.com/science/article/pii/S0960148122012915`.

[36] N. O. Jensen. *A note on wind generator interaction.* Vol. 2411. Citeseer, 1983.

[37] I. Katic et al. “A simple model for cluster efficiency”. In: *European wind energy association conference and exhibition.* Vol. 1. A. Raguzzi Rome, Italy. 1986, pp. 407–410.

[38] M. Bastankhah et al. “A new analytical model for wind-turbine wakes”. In: *Renewable energy* 70 (2014), pp. 116–123.

[39] S. Frandsen et al. “Analytical modelling of wind speed deficit in large offshore wind farms”. In: *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology* 9.1-2 (2006), pp. 39–53.

[40] T. Ishihara et al. “A new Gaussian-based analytical wake model for wind turbines considering ambient turbulence intensities and thrust coefficient effects”. In: *Journal of Wind Engineering and Industrial Aerodynamics* 177 (2018), pp. 275–292.

[41] H. Zong et al. “A momentum-conserving wake superposition method for wind farm power prediction”. In: *Journal of Fluid Mechanics* 889 (2020), A8.

[42] International Electrotechnical Commission (IEC). *Wind turbine generator systems - Part 1: Safety requirements.* 4th ed. ISBN: 123-456-789-0. IEC, Feb. 2019.

[43] E. Branlard. “Wind turbine aerodynamics and vorticity-based methods: Fundamentals and recent applications”. In: (2017).

[44] J. T. Reid et al. “General approach to lifting-line theory, applied to wings with sweep”. In: *Journal of Aircraft* 58.2 (2021), pp. 334–346.

[45] S. G. Voutsinas et al. “Investigation of the yawed operation of wind turbines by means of a vortex particle method”. In: *AGARD CONFERENCE PROCEEDINGS AGARD CP.* AGARD Neuilly sur Seine, France. 1995, pp. 11–11.

[46] B. Sanderse. *Aerodynamics of wind turbine wakes.* Petten: ECN, 2009.

[47] H. Snel. “Review of aerodynamics for wind turbines”. In: *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology* 6.3 (2003), pp. 203–211.

[48] S. van der Pijl. “Numerical modelling of wind farm aerodynamics”. In: *Technical Report ECN Internal, Energy Research Centre of the Netherlands* (2007).

[49] Y. Zhiyin. “Large-eddy simulation: Past, present and the future”. In: *Chinese journal of Aeronautics* 28.1 (2015), pp. 11–24.

[50] C. Meneveau et al. “Scale-invariance and turbulence models for large-eddy simulation”. In: *Annual Review of Fluid Mechanics* 32.1 (2000), pp. 1–32.

[51] G. C. Larsen et al. *Dynamic wake meandering modeling.* English. Denmark. Forskningscenter Risoe. Risoe-R 1607(EN). Risø National Laboratory, 2007.

[52] H. Aagaard Madsen et al. “Calibration and validation of the dynamic wake meandering model for implementation in an aeroelastic code”. In: (2010).

[53] I. Reinwardt et al. “Validation of wind turbine wake models with focus on the dynamic wake meandering model”. In: *Journal of Physics: Conference Series*. Vol. 1037. 7. IOP Publishing. 2018, p. 072028.

[54] J. F. Ainslie. “Calculating the flowfield in the wake of wind turbines”. In: *Journal of wind engineering and Industrial Aerodynamics* 27.1-3 (1988), pp. 213–224.

[55] T. J. Larsen et al. “Validation of the dynamic wake meander model for loads and power production in the Egmond aan Zee wind farm”. In: *Wind Energy* 16.4 (2013), pp. 605–624.

[56] P. A. Taylor. “On wake decay and row spacing for WECS farms”. In: *3rd International Symposium on Wind Energy Systems*. 1980, pp. 451–468.

[57] M.-K. Liu et al. “Mathematical model for the analysis of wind-turbine wakes”. In: *Journal of Energy* 7.1 (1983), pp. 73–78.

[58] A. Crespo et al. “Survey of modelling methods for wind turbine wakes and wind farms”. In: *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology* 2.1 (1999), pp. 1–24.

[59] M. Magnusson. “Near-wake behaviour of wind turbines”. In: *Journal of Wind Engineering and Industrial Aerodynamics* 80.1-2 (1999), pp. 147–167.

