Politecnico di Torino

Master's Degree in Aerospace Engineering
A.y. 2025/2026
Graduation Session July 2026

Development and Assessment of Aerodynamic Solvers

Framework for Wings, Propellers, and Wake-Dominated Flows

Supervisor: prof. Domenic D'Ambrosio

VKI Supervisor: prof. Christophe Schram

Candidate:
Matteo Sanetti

von KARMAN INSTITUTE
FOR FLUID DYNAMICS

von Karman Institute For Fluid Dynamics

Chaussèe de Waterloo, 72
B-1640 Rhode-Saint-Genèse - Belgium

Department of Environmental and Applied Fluid Dynamics
Aeroacoustics and Low-Speed Aerodynamics Group

Short Training Program Final Report

Development and Assessment of Aerodynamic Solvers

Framework for Wings, Propellers, and Wake-Dominated Flows

Student:	Matteo Sanetti
VKI Supervisor:	prof. Christophe Schram
VKI Advisors:	prof. Grigorios Dimitriadis
	dr. Alessandro Zarri
	Ewann Passelac Estrada

Abstract

The development of computationally efficient aerodynamic tools is essential for the preliminary analysis of lifting surfaces, propellers, and complex configurations. While high-fidelity computational fluid dynamics (CFD) methods provide detailed flow predictions, their computational cost often limits their use in parametric studies and early-stage design. This motivates the development of lower-fidelity solvers able to retain relevant physical modelling while remaining computationally affordable. This thesis presents the development and assessment of two such solvers, implemented from scratch within an in-house numerical library at the von Karman Institute for Fluid Dynamics (VKI).

The first solver, VKI-BELL, is intended for steady propeller and rotor applications. It combines a blade-element treatment of sectional aerodynamic loads with a lifting-line formulation and a semi-prescribed wake model, resulting in a low-fidelity representation.

The second solver is VKI-SDPM, which represents the main methodological advancement of this work. It is a source–doublet singularity panel code that provides a more general framework applicable to wings, propellers, and rotors, while also supporting unsteady simulations. The solver also includes several viscous-core models, as well as vectorization and parallel-computing strategies aimed at keeping the computational cost low.

Both solvers were assessed through dedicated test cases and validated against experimental benchmark data from the literature. While VKI-BELL was evaluated on rotor configurations, VKI-SDPM was validated on both wing and rotor cases. The results show that both solvers can capture the main physical phenomena with acceptable levels of accuracy when applied within their proper domains of validity.

Both solvers remain computationally affordable. In VKI-BELL, this is mainly a consequence of its relatively simple formulation. In VKI-SDPM, instead, the affordability is enabled by the efficiency of its implementation and numerical architecture, with measured runtimes that fall within the range typically expected for unsteady aerodynamic solvers of similar complexity.

Overall, the developed solvers provide a flexible and expandable basis for preliminary aerodynamic investigations, with VKI-SDPM offering a framework for future extensions toward multibody aerodynamic interactions.

Acknowledgements

At the completion of this work, I wish to express my deepest gratitude to Prof. Domenic D'Ambrosio, who offered me the opportunity to carry out my thesis at the von Karman Institute, and to Prof. Christophe Schram, who supervised me during my stay. I also extend heartfelt thanks to Prof. Grigorios Dimitriadis: his guidance steered the project, and his availability, commitment, and constant help have been a true source of inspiration to me. I would also like to thank A. Zarri and E. Estrada for their support and assistance.

A special thanks goes to the people I met during these fourteen months on Belgian soil. Elisa, Daniele, Simone, and Matteo, you made Brussels a little sunnier. To the Sart-Tilman group—Rodolfo, Lucia, Giulia, Fatima, Anita, and Bara—with you I discovered the meaning of companionship and sharing; living together was unforgettable. A warm embrace to Michela, Leonardo, Dani, and all the Yesterday's scattered across Europe and beyond, architects of an Erasmus experience that will remain forever in my heart.

A due and indispensable thank-you goes to the brightest minds of the century: Cicco Dama, Luca, Sir Claudio Baldi Tenia, Massimo Prigione, Stefano Ecooltra, Er Piastra, and Gigi, known as Luì. Thank you for your enduring friendship.

A special thought is for Maria: this past year with you has been incredible. Between Athens, Rome, Brussels, and Liège, I realized that a person can also be home.

I dedicate this milestone to those to whom it truly belongs: my parents, who have always supported me with love and sacrifice, and to my roots—my grandparents Aldo, Anna, Italo, and Linda—whose example continues, and will always continue, to inspire me.

To all of you goes my deepest gratitude.

Table of Contents

List of Tables

List of Figures

Nomenclature

Symbol	Description	Unit
Latin alphabet		
a_{ind}	Axial induction factor	[–]
a'_{ind}	Tangential induction factor	[–]
$\boldsymbol{A}_{\phi}$	Body source potential influence matrix	[m]
a	Speed of sound	[m s^{-1}]
AR	Aspect ratio	[–]
b	Wing span	[m]
$\boldsymbol{B}_{\phi}$	Body doublet potential influence matrix	[–]
c	Chord length	[m]
$\boldsymbol{C}_{\phi}$	Wake doublet potential influence matrix	[–]
$\mathcal{C}$	Curve domain	[m]
C_D	Drag coefficient	[–]
C_d	Sectional drag coefficient	[–]
C_L	Lift coefficient	[–]
C_l	Sectional lift coefficient	[–]
C_M	Pitching moment coefficient	[–]
C_m	Sectional pitching moment coefficient	[–]
C_N	Normal force coefficient	[–]
C_n	Sectional normal force coefficient	[–]
C_P	Power coefficient	[–]
C_p	Pressure coefficient	[–]
C_Q	Torque coefficient	[–]
C_T	Thrust coefficient	[–]
d	Orthogonal distance	[m]
D	Rotor diameter	[m]
$\mathbf{dl}$	Bound segment unit vector	[–]
$\boldsymbol{D}$	Drag vector	[N]
$\hat{\mathbf{e}}_{\text{c}}$	Chord axis	[–]