[60] A. Duckworth et al. “Investigation and validation of wind turbine wake models”. In: *Wind Engineering* 32.5 (2008), pp. 459–475.

[61] H. Kim et al. “Wind turbine wake characterization for improvement of the Ainslie eddy viscosity wake model”. In: *Energies* 11.10 (2018), p. 2823.

[62] G. V. Iungo et al. “Parabolic RANS solver for low-computational-cost simulations of wind turbine wakes”. In: *Wind Energy* 21.3 (2018), pp. 184–197.

[63] T. J. Larsen et al. “Validation of the dynamic wake meander model for loads and power production in the Egmond aan Zee wind farm”. In: *Wind Energy* 16.4 (2013), pp. 605–624.

[64] S. Ott et al. *Linearised CFD models for wakes*. Danmarks Tekniske Universitet, Risø Nationallaboratoriet for Bæredygtig Energi, 2011.

[65] A. Peña et al. “On wake modeling, wind-farm gradients, and AEP predictions at the Anholt wind farm”. In: *Wind Energy Science* 3.1 (2018), pp. 191–202.

[66] J. Shen et al. *Spectral methods: algorithms, analysis and applications*. Vol. 41. Springer Science & Business Media, 2011.

[67] J. Peiró et al. “Finite difference, finite element and finite volume methods for partial differential equations”. In: *Handbook of Materials Modeling: Methods*. Springer, 2005, pp. 2415–2446.

[68] M. Zeneli et al. “Numerical methods for solid-liquid phase-change problems”. In: *Ultra-high temperature thermal energy storage, transfer and conversion*. Elsevier, 2021, pp. 165–199.

[69] J. Blazek. *Computational fluid dynamics: principles and applications*. Butterworth-Heinemann, 2015.

[70] A. Madhlopa. “Chapter 10 - Modeling and optimization of solar receivers”. In: *Solar Receivers for Thermal Power Generation*. Ed. by A. Madhlopa. Academic Press, 2022, pp. 297–369. DOI: `https://doi.org/10.1016/B978-0-323-85271-5.00010-0`. URL: `https://www.sciencedirect.com/science/article/pii/B9780323852715000100`.

[71] O. Kolditz. “Finite Volume Method”. In: *Computational Methods in Environmental Fluid Mechanics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 173–190. DOI: `10.1007/978-3-662-04761-3_8`. URL: `https://doi.org/10.1007/978-3-662-04761-3_8`.

[72] J. M. Maljaars. “When Euler meets Lagrange: Particle-Mesh Modeling of Advection Dominated Flows”. en. PhD Thesis. Delft: Delft University of Technology, Dec. 2019. DOI: `10.4233/UUID:A400512D-966D-402A-A40A-FEDF60ACF22C`. URL: `http://resolver.tudelft.nl/uuid:a400512d-966d-402a-a40a-fedf60acf22c` (visited on 10/31/2023).

[73] S. C. Brenner et al. “Finite element methods”. In: *Encyclopedia of computational mechanics* 1 (2004), pp. 73–114.

[74] R. J. Labeur. “Finite element modelling of transport and non-hydrostatic flow in environmental fluid mechanics”. In: (2009).

[75] A. Kumari et al. “Survey of Hermite Interpolating Polynomials for the Solution of Differential Equations”. In: *Mathematics* 11.14 (2023), p. 3157.

[76] D. N. Arnold et al. “Unified analysis of discontinuous Galerkin methods for elliptic problems”. In: *SIAM journal on numerical analysis* 39.5 (2002), pp. 1749–1779.

[77] T. J. R. Hughes et al. “A comparison of discontinuous and continuous Galerkin methods based on error estimates, conservation, robustness and efficiency”. In: *Discontinuous Galerkin Methods: Theory, Computation and Applications.* Springer. 2000, pp. 135–146.

[78] K. Shahbazi et al. “A high-order discontinuous Galerkin method for the unsteady incompressible Navier-Stokes equations”. en. In: *Journal of Computational Physics* 222.1 (Mar. 2007), pp. 391–407. DOI: `10.1016/j.jcp.2006.07.029`. URL: `https://linkinghub.elsevier.com/retrieve/pii/S0021999106003585` (visited on 11/09/2023).

[79] C. Dawson et al. “Continuous, discontinuous and coupled discontinuous–continuous Galerkin finite element methods for the shallow water equations”. In: *International Journal for Numerical Methods in Fluids* 52.1 (2006), pp. 63–88.

[80] A. Dodgson. *Discontinuous Galerkin methods.* Accessed on January 4, 2024. 2018. URL: `https://mat1.uibk.ac.at/heiko/na18/`.

[81] B. Granzow. *Dirichlet boundary conditions for nonlinear systems.* Accessed: 25-3-2024. May 2017.

[82] C. Bak et al. “Description of the DTU 10 MW reference wind turbine”. In: *DTU Wind Energy Report-I-0092* 5 (2013).

[83] M. Wu et al. “Hermite type spline spaces over rectangular meshes with complex topological structures”. In: *Communications in Computational Physics* 21.3 (2017), pp. 835–866.

[84] M. Bacciocchi et al. “On the mapping procedure based on higher-order Hermite polynomials for laminated thin plates with arbitrary domains in gradient elasticity”. In: *International Journal for Numerical Methods in Engineering* 124.11 (2023), pp. 2552–2570.

[85] Y. Li et al. “Compact feature-aware Hermite-style high-order surface reconstruction”. In: *Engineering with Computers* 37.1 (2021), pp. 187–210.

[86] J. M. Burgerscentrum. “Finite element methods for the incompressible Navier-Stokes equations”. In: (2015).

[87] LONG C. “Finite Element Methods for Stokes Equations”. In: ().

[88] E. Süli. “A brief excursion into the mathematical theory of mixed finite element methods”. In: *Lecture Notes, University of Oxford* (2013), pp. 24–29.

[89] O. C. Zienkiewicz et al. *The finite element method: its basis and fundamentals.* Elsevier, 2005.

[90] J. Burkardt. “Finite Elements for the (Navier) Stokes Equations”. In: *Numerical Analysis Seminar, Department of Scientific Computing, Florida State USA University.* Vol. 8. 2011.

[91] G. A. M. Van Kuik et al. “Potential flow solutions for energy extracting actuator disc flows”. In: *Wind Energy* 19.8 (2016), pp. 1391–1406.

[92] S. J. Newman. “The induced velocity of a vortex ring filament”. In: (2011).

[93] J. B. De Vaal et al. “Validation of a vortex ring wake model suited for aeroelastic simulations of floating wind turbines”. In: *Journal of Physics: Conference Series.* Vol. 555. 1. IOP Publishing. 2014, p. 012025.

[94] S. S. Yoon et al. “Analytical formulas for the velocity field induced by an infinitely thin vortex ring”. In: *Int. J. Numer. Meth. Fluids* 44 (2004), pp. 665–672.

[95] M. Abramowitz et al. “Handbook of mathematical functions Dover Publications”. In: *New York* 361 (1965).

[96] E. Branlard. “Vortex and Source Rings”. In: *Wind Turbine Aerodynamics and Vorticity-Based Methods: Fundamentals and Recent Applications* (2017), pp. 419–428.

[97] R. Shankar Subramanian. *Lagrange and Stokes Streamfunctions.* Accessed: 28-5-2024.

[98] H. Dumitrescu et al. “Wind turbine aerodynamic performance by lifting line method”. In: *International Journal of Rotating Machinery* 4.3 (1998), pp. 141–149.

[99] N. G. W. Warncke et al. “Wake Modelling Based on the Stokes Stream Function and the Toroidal-Poloidal Decomposition”. In: *Journal of Physics: Conference Series.* Vol. 2767. 5. IOP Publishing. 2024, p. 052028.

[100] G. A. M. van Kuik. “On the limitations of Froude’s actuator disc concept”. In: (1991).

[101] E. S. P. Branlard et al. “Using a cylindrical vortex model to assess the induction zone infront of aligned and yawed rotors”. In: *EWEA Offshore 2015 Conference.* European Wind Energy Association (EWEA). 2015.

[102] E. S. P. Branlard et al. “Wind farm blockage effects: comparison of different engineering models”. In: *Journal of Physics: Conference Series.* Vol. 1618. 6. IOP Publishing. 2020, p. 062036.