Symbol	Description	Unit
E	Potential coefficient	[–]
$\hat{\mathbf{e}}_D$	Wind-frame drag/inflow axis	[–]
$\hat{\mathbf{e}}_L$	Wind-frame lift axis	[–]
E_{pw}	Parallel computation efficiency	[–]
$\hat{\mathbf{e}}_{\mathrm{PA}}$	Pitch axis	[–]
$\hat{\mathbf{e}}_{\mathrm{rot}}$	Rotor axis	[–]
F	Regularisation function	[–]
$\mathbf{F}$	Force vector	[N]
F_{tip}	Tip loss factor	[–]
f_{wp}	Workload parallel fraction	[–]
G	Green's function	[m^{-1}]
$\mathcal{G}_{\mathrm{B}}$	Body panel geometry assembly	[–]
$\tilde{\mathbf{G}}$	Perturbation velocities transformation matrix	[–]
$\mathbf{I}_{\mathrm{BS}}$	Biot–Savart kernel matrix	[–]
$\mathcal{I}$	Body panel topological assembly	[–]
J	Advance ratio	[–]
K_{w}	Wake-length factor	[–]
k_{c}	Wake contraction factor	[–]
k_{Lan}	Landgrebe wake contraction factor	[–]
$\boldsymbol{L}$	Lift vector	[N]
l	Length	[m]
$\hat{\mathbf{l}}$	Chordwise panel unit vector	[–]
$\mathbf{L}^{\star}$	Lower triangular matrix	[–]
$\mathcal{L}$	Logarithmic contribution to velocity potential	[m]
M	Mach number	[–]
$\mathbf{M}$	Moment vector	[Nm]
$\hat{\mathbf{m}}$	Panel spanwise unit vector	[–]
$\mathbf{n}$	Surface normal unit vector	[–]
N_{body}	Number of bodies	[–]
$\hat{\mathbf{n}}$	Panel normal unit vector	[–]
N_{seg}	VKI-BELL azimuthal discretisation index	[–]
n	Time step level	[–]
N_B	Number of body panels	[–]
N_b	Number of blades	[–]
N_c	Number of chordwise panels	[–]
N_r	Number of spanwise panels	[–]
N_t	Number of time steps	[–]
N_{w}	Number of wake panels	[–]
P	Shaft power	[W]
p	Static pressure	[Pa]

Symbol	Description	Unit
$\mathbf{P}$	Panel vertex	[m]
$\mathbf{P}^{\star}$	Permutation matrix for LU factorisation	[–]
p_{work}	Parallel workers	[–]
$\boldsymbol{Q}$	Torque vector	[N m]
q	Dynamic pressure	[Pa]
R	Rotor radius	[m]
r	Radius/radial coordinate	[m]
r_c	Vortex core radius	[m]
Re	Reynolds number	[–]
$\mathcal{R}$	Numerical residual	[–]
$\mathbf{r}$	Radial or distance vector	[m]
$\mathcal{S}$	Surface domain	[m^2]
S_{pw}	Parallel computing speed-up factor	[–]
S	Surface area	[m^2]
$\boldsymbol{T}$	Thrust vector	[N]
t	Time	[s]
$\hat{\boldsymbol{t}}_l$,$\hat{\boldsymbol{t}}_m$	Metric-aware tangential unit vectors	[–]
$\tilde{\mathbf{T}}$	Local panel frame transformation matrix	[–]
u	Chordwise perturbation velocity	[m s^{-1}]
$\mathbf{U}^{\star}$	Upper triangular matrix	[–]
$\mathbf{V}$	Velocity vector	[m s^{-1}]
$\mathbf{V}_{\text{conv}}$	Convective velocity for the VKI-BELL prescribed wake	[m s^{-1}]
v	Spanwise perturbation velocity	[m s^{-1}]
$\mathbf{v}_{\text{surf}}$	Surface perturbation velocity vector	[m s^{-1}]
$\mathcal{V}$	Volumetric domain	[m^3]
$\mathcal{W}$	Wake state assembly	[–]
w	Normal perturbation velocity	[m s^{-1}]
$\mathbf{x}_{\text{CP}}$	Control/collocation point vector	[m]
$\mathbf{x}$	Position vector	[m]

Greek alphabet

α	Angle of attack	[deg]
α_{Oseen}	Oseen constant	[–]
Γ	Circulation	[m^2 s^{-1}]
γ	Circulation density	[m s^{-1}]
δ	Diffusion parameter / Dirac delta	[–]
ε	Convergence threshold/vortex strain	[–]
ζ	Wake age	[rad]

Symbol	**Description**	**Unit**
$\boldsymbol{\zeta}$	Vorticity vector	$[\mathrm{s}^{-1}]$
η	Propulsive efficiency/normalised coordinate	[–]
Θ	Angular contribution to velocity potential	[–]
θ	Tangential coordinate	[deg]
θ_c	Collective pitch angle	[deg]
θ_{tw}	Twist angle	[deg]
Λ	Quarter-chord sweep angle	[deg]
λ	Taper ratio	[–]
μ	Doublet strength	$[\mathrm{m}^2\,\mathrm{s}^{-1}]$
$\boldsymbol{\mu}$	Doublet vector	$[\mathrm{m}^2\,\mathrm{s}^{-1}]$
ν	Kinematic viscosity	$[\mathrm{m}^2\,\mathrm{s}^{-1}]$
ξ	Non-dimensional abscissa	[–]
ρ	Density	$[\mathrm{kg}\,\mathrm{m}^{-3}]$
σ	Source strength	$[\mathrm{m}\,\mathrm{s}^{-1}]$
$\boldsymbol{\sigma}$	Source vector	$[\mathrm{m}\,\mathrm{s}^{-1}]$
σ'	Local blade solidity	[–]
σ_{std}	Standard deviation	[–]
Φ	Total velocity potential	$[\mathrm{m}^2\,\mathrm{s}^{-1}]$
ϕ	Perturbation velocity potential	$[\mathrm{m}^2\,\mathrm{s}^{-1}]$
φ	Inflow angle	[deg]
ψ	Azimuth	[deg]
Ω	Angular speed	$[\mathrm{rad}\,\mathrm{s}^{-1}]$
ω	Relaxation factor	[–]

Other symbols

$\nabla(\cdot)$	Gradient operator	[–]
$\nabla^2(\cdot)$	Laplace operator	[–]
$\Delta(\cdot)$	Difference between states	[–]
$(\cdot)\cdot(\cdot)$	Scalar product	[–]
$(\cdot)\times(\cdot)$	Vector product	[–]
$\partial(\cdot)$	Domain boundary	[–]
$d(\cdot)$	Differential quantity	[–]
$\lvert\cdot\rvert$	Vector norm	[–]
$(\cdot)'$	Spanwise quantity	[–]
$(\cdot)_{\mathrm{ax}}$	Axial quantity	[–]
$(\cdot)_{\mathrm{B}}$	Body quantity	[–]
$(\cdot)_{\mathrm{ind}}$	Induced quantity	[–]
$(\cdot)_{\mathrm{rad}}$	Radial quantity	[–]
$(\cdot)_{\mathrm{ref}}$	Reference quantity	[–]