[103] D. Funaro et al. “Stability and conservation properties of Hermite-based approximations of the Vlasov-Poisson system”. In: *Journal of Scientific Computing* 88.1 (2021), p. 29.

A

Matrices Continuity Equation

In this Appendix, the matrices that constitute the discretised divergence operator, as provided in Eq. (4.21), are presented. First, Appendix A.1 presents the matrices obtained when employing piecewise linear basis functions in the radial direction. Then, Appendix A.2 shows the matrices for the case when utilising cubic Hermite interpolation basis functions in the radial direction. Lastly, Appendix A.3 provides an overview of the matrices obtained when employing quintic Hermite interpolation basis functions in the radial direction. Moreover, the mesh used to obtain these matrices has 3 radial nodes and 2 nodes in the axial direction, with a spacing of 1 m between each node.

A.1. Matrices when Employing Piecewise Linear Basis Functions in the Radial Direction

The continuity equation as presented in Eq. (4.21) consists of four distinct matrices. The first matrix that is presented in Eq. (A.1) is associated with the term $\frac{\partial U}{\partial x}$ where U is known inflow velocity, as denoted by $U_{(n,k_l)}$ in Eq. (4.21), while the second matrix, provided in Eq. (A.3), is also related to the term $\frac{\partial U}{\partial x}$, but now U represents the unknown velocity coefficients, as denoted by $U_{(n,k_r)}$ in Eq. (4.21). Furthermore, the vectors resulting from a summation along the projection index for these two matrices are given in Eqs. (A.2) and (A.4), respectively.

$$\begin{bmatrix} -0.1667 & -0.0833 & 0 & 0 \\ -0.0833 & -0.3333 & -0.0833 & 0 \\ 0 & -0.0833 & -0.3333 & -0.0833 \\ 0 & 0 & -0.0833 & -0.1667 \end{bmatrix} \tag{A.1}$$

$$\begin{bmatrix} -0.2500 & -0.5000 & -0.5000 & -0.2500 \end{bmatrix} \tag{A.2}$$

$$\begin{bmatrix} 0.1667 & 0.0833 & 0 & 0 \\ 0.0833 & 0.3333 & 0.0833 & 0 \\ 0 & 0.0833 & 0.3333 & 0.0833 \\ 0 & 0 & 0.0833 & 0.1667 \end{bmatrix} \tag{A.3}$$

$$\begin{bmatrix} 0.2500 & 0.5000 & 0.5000 & 0.2500 \end{bmatrix} \tag{A.4}$$

Then, the two matrices related to the term $\frac{\partial V}{\partial y}$ in the continuity equation, associated with the known and unknown transverse velocity coefficients are presented in Eqs. (A.5) and (A.7), respectively. Moreover, the vectors resulting from the summation along the projection index are provided in Eqs. (A.6) and (A.8), respectively.

$$\begin{bmatrix} -0.0833 & 0.0833 & 0 & 0 \\ -0.0833 & 0 & 0.0833 & 0 \\ 0 & -0.0833 & 0 & 0.0833 \\ 0 & 0 & -0.0833 & 0.0833 \end{bmatrix} \tag{A.5}$$

$$\begin{bmatrix} -0.1667 & 0 & 0 & 0.1667 \end{bmatrix} \tag{A.6}$$

$$\begin{bmatrix} -0.1667 & 0.1667 & 0 & 0 \\ -0.1667 & 0 & 0.1667 & 0 \\ 0 & -0.1667 & 0 & 0.1667 \\ 0 & 0 & -0.1667 & 0.1667 \end{bmatrix} \tag{A.7}$$

$$\begin{bmatrix} -0.3333 & 0 & 0 & 0.3333 \end{bmatrix} \tag{A.8}$$

A.2. Matrices when Employing Cubic Hermite Interpolation in the Radial Direction

Similar as to the matrices presented in Appendix A.1, also here the four matrices constituting the continuity equation will be presented, but know for the case when employing cubic Hermite interpolation functions in the radial direction. The first matrix, shown in Eq. (A.9), is associated with the term $\frac{\partial U}{\partial x}$ where U is known inflow velocity, as denoted by $U_{(n,k_l)}$ in Eq. (4.21). The second matrix, provided in Eq. (A.11), is also related to the term $\frac{\partial U}{\partial x}$, but here U represents the unknown velocity coefficients, denoted as $U_{(n,k_r)}$ in Eq. (4.21). Additionally, the vectors resulting from a summation along the projection index for these two matrices are given in Eqs. (A.10) and (A.12), respectively.

$$\begin{bmatrix} -0.1857 & -0.0262 & -0.0643 & 0.0155 & 0 & 0 & 0 & 0 \\ -0.0262 & -0.0048 & -0.0155 & 0.0036 & 0 & 0 & 0 & 0 \\ -0.0643 & -0.0155 & -0.3714 & 0 & -0.0643 & 0.0155 & 0 & 0 \\ 0.0155 & 0.0036 & 0 & -0.0095 & -0.0155 & 0.0036 & 0 & 0 \\ 0 & 0 & -0.0643 & -0.0155 & -0.3714 & 0 & -0.0643 & 0.0155 \\ 0 & 0 & 0.0155 & 0.0036 & 0 & -0.0095 & -0.0155 & 0.0036 \\ 0 & 0 & 0 & 0 & -0.0643 & -0.0155 & -0.1857 & 0.0262 \\ 0 & 0 & 0 & 0 & 0.0155 & 0.0036 & 0.0262 & -0.0048 \end{bmatrix} \tag{A.9}$$

$$\begin{bmatrix} -0.2607 & -0.0429 & -0.5000 & -0.0024 & -0.5000 & -0.0024 & -0.2393 & 0.0405 \end{bmatrix} \tag{A.10}$$

$$\begin{bmatrix} 0.1857 & 0.0262 & 0.0643 & -0.0155 & 0 & 0 & 0 & 0 \\ 0.0262 & 0.0048 & 0.0155 & -0.0036 & 0 & 0 & 0 & 0 \\ 0.0643 & 0.0155 & 0.3714 & 0 & 0.0643 & -0.0155 & 0 & 0 \\ -0.0155 & -0.0036 & 0 & 0.0095 & 0.0155 & -0.0036 & 0 & 0 \\ 0 & 0 & 0.0643 & 0.0155 & 0.3714 & 0 & 0.0643 & -0.0155 \\ 0 & 0 & -0.0155 & -0.0036 & 0 & 0.0095 & 0.0155 & -0.0036 \\ 0 & 0 & 0 & 0 & 0.0643 & 0.0155 & 0.1857 & -0.0262 \\ 0 & 0 & 0 & 0 & -0.0155 & -0.0036 & -0.0262 & 0.0048 \end{bmatrix} \tag{A.11}$$

$$\begin{bmatrix} 0.2607 & 0.0429 & 0.5000 & 0.0024 & 0.5000 & 0.0024 & 0.2393 & -0.0405 \end{bmatrix} \tag{A.12}$$

Furthermore, the two matrices related to the term $\frac{\partial V}{\partial y}$ in the continuity equation, associated with the known and unknown transverse velocity coefficients, denoted by $V_{(n,k_l)}$ and $V_{(n,k_r)}$ in Eq. (4.21) respectively, are presented in Eqs. (A.13) and (A.15). Moreover, the vectors resulting from the summation along the projection index are provided in Eqs. (A.14) and (A.16), respectively.

$$\begin{bmatrix} -0.0833 & 0.0167 & 0.0833 & -0.0167 & 0 & 0 & 0 & 0 \\ -0.0167 & 0 & 0.0167 & -0.0028 & 0 & 0 & 0 & 0 \\ -0.0833 & -0.0167 & 0 & 0.0333 & 0.0833 & -0.0167 & 0 & 0 \\ 0.0167 & 0.0028 & -0.0333 & 0 & 0.0167 & -0.0028 & 0 & 0 \\ 0 & 0 & -0.0833 & -0.0167 & 0 & 0.0333 & 0.0833 & -0.0167 \\ 0 & 0 & 0.0167 & 0.0028 & -0.0333 & 0 & 0.0167 & -0.0028 \\ 0 & 0 & 0 & 0 & -0.0833 & -0.0167 & 0.0833 & 0.0167 \\ 0 & 0 & 0 & 0 & 0.0167 & 0.0028 & -0.0167 & 0 \end{bmatrix} \tag{A.13}$$