Symbol	Description	Unit
$(\cdot)_{\mathrm{rel}}$	Relative quantity	[–]
$(\cdot)_{\mathrm{tan}}$	Tangential quantity	[–]
$(\cdot)_{\mathrm{scale}}$	Scaling quantity	[–]
$\tilde{(\cdot)}$	Target quantity	[–]
$(\cdot)_{\mathrm{tip}}$	Blade tip quantity	[–]
$(\cdot)_{\mathrm{tot}}$	Total quantity	[–]
$(\cdot)_{\mathrm{w}}$	Wake quantity	[–]
$(\cdot)_{\infty}$	Undisturbed quantity	[–]

Acronyms

AIC Aerodynamic Influence Coefficient

BELL Blade Element Lifting Line

BEMT Blade Element Momentum Theory

CFD Computational Fluid Dynamics

CP Control / collocation point

LE Leading Edge

NACA National Advisory Committee for Aeronautics

NASA National Aeronautics and Space Administration

NREL National Renewable Energy Laboratory

RANS Reynolds-Averaged Navier–Stokes

RHS Right-Hand Side

SDPM Source and Doublet Panel Method

TE Trailing Edge

UAE Unsteady Aerodynamic Experiment

VKI von Karman Institute

Chapter 1

Introduction

1.1 Motivation and Methodological Positioning

Aerodynamic analysis during preliminary design often requires repeated evaluations over many geometries, operating conditions, and modelling choices. High-fidelity computational fluid dynamics (CFD) can provide detailed flow predictions, including viscous, turbulent, separated, and compressible-flow effects, but its computational cost and meshing requirements may become prohibitive in iterative studies. Low- and mid-fidelity aerodynamic methods therefore remain important engineering tools: their purpose is not to reproduce the complete flow physics, but to retain the dominant aerodynamic mechanisms at a cost compatible with rapid analysis, verification, and design exploration [1], [2].

The present thesis is positioned within the low- and mid-fidelity modelling range. The development of in-house solvers was motivated not only by the need for aerodynamic predictions, but also by the need for direct access to the numerical formulation. For the purposes of this work, it was important to control the geometry generation, wake representation, boundary-condition treatment, load evaluation, and validation procedure within a unified framework. This level of control is difficult to obtain when using closed or highly specialised tools, and it is essential when the objective is not only to compute aerodynamic loads, but also to understand, verify, modify, and progressively extend the modelling approach.

Classical blade-element momentum theory (BEMT) represents one of the most efficient approaches for rotor analysis. By combining sectional aerodynamic data with momentum-theory induction estimates, it provides rapid predictions of quantities such as thrust, torque, and efficiency. Its computational efficiency, however, follows directly from simplifying assumptions concerning the induced flow and wake representation. These assumptions make BEMT particularly suitable for early-stage design applications, but they restrict the level of aerodynamic detail available from the model and may reduce its accuracy outside the range for which the underlying assumptions remain valid [1], [2], [3].

Lifting-line and vortex-based methods provide a first step beyond this level of modelling. They retain a compact geometrical representation of the lifting surfaces, but describe lift and induced velocities through bound and trailing vorticity. This makes it possible to represent wake effects more explicitly while preserving a relatively low computational cost. The von Karman Institute Blade-Element Lifting-Line solver (VKI-BELL) was developed within this modelling class. It is a rotor-oriented solver for steady operating conditions, combining blade-element sectional loads with a lifting-line representation of induced velocity and a semi-prescribed wake model. In this sense, VKI-BELL is not a BEMT code, although it retains the use of sectional aerodynamic closure based on airfoil data [4]. Within the thesis, VKI-BELL provides an intermediate reference level between VKI-BEMT and the more general formulation implemented in VKI-SDPM.

VKI-SDPM was developed as a surface-based potential-flow solver intended for wings and rotors. Source-and-doublet panel methods discretise the body surface with flow singularities such as source and doublet distributions, and they can account for thickness, provide pressure distributions, and treat lifting and non-lifting bodies within a unified framework. Unlike VKI-BELL, VKI-SDPM directly computes pressure-induced aerodynamic loads without relying on sectional aerodynamic data. Compared with volumetric CFD, source-and-doublet panel methods require only the discretization of the body surface and the wake, reducing both meshing complexity and computational cost. Their limitations are physical rather than geometrical: in their basic form they are inviscid and incompressible methods, and therefore cannot directly predict viscous drag, transition, separation, or compressibility effects [5].

The central motivation for VKI-SDPM is the development of a mid-fidelity solver capable of predicting the aerodynamic behaviour of both wings and rotors while remaining independent of external aerodynamic databases and any empirical model [5]. In addition, the level of detail provided by the solution is substantially higher than that obtainable with VKI-BEMT and VKI-BELL. Access to surface pressure distributions, rather than only sectional or integral performance quantities, provides a more detailed description of the aerodynamic loading and allows more direct comparisons with experimental or CFD data. VKI-SDPM supports both prescribed- and free-wake formulations and can be used for unsteady analyses within an inviscid, incompressible potential-flow framework.

The two solvers developed in this work are not competing alternatives, but tools placed at different levels of the aerodynamic-modelling hierarchy. VKI-BELL favours efficiency and rotor-oriented analysis. VKI-SDPM favours generality, surface-based load prediction, and future extensibility. Together, they define the two modelling levels investigated in this thesis.

1.2 Scope and Originality of the Work

The thesis does not introduce a new aerodynamic theory. Its theoretical basis is built on established concepts in the literature such as blade-element theory, lifting-line theory, incompressible potential flows, and source-and-doublet panel methods [4], [5], [6]. The originality of the work lies in the construction of two complete in-house solvers in which these concepts are implemented, coupled, verified, and validated within controllable numerical frameworks.

For VKI-BELL, the existing VKI-BEMT code provided a practical reference for the sectional aerodynamic treatment [2], [7], [8]. The new contribution is the development of a rotor solver architecture by integrating blade-element sectional loads, lifting-line-induced velocities, semi-prescribed wake geometry, and a vortex-core regularisation model. The architecture is designed to retain the efficiency of sectional modelling while introducing a more explicit description of wake-induced effects.

VKI-SDPM represents the main methodological development of the thesis. It was developed as a source-and-doublet panel solver for incompressible potential flow. Its contribution lies in the integration of the numerical elements required for a complete surface-based solver: influence-coefficient evaluation, pressure computation, load integration, prescribed- and free-wake evolution, vortex-core modelling, and unsteady time integration.