$$\begin{bmatrix} -0.1667 & 0.0028 & -1.0408e-17 & -8.6736e-19 & -1.0408e-17 & -8.6736e-19 & 0.1667 & -0.0028 \end{bmatrix} \tag{A.14}$$

$$\begin{bmatrix} -0.1667 & 0.0333 & 0.1667 & -0.0333 & 0 & 0 & 0 & 0 \\ -0.0333 & 0 & 0.0333 & -0.0056 & 0 & 0 & 0 & 0 \\ -0.1667 & -0.0333 & 0 & 0.0667 & 0.1667 & -0.0333 & 0 & 0 \\ 0.0333 & 0.0056 & -0.0667 & 0 & 0.0333 & -0.0056 & 0 & 0 \\ 0 & 0 & -0.1667 & -0.0333 & 0 & 0.0667 & 0.1667 & -0.0333 \\ 0 & 0 & 0.0333 & 0.0056 & -0.0667 & 0 & 0.0333 & -0.0056 \\ 0 & 0 & 0 & 0 & -0.1667 & -0.0333 & 0.1667 & 0.0333 \\ 0 & 0 & 0 & 0 & 0.0333 & 0.0056 & -0.0333 & 0 \end{bmatrix} \tag{A.15}$$

$$\begin{bmatrix} -0.3333 & 0.0056 & -2.0817e-18 & -1.7347e-18 & -2.0817e-18 & -1.7347e-18 & 0.3333 & -0.0056 \end{bmatrix} \tag{A.16}$$

A.3. Matrices when Employing Cubic Hermite Interpolation in the Radial Direction

This section will present the matrices constituting the continuity equation in Eq. (4.21) when employing quintic Hermite interpolation basis functions. First, the matrices related to the term $\frac{\partial U}{\partial x}$, specifically associated with the known and unknown velocity coefficients, are provided in Eqs. (A.17) and (A.19), respectively. Moreover, the corresponding vectors obtained from a summation along their projection index are outlined in Eqs. (A.18) and (A.20), respectively. Lastly, the two matrices related to the term $\frac{\partial V}{\partial y}$ in the continuity equation are presented. These matrices, provided in Eqs. (A.21) and (A.23), correspond to the known and unknown transverse velocity coefficients, respectively. The corresponding vectors resulting from a summation along the projection index are provided in Eqs. (A.22) and (A.24).

$$\begin{bmatrix}
-0.1959 & -0.0337 & -0.0025 & -0.0541 & 0.0163 & -0.0016 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.0337 & -0.0075 & -0.0006 & -0.0163 & 0.0048 & -0.0005 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.0025 & -0.0006 & -5.4113e-05 & -0.0016 & 0.0005 & -4.5094e-05 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.0541 & -0.0163 & -0.0016 & -0.3918 & 0 & -0.0051 & -0.0541 & 0.0163 & -0.0016 & 0 & 0 & 0 \\
0.0163 & 0.0048 & 0.0005 & 0 & -0.01501 & 0 & -0.0163 & 0.0048 & -0.0005 & 0 & 0 & 0 \\
-0.0016 & -0.0005 & -4.5094e-05 & -0.0051 & 0 & -0.0001 & -0.0016 & 0.0005 & -4.5094e-05 & 0 & 0 & 0 \\
0 & 0 & 0 & -0.0541 & -0.0163 & -0.0016 & -0.3918 & 0 & -0.0051 & -0.0541 & 0.01634 & -0.0016 \\
0 & 0 & 0 & 0.0163 & 0.0048 & 0.0005 & 0 & -0.0150 & 0 & -0.0163 & 0.0048 & -0.0005 \\
0 & 0 & 0 & -0.0016 & -0.0005 & -4.5094e-05 & -0.0051 & 0 & -0.0001 & -0.0016 & 0.0005 & -4.5094e-05 \\
0 & 0 & 0 & 0 & 0 & 0 & -0.0541 & -0.0163 & -0.0016 & -0.1959 & 0.0337 & -0.0025 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.0163 & 0.0048 & 0.0005 & 0.0337 & -0.0075 & 0.0006 \\
0 & 0 & 0 & 0 & 0 & 0 & -0.0016 & -0.0005 & -4.5094e-05 & -0.0025 & 0.0006 & -5.4113e-05
\end{bmatrix} \tag{A.17}$$

$$\begin{bmatrix} -0.2715 & -0.0538 & -0.0044 & -0.5083 & -0.0054 & -0.0085 & -0.5083 & -0.0054 & -0.0085 & -0.2369 & 0.0484 & -0.0041 \end{bmatrix} \tag{A.18}$$

$$\begin{bmatrix}
0.1959 & 0.0337 & 0.0025 & 0.0541 & -0.0163 & 0.0016 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.0337 & 0.0075 & 0.0006 & 0.0163 & -0.0048 & 0.0005 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.0025 & 0.0006 & 5.4113e-05 & 0.0016 & -0.0005 & 4.5094e-05 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.0541 & 0.0163 & 0.0016 & 0.3918 & 0 & 0.0051 & 0.0541 & -0.0163 & 0.0016 & 0 & 0 & 0 \\
-0.0163 & -0.0048 & -0.0005 & 0 & 0.0150 & 0 & 0.0163 & -0.0048 & 0.0005 & 0 & 0 & 0 \\
0.0016 & 0.0005 & 4.5094e-05 & 0.0051 & 0 & 0.0001 & 0.0016 & -0.0005 & 4.5094e-05 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.0541 & 0.0163 & 0.0016 & 0.3918 & 0 & 0.0051 & 0.0541 & -0.0163 & 0.0016 \\
0 & 0 & 0 & -0.0163 & -0.0048 & -0.0005 & 0 & 0.0150 & 0 & 0.0163 & -0.0048 & 0.0005 \\
0 & 0 & 0 & 0.0016 & 0.0005 & 4.5094e-05 & 0.0051 & 0 & 0.0001 & 0.0016 & -0.0005 & 4.5094e-05 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.0541 & 0.0163 & 0.0016 & 0.1959 & -0.0337 & 0.0025 \\
0 & 0 & 0 & 0 & 0 & 0 & -0.0163 & -0.0048 & -0.0005 & -0.0337 & 0.0075 & -0.0006 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.0016 & 0.0005 & 4.5094e-05 & 0.0025 & -0.0006 & 5.4113e-05
\end{bmatrix} \tag{A.19}$$

$$\begin{bmatrix} 0.2715 & 0.0538 & 0.0044 & 0.5083 & 0.0054 & 0.0085 & 0.5083 & 0.0054 & 0.0085 & 0.2369 & -0.0484 & 0.0041 \end{bmatrix} \tag{A.20}$$

$$\begin{bmatrix}
-0.0833 & 0.0218 & 0.0020 & 0.0833 & -0.0218 & 0.0020 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.0218 & 0 & 0.0002 & 0.0218 & -0.0052 & 0.0004 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.0020 & -0.0002 & 0 & 0.0020 & -0.0004 & 3.3069e-05 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.0833 & -0.0218 & -0.0020 & 0 & 0.0437 & 0 & 0.0833 & -0.0218 & 0.0020 & 0 & 0 & 0 \\
0.0218 & 0.0052 & 0.0004 & -0.0437 & 0 & 0.0003 & 0.0218 & -0.0052 & 0.0004 & 0 & 0 & 0 \\
-0.0020 & -0.0004 & -3.3069e-05 & 0 & -0.0003 & 0 & 0.0020 & -0.0004 & 3.3069e-05 & 0 & 0 & 0 \\
0 & 0 & 0 & -0.0833 & -0.0218 & -0.0020 & 0 & 0.0437 & 0 & 0.0833 & -0.0218 & 0.0020 \\
0 & 0 & 0 & 0.0218 & 0.0052 & 0.0004 & -0.0437 & 0 & 0.0003 & 0.0218 & -0.0052 & 0.0004 \\
0 & 0 & 0 & -0.0020 & -0.0004 & -3.3069e-05 & 0 & -0.0003 & 0 & 0.0020 & -0.0004 & 3.3069e-05 \\
0 & 0 & 0 & 0 & 0 & 0 & -0.0833 & -0.0218 & -0.0020 & 0.0833 & 0.0218 & -0.0020 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.0218 & 0.0052 & 0.0004 & -0.0218 & 0 & 0.0002 \\
0 & 0 & 0 & 0 & 0 & 0 & -0.0020 & -0.0004 & -3.3069e-05 & 0.0020 & -0.0002 & 0
\end{bmatrix} \tag{A.21}$$