The current implementations are limited to aerodynamic modelling within the assumptions stated above. Multibody aerodynamic interactions, compressibility effects, prescribed relative body motions, and viscous–inviscid coupling are not included in the present solvers. They are discussed only as possible future extensions, especially for VKI-SDPM. Aeroelastic and aeroacoustic coupling are also outside the scope of the thesis.

The validation strategy is consistent with the fidelity level of the solvers. The objective is not to achieve CFD-level accuracy, but to assess whether the implemented formulations capture the dominant aerodynamic trends within their expected ranges of validity. For VKI-BELL, the validation concerns steady rotor-performance quantities. For VKI-SDPM, the assessment also includes sectional and integral aerodynamic loads, pressure distributions, and analyses of wake geometry for rotor configurations. For both solvers, the reference data are obtained from experimental studies reported in the literature [9], [10], [11], [12].

1.3 Objectives and Contributions

The general objective of this thesis is to develop, verify, and validate two original aerodynamic solvers for the low-speed aerodynamic analysis of wings and rotors [5]. The main objectives are:

- to formulate and implement VKI-BELL as a steady rotor-oriented blade-element lifting-line solver;
- to formulate and implement VKI-SDPM as a source-and-doublet panel solver for unsteady applications;
- to assess the sensitivity, convergence behaviour, robustness, and computational cost of the two solvers;
- to validate the numerical results against experimental reference data from the literature;
- to identify the main limitations of the current implementations and the most relevant directions for future development.

The main VKI-BELL contributions are:

- a rotor geometry-generation procedure based on radial distributions of chord, pitch, and radius;
- a semi-prescribed wake model representative of the rotor operating condition;
- a coupling strategy between blade-element sectional aerodynamics and lifting-line-induced velocities;
- an assessment of numerical sensitivity, computational performance, and predictive capability of the solver for steady rotor performance.

The main VKI-SDPM contributions are:

- a surface-discretisation framework for wings and rotors;
- a source-and-doublet formulation for incompressible potential-flow analysis of lifting components;
- prescribed- and free-wake capabilities;
- vortex-core models and core-growth evolution strategies for wake regularisation;
- an assessment of numerical sensitivity, computational performance, and predictive capability against wing and rotor reference data under different operating conditions.

1.4 Thesis Outline

The work is organised as follows.

Chapter 2 introduces the theoretical background required for the development of the two solvers. It presents the potential-flow framework and the boundary integral solution, blade-element concepts, lifting-line theory, and fundamentals of vortex theory.

Chapter 3 presents VKI-BELL by integrating the relevant theoretical background with the corresponding numerical implementation. The chapter discusses the rotor geometry, sectional aerodynamic closure, wake representation, vortex-core regularization, and numerical solution strategy as they are implemented in the solver.

Chapter 4 assesses the numerical behaviour and computational performance of VKI-BELL. A sensitivity analysis is conducted before the validation against the Caradonna–Tung test case [12]. VKI-BEMT results are also presented for a code-to-code comparison.

Chapter 5 presents the numerical implementation of the source-and-doublet formulation in VKI-SDPM, including surface discretisation, boundary-condition enforcement, wake modelling, pressure and load evaluation, and the time-marching procedure.

Chapter 6 assesses the numerical behaviour and computational performance of VKI-SDPM. The influence of discretization parameters, wake modelling choices, and regularisation parameters is investigated. Finally, an analysis on the parallel-computing performance is reported.

Chapter 7 validates VKI-SDPM on wing and rotor configurations. The selected test cases from the literature are the Yip–Shubert straight wing test case [9], the Kolbe–Boltz tapered and swept wing test case [10], the NREL Phase VI wind-turbine rotor test case [11], and the Caradonna–Tung rotor test case [12]. Each of these experimental benchmarks assesses a different characteristic of VKI-SDPM to provide a comprehensive view of the capabilities and limitations of the solver.

Chapter 8 provides an overview of the two solvers developed in this work and quantitatively compares them. The final conclusions are then drawn, and the possible future developments for both VKI-BELL and VKI-SDPM are discussed.

At the end of this manuscript, the appendices provide supplementary theoretical derivations, the classical BEMT formulation, implementation references for the solvers, and additional validation results for VKI-SDPM.

Chapter 2

Theoretical Background

2.1 Potential-Flow Formulation

The aerodynamic formulations developed in this thesis are based on a set of simplifying assumptions that are commonly adopted in aerodynamic modelling. These assumptions allow the governing fluid-dynamic problem to be reduced from the full Navier–Stokes equations to simpler formulations. The purpose of this section is to introduce the physical hypotheses underlying the solvers developed in this work. The focus is on the potential-flow problem, on the theoretical basis for the source-and-doublet panel formulation, and on the vortex methods.

2.1.1 Inviscid, Incompressible, and Irrotational Assumptions

A key nondimensional parameter in fluid-dynamic theory is the Reynolds number, which compares inertial and viscous effects in the flow. For a characteristic velocity V_{ref}, characteristic length l_{ref}, and kinematic viscosity ν, it is defined as

$$Re = \frac{V_{\mathrm{ref}}\ l_{\mathrm{ref}}}{\nu}. \tag{2.1}$$

For many aerodynamic applications involving wings and rotors, the Reynolds number is sufficiently high for inertial effects to dominate most of the flow domain, while viscous effects are mainly confined to thin boundary-layer regions close to solid boundaries and to wakes. Outside these viscous regions, the flow may be approximated as inviscid, and the outer solution can be used to determine the pressure distribution and the pressure-induced aerodynamic loads. In the present formulation, the outer aerodynamic field is assumed to be inviscid, while viscous effects are either introduced indirectly through modelling inputs or left outside the resolved flow problem.

This separation between inviscid and viscous regions represents one of the classical foundations of fluid-dynamic modelling. In the inviscid outer region, the governing equations reduce from the full Navier–Stokes equations to the Euler equations. The thin viscous region close to the wall is governed by the boundary-layer equations. These equations retain the dominant viscous effects in the wall-normal direction and are driven by the pressure distribution imposed by the external inviscid flow [5].

The relevance of compressibility effects is estimated through the Mach number, defined as the ratio between a characteristic flow velocity V_{ref} and the speed of sound a,

$$M = \frac{V_{\text{ref}}}{a}, \tag{2.2}$$

When the characteristic Mach number is sufficiently low, pressure-induced density variations are negligible and the density of each fluid element can be treated as constant. Under this assumption, the continuity equation can be written in its incompressible formulation as

$$\nabla \cdot \mathbf{V} = 0, \tag{2.3}$$

where $\mathbf{V}$ is the velocity vector field [5].

In addition to being treated as incompressible and inviscid in the outer region, the flow is assumed to be irrotational outside boundary layers and wakes. This assumption can be expressed by the curl of the velocity,

$$\nabla \times \mathbf{V} = \mathbf{0}. \tag{2.4}$$

Irrotational fields can be written as the gradient of a scalar potential function, in this case indicated by Φ,

$$\mathbf{V} = \nabla\Phi. \tag{2.5}$$

Substituting Eq. (2.5) into Eq. (2.3) allows the aerodynamic problem to be formulated through Laplace's equation,

$$\nabla^2\Phi = 0. \tag{2.6}$$

The total potential Φ can be decomposed into an undisturbed component Φ_∞, which is related to the undisturbed potential flow, and a perturbation component ϕ related to the velocity field induced by the presence of a body within the domain,

$$\Phi = \Phi_\infty + \phi, \tag{2.7}$$

For the undisturbed freestream, the velocity field is

$$\nabla\Phi_\infty = \mathbf{V}_\infty. \tag{2.8}$$

Since $\mathbf{V}_\infty$ is uniform, its divergence is zero. Therefore, using Eq. (2.7) in Eq. (2.6), the governing equation for the perturbation potential becomes

$$\nabla^2\Phi = \nabla^2\Phi_\infty + \nabla^2\phi = 0 \quad \rightarrow \quad \nabla^2\phi = 0. \tag{2.9}$$

The Laplace equation for the aerodynamic problem can therefore be formulated in terms of the perturbation potential.

2.1.2 Boundary and Far-Field Conditions

Once the potential problem has been formulated, a set of boundary conditions is required to determine the solution. These conditions connect the mathematical formulation of the perturbation field to the physical constraints.

Let $\mathcal{V}$ denote the fluid domain. For a lifting configuration, the boundary of this domain includes the body surface $\mathcal{S}_\mathrm{B}$, the wake surface $\mathcal{S}_\mathrm{w}$, and a far-field boundary $\mathcal{S}_\infty$ placed formally at infinity. The body surface is a solid boundary, while the wake surface is not.

The perturbation potential is required to vanish sufficiently far from the body, so that

$$\nabla\phi \rightarrow \mathbf{0} \qquad \text{as} \qquad |\mathbf{x}| \rightarrow \infty. \tag{2.10}$$

This condition expresses the fact that the disturbance induced by the body and the wake must decay at large distances, leaving only the undisturbed flow.

The second condition is imposed on the body surface and is known as the impermeability condition. On a solid boundary, this condition requires that the normal component of the relative velocity between the fluid and the body vanishes as the surface is approached. For a body moving with local velocity $\mathbf{V}_\mathrm{B}$, this condition can be written in terms of the total potential as

$$\mathbf{n} \cdot (\nabla\Phi - \mathbf{V}_\mathrm{B}) = 0 \qquad \text{on } \mathcal{S}_\mathrm{B}, \tag{2.11}$$

where $\mathbf{n}$ is the outward unit normal to the body surface. Using the decomposition introduced in Eq. (2.7), the same condition can be written explicitly for the perturbation potential as

$$\mathbf{n} \cdot \nabla\phi = \mathbf{n} \cdot (\mathbf{V}_\mathrm{B} - \nabla\Phi_\infty) \qquad \text{on } \mathcal{S}_\mathrm{B}. \tag{2.12}$$

For a fixed body in the chosen frame of reference $\mathbf{V}_\mathrm{B} = \mathbf{0}$ and Eq. (2.12) reduces to

$$\mathbf{n} \cdot \nabla\phi = -\mathbf{n} \cdot \nabla\Phi_\infty \qquad \text{on } \mathcal{S}_\mathrm{B}. \tag{2.13}$$

This condition does not impose zero tangential velocity at the wall. It only prevents the fluid from crossing the solid boundary, while allowing tangential slip, consistently with the inviscid-flow assumption. In unsteady formulations or moving-body problems, the body velocity $\mathbf{V}_\mathrm{B}$ must be retained in the boundary condition.

2.1.3 Superposition Principle and Singularities

The central mathematical property of Eq. (2.6) is its linearity. In the same fluid domain $\mathcal{V}$, let $\phi_1, \phi_2, \ldots, \phi_N$ be solutions of Laplace's equation written for the perturbation potential, so that

$$\nabla^2\phi_k = 0, \qquad k = 1, \ldots, N. \tag{2.14}$$

Then any linear combination of these solutions can be written as

$$\phi = \sum_{k=1}^{N} c_k \phi_k, \tag{2.15}$$

where c_k are arbitrary constants. Equation (2.15) is also a solution of Laplace's equation. Indeed,

$$\nabla^2\phi = \nabla^2 \left(\sum_{k=1}^{N} c_k \phi_k \right) = \sum_{k=1}^{N} c_k \nabla^2 \phi_k = 0. \tag{2.16}$$

This result is known as the superposition principle and it allows complex fields to be constructed as linear combinations of elementary solutions of Laplace's equation. In potential-flow theory, these elementary solutions are commonly introduced as singularity solutions. They satisfy Laplace's equation everywhere except at their singular location and are used as mathematical building blocks to satisfy prescribed boundary conditions [4], [5].

Although the final aerodynamic problem is generally three-dimensional, the physical meaning of the basic singularities is often most clearly understood in two dimensions. The basic singularities considered here are the source, doublet, and potential vortex.

Two-Dimensional Point Source

The source singularity is associated with a pure radial flux of fluid. Therefore, in a polar coordinate system (r, θ) centred at the source location, the velocity field of a two-dimensional source has components

$$V_{\text{rad}}(r,\theta) = \frac{\sigma}{2\pi r}, \qquad V_{\tan}(r,\theta) = 0, \tag{2.17}$$

where σ is the source strength. The corresponding velocity potential is obtained by integrating $V_{\text{rad}} = \partial\phi/\partial r$, giving

$$\phi = \frac{\sigma}{2\pi}\ln r. \tag{2.18}$$

The strength σ represents the flux induced by the source. Indeed, the flux across a circle of radius R centred at the singularity is

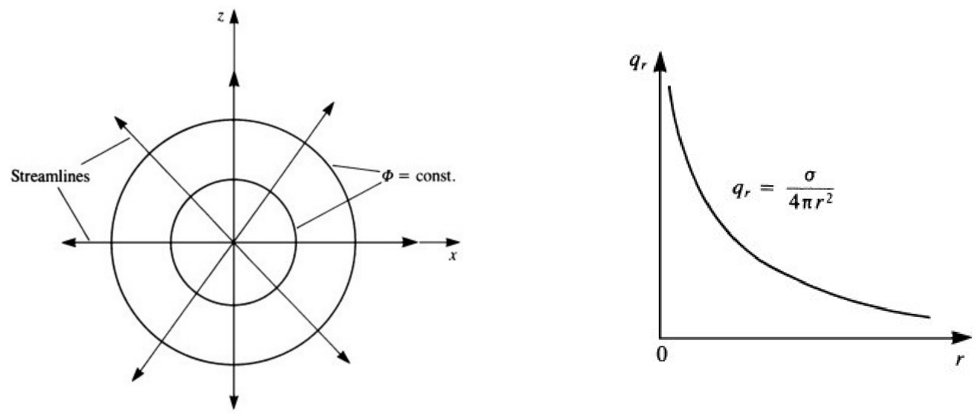
$$\int_0^{2\pi} V_{\text{rad}}(R,\theta)R\,d\theta = \int_0^{2\pi}\frac{\sigma}{2\pi R}R\,d\theta = \sigma. \tag{2.19}$$

Thus, a positive source produces a radial velocity directed away from the singularity, while a negative source represents a sink. The velocity magnitude decays as $1/r$, and the singular point $r = 0$, where the velocity becomes infinite, must be excluded from the regular flow domain.

For a source located at (x_0, z_0), the potential and velocity components in Cartesian coordinates can be written as

$$\phi(x,z) = \frac{\sigma}{2\pi}\ln\left(\sqrt{(x-x_0)^2+(z-z_0)^2}\right), \tag{2.20}$$

$$u = \frac{\partial\phi}{\partial x} = \frac{\sigma}{2\pi}\frac{x-x_0}{(x-x_0)^2+(z-z_0)^2}, \quad w = \frac{\partial\phi}{\partial z} = \frac{\sigma}{2\pi}\frac{z-z_0}{(x-x_0)^2+(z-z_0)^2}. \tag{2.21}$$



(a) Streamlines and equipotential lines. **(b)** Source-induced velocity field.

Figure 2.1: Two-dimensional source singularity [5].

Two-Dimensional Point Doublet

A doublet can be obtained as the limiting combination of a source and a sink of equal strength brought infinitesimally close to each other, such that their strength multiplied by their separation distance becomes the constant μ [5]. Unlike the source, its effect is directional,

$$\phi(r,\theta) = \frac{-\boldsymbol{\mu}\cdot\mathbf{r}}{2\pi\ r^2} \tag{2.22}$$

For example, if the doublet is aligned with the x-axis, such that $\boldsymbol{\mu} = (\mu,0)$, the velocity potential for a two-dimensional doublet located at the origin becomes

$$\phi(r,\theta) = -\frac{\mu}{2\pi}\frac{\cos\theta}{r}. \tag{2.23}$$

The corresponding polar velocity components are

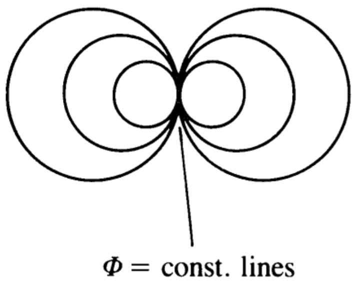
$$V_{\mathrm{rad}}(r,\theta) = \frac{\partial\phi}{\partial r} = \frac{\mu\cos\theta}{2\pi r^2}, \qquad V_{\mathrm{tan}}(r,\theta) = \frac{1}{r}\frac{\partial\phi}{\partial\theta} = \frac{\mu\sin\theta}{2\pi r^2}. \tag{2.24}$$

In Cartesian coordinates, for a doublet located at (x_0, z_0) and oriented in the x-direction, the potential and the corresponding velocity components become

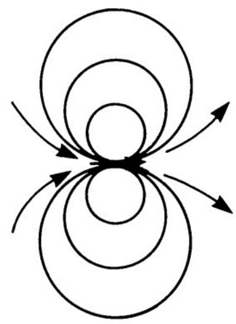
$$\phi(x,z) = \frac{-\mu}{2\pi}\frac{x-x_0}{(x-x_0)^2+(z-z_0)^2}, \tag{2.25}$$

$$u(x,z) = \frac{\partial\phi}{\partial x} = \frac{\mu}{2\pi}\frac{(x-x_0)^2-(z-z_0)^2}{[(x-x_0)^2+(z-z_0)^2]^2}, \tag{2.26}$$

$$w(x,z) = \frac{\partial\phi}{\partial z} = \frac{\mu}{2\pi}\frac{2(x-x_0)(z-z_0)}{[(x-x_0)^2+(z-z_0)^2]^2}. \tag{2.27}$$



(a) Equipotential lines.



(b) Streamlines.

Figure 2.2: Two-dimensional doublet singularity aligned with the x-axis [5].

Two-Dimensional Potential Vortex

The concept of circulation is central to the definition of a potential vortex. The circulation around a closed contour $\mathcal{C}$, with $d\mathbf{l}$ denoting the tangential differential element, is defined as

$$\Gamma = \oint_{\mathcal{C}} \mathbf{V} \cdot d\mathbf{l}, \tag{2.28}$$

Circulation is the net tangential velocity integrated along a closed path. In an irrotational flow, it vanishes unless the contour encloses a singularity. A potential vortex is irrotational everywhere except at its centre, yet produces finite circulation around any enclosing contour and induces only a tangential velocity component. In polar coordinates centred at the vortex location, the velocity field and the corresponding velocity potential are

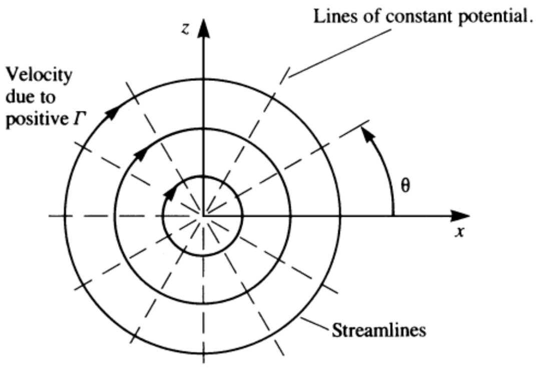
$$V_{\text{rad}}(r,\theta) = 0, \qquad V_{\tan}(r,\theta) = -\frac{\Gamma}{2\pi r}, \qquad \phi(r,\theta) = -\frac{\Gamma}{2\pi}\theta. \tag{2.29}$$

where Γ is the circulation strength, with the sign depending on the adopted orientation convention. The corresponding form in Cartesian coordinates for a vortex at (x_0, z_0) may be written as

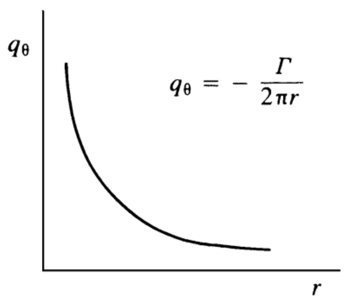
$$\phi(x,z) = -\frac{\Gamma}{2\pi}\tan^{-1}\left(\frac{z-z_0}{x-x_0}\right), \tag{2.30}$$

$$u(x,z) = \frac{\partial\phi}{\partial x} = \frac{\Gamma}{2\pi}\frac{z-z_0}{(x-x_0)^2+(z-z_0)^2}, \tag{2.31}$$

$$w(x,z) = \frac{\partial\phi}{\partial z} = -\frac{\Gamma}{2\pi}\frac{x-x_0}{(x-x_0)^2+(z-z_0)^2}. \tag{2.32}$$



(a) Streamlines and equipotential lines.



(b) Induced velocity.

Figure 2.3: Two-dimensional potential vortex [5].

2.1.4 Circulation, Wake, and Kutta Condition

Combining the definition of circulation in Eq. (2.28) with the velocity-potential relation in Eq. (2.5) gives

$$\Gamma = \oint_{\mathcal{C}} \nabla\Phi \cdot d\mathbf{l}. \tag{2.33}$$

If the potential Φ is single-valued and regular inside the contour, the line integral of its gradient over a closed curve is zero. A non-zero circulation can therefore arise only if the contour encloses a singularity or crosses a discontinuity of the potential. This is illustrated in Fig. 2.4 where the integration lines do not intersect the discontinuity and the net circulation is zero.

A doubly connected region is obtained when a cut is introduced in the domain, allowing the potential to take different values on the two sides of the cut. In lifting potential-flow formulations, this role is played by the wake surface $\mathcal{S}_\mathrm{w}$, which acts as a force-free surface across which the pressure is continuous but the potential is not. Therefore, when the contour $\mathcal{C}$ crosses the wake, a discontinuity in the velocity potential is present and Γ can then be related directly to this jump up to the sign convention for the normals,

$$\Gamma = [\Phi]_{\mathcal{S}_\mathrm{w}} = \Phi^+ - \Phi^-, \tag{2.34}$$

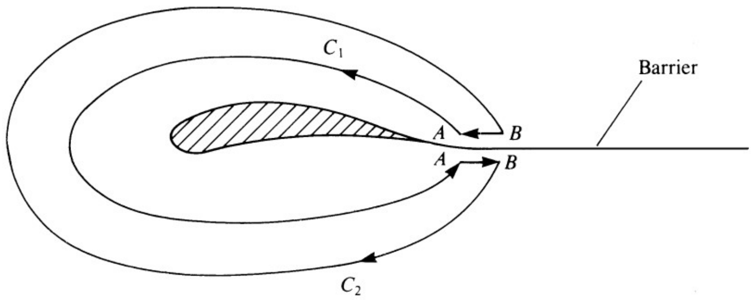


Figure 2.4: Simply connected domain around an airfoil profile [5].

Since the undisturbed potential Φ_∞ is single-valued, the same jump can be expressed in terms of the perturbation potential,

$$\Gamma = [\phi]_{\mathcal{S}_\mathrm{w}}. \tag{2.35}$$

The wake surface therefore provides the potential-flow formulation with the mathematical mechanism required to represent a lifting solution with non-zero circulation while retaining an irrotational outer flow.

The introduction of a potential jump raises the question of how its value should be determined. At this stage, the potential-flow problem still admits an infinite family of solutions satisfying both the impermeability and far-field conditions. Each solution is associated with a different circulation and, therefore, with a different velocity and pressure distribution around the body. Examples of solutions for different values of Γ are illustrated in Fig. 2.5; among these, only one is physically admissible.

To resolve this issue, an additional condition must be imposed. This condition is known as the Kutta condition and it was formulated based on physical observations of flows around aerodynamic bodies [13]. For a body with a sharp trailing edge at sufficiently high Reynolds number, the flow leaves the body at the trailing edge smoothly. This mechanism is viscous in nature. Since viscosity is not resolved by the inviscid potential-flow formulation, the Kutta condition must be imposed as an additional constraint. From a mathematical point of view, the Kutta condition defines the jump across the wake surface so that it selects the only physically admissible solution.

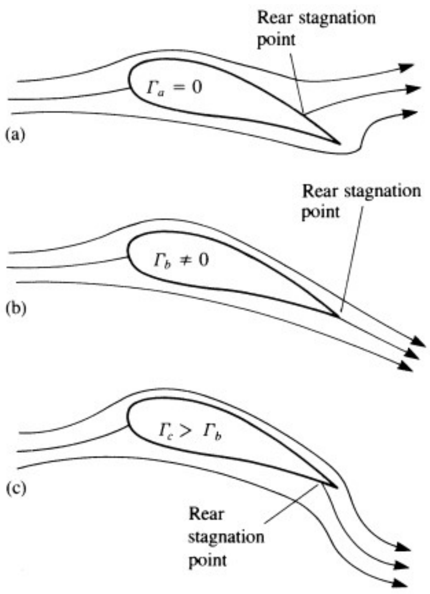


Figure 2.5: Different solutions to the potential-flow problem. The physical solution is case (b), for which the Kutta condition is imposed [5].

An important result connecting circulation and wake formation is Kelvin's theorem. For an inviscid and incompressible flow subject only to conservative body forces, the circulation around a closed material curve is conserved in time,

$$\frac{D\Gamma}{Dt} = 0. \tag{2.36}$$

The term material is essential: the contour considered in Kelvin's theorem is made of the same fluid particles as it moves with the flow. Consequently, circulation cannot be arbitrarily created or destroyed within such a contour. In lifting flows, the development or variation of bound circulation must therefore be balanced by circulation shed into the surrounding flow.

When an initially stationary wing starts moving, this balance appears through the shedding of a starting vortex of opposite sign. More generally, the wake carries downstream the circulation history of the lifting body. It is therefore not only a numerical device used to close the potential-flow formulation, but also the physical mechanism through which the circulation balance required by Kelvin's theorem is maintained in steady and unsteady formulations [5], [14]. The derivation of Kelvin's theorem is reported in App. A.1.1.

2.1.5 Limitations of the Potential-Flow Formulation

The limitations of the potential-flow formulation follow directly from the assumptions on which it is based. Since viscosity is not explicitly resolved, physical effects that depend on it cannot be directly predicted. These include viscous drag, boundary-layer growth, flow separation, turbulence, and several unsteady phenomena associated with separated flows, such as dynamic stall and vortex shedding. This limitation becomes relevant when more realistic flow descriptions are required, particularly around bluff bodies or configurations where separation plays a dominant role.

Compressibility effects are also outside the present formulation. Phenomena such as finite propagation speed of pressure disturbances, density variations due to local acceleration, shock waves, and drag divergence cannot be captured by an incompressible potential-flow model. In low-speed aerodynamics, incompressible formulations are generally considered appropriate when the Mach number is sufficiently small, commonly for $M \leq 0.3$. To extend the range of validity at moderately higher subsonic Mach numbers without introducing more complex formulations, the Prandtl–Glauert transformation may be used

$$C_{p,\mathrm{comp}} = \frac{C_{p,M=0}}{\sqrt{1-M^2}} \tag{2.37}$$

However, this correction loses accuracy towards the high-subsonic regime as the Mach number approaches the transonic range and the linearity assumption on which it is based becomes increasingly invalid.[4], [5], [15].

2.2 Vortex Theory and Modelling

The present section introduces the main vortex-based concepts required in the implementation of the solvers. The vorticity field $\boldsymbol{\zeta}$ is defined as the curl of the velocity field,

$$\boldsymbol{\zeta} = \nabla \times \mathbf{V}, \tag{2.38}$$

Recalling the definition of circulation in Eq. (2.28), by applying Stokes' theorem this can be related to the vorticity flux through any open surface $\mathcal{S}$ bounded by the contour $\mathcal{C}$,

$$\Gamma = \int_{\mathcal{S}} \boldsymbol{\zeta} \cdot \mathbf{n}\, dS, \tag{2.39}$$

where $\mathbf{n}$ is the outward unit normal to the surface. This relation shows that circulation is a measure of the vorticity content crossing the surface enclosed by the contour. It also provides the connection between a global quantity, the circulation around a curve, and a local quantity, the vorticity field.

2.2.1 Helmholtz Vortex Theorems

The behaviour of vortical structures in an inviscid, incompressible flow is described by the Helmholtz vortex theorems [5], [16]. These theorems establish the main kinematic properties of vortex tubes and provide the physical basis for treating wake vortices as connected and coherent structures.

Using the definitions introduced above, a vortex tube is formed by a bundle of vortex lines, and its strength is given by the circulation around a contour enclosing the tube, or equivalently by the flux of vorticity through any cross-section, as expressed in Eqs. (2.28) and (2.39). Representations of a vortex line and a vortex tube are reported in Fig. 2.6. In the limiting case in which the cross-sectional area tends to zero while the circulation remains finite, the vortex tube is idealised as a vortex filament.

For an inviscid incompressible flow, the Helmholtz vortex theorems can be stated as follows:

1. the strength of a vortex tube is constant along its length;
2. a vortex tube cannot start or end arbitrarily inside the fluid domain;
3. vortex lines move with the fluid.

The first result follows from the solenoidal character of the vorticity field. If two cross-sections of the same vortex tube are considered, the vorticity flux through them must be the same. Therefore,

$$\Gamma_1 = \Gamma_2, \tag{2.40}$$

where Γ_1 and Γ_2 are the vortex-tube strengths measured at the two sections. For a thin vortex tube with approximately uniform vorticity over each cross-section, this result can be written as

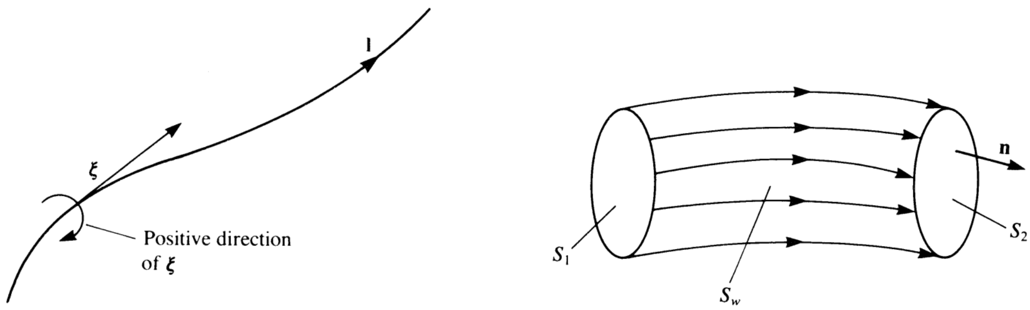
$$\zeta_1 A_1 = \zeta_2 A_2, \tag{2.41}$$

where A is the cross-sectional area. This relation shows that stretching a vortex tube reduces its cross-section and increases the vorticity magnitude, while expansion produces the opposite effect.

The second theorem implies that a vortex filament cannot be represented as an isolated segment ending inside the fluid. It must form a closed loop, intersect a boundary, connect with other vortex structures, or extend outside the domain considered. This has a direct consequence for aerodynamic wake modelling: trailing vortices and wake elements must be arranged as part of a connected vortical system.

Finally, since vortex lines move with the fluid, the position of a wake vortex evolves with the local flow motion. This result provides the theoretical basis for free-wake methods, in which wake nodes or vortex elements are convected according to the local velocity field.

A mathematical derivation of the Helmholtz vortex theorems is reported in Appendix A.1.



(a) Vortex line. **(b)** Vortex tube.

Figure 2.6: Vortex line and vortex tube [5].

2.2.2 Biot–Savart Law

The Biot–Savart law provides the relation between a vorticity distribution and the velocity field induced by it. For a vorticity field $\boldsymbol{\zeta}(\mathbf{x}_0)$ distributed over a fluid volume $\mathcal{V}$, the induced velocity at a field point $\mathbf{x}$ is

$$\mathbf{V}_{\mathrm{ind}}(\mathbf{x}) = \frac{1}{4\pi}\int_{\mathcal{V}} \frac{\boldsymbol{\zeta}(\mathbf{x}_0)\times(\mathbf{x}-\mathbf{x}_0)}{|\mathbf{x}-\mathbf{x}_0|^3}\, d\mathcal{V}_0. \tag{2.42}$$

Here, $\mathbf{x}_0$ is