$$\begin{bmatrix} -0.1706 & 0.0046 & 0.0006 & -1.3878e-17 & -0.0012 & 0.0012 & -1.3878e-17 & -0.0012 & 0.0012 & 0.1706 & -0.0058 & 0.0006 \end{bmatrix} \tag{A.22}$$

$$\begin{bmatrix}
-0.1667 & 0.0437 & 0.0040 & 0.1667 & -0.0437 & 0.0040 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.0437 & 0 & 0.0003 & 0.0437 & -0.0103 & 0.0009 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.0040 & -0.0003 & 0 & 0.0040 & -0.0009 & 6.6138e-05 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.1667 & -0.0437 & -0.0040 & 0 & 0.0873 & 0 & 0.1667 & -0.0437 & 0.0040 & 0 & 0 & 0 \\
0.0437 & 0.0103 & 0.0009 & -0.0873 & 0 & 0.0007 & 0.0437 & -0.0103 & 0.0009 & 0 & 0 & 0 \\
-0.0040 & -0.0009 & -6.6138e-05 & 0 & -0.0007 & 0 & 0.0040 & -0.0009 & 6.6138e-05 & 0 & 0 & 0 \\
0 & 0 & 0 & -0.1667 & -0.0437 & -0.0040 & 0 & 0.0873 & 0 & 0.1667 & -0.0437 & 0.0040 \\
0 & 0 & 0 & 0.0437 & 0.0103 & 0.0009 & -0.0873 & 0 & 0.0007 & 0.0437 & -0.0103 & 0.0009 \\
0 & 0 & 0 & -0.0040 & -0.0009 & -6.6138e-05 & 0 & -0.0007 & 0 & 0.0040 & -0.0009 & 6.6138e-05 \\
0 & 0 & 0 & 0 & 0 & 0 & -0.1667 & -0.0437 & -0.0040 & 0.1667 & 0.0437 & -0.0040 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.0437 & 0.0103 & 0.0009 & -0.0437 & 0 & 0.0003 \\
0 & 0 & 0 & 0 & 0 & 0 & -0.0040 & -0.0009 & -6.6138e-05 & 0.0040 & -0.0003 & 0
\end{bmatrix} \tag{A.23}$$

$$\begin{bmatrix} -0.3413 & 0.00913 & 0.0011 & -2.7756e-17 & -0.0024 & 0.0024 & -2.7756e-17 & -0.0024 & 0.0024 & 0.3413 & -0.0115 & 0.0013 \end{bmatrix} \tag{A.24}$$

B

Gauss-Legendre Numerical Quadrature

Gauss-Legendre quadrature is a numerical integration technique used to approximate definite integrals. It is a form of Gaussian quadrature that utilises the roots of Legendre polynomials as sampling points. The equation governing Gauss-Legendre quadrature is provided in Eq. (B.1). Here $P_n(x)$ represents the n-th orthogonal Legendre polynomials, normalised such that $P_n(1) = 1$, and x_i denotes the i-th root of $P_n(x)$.

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx \sum_{i=1}^{n} w_i f(x_i) \quad \text{with} \quad w_i = \frac{2}{(1 - x_i^2)[P_n'(x_i)]^2} \tag{B.1}$$

However, the integral bounds must align with the mesh utilised in the finite volume method. Therefore, the Gauss-Legendre quadrature rule needs to be adapted to accommodate general integral bounds. This adaptation requires adjusting the weights and roots, as detailed in Eq. (B.2).

$$\int_{x_{min}}^{x_{max}} f(x)\mathrm{d}x \approx \sum_{i=1}^{n} w_i' f(x_i') \quad \text{with} \quad w_i' = \frac{x_{max} - x_{min}}{2} w_i; \; x_i' = \frac{x_{max} - x_{min}}{2} x_i + \frac{x_{max} + x_{min}}{2} \tag{B.2}$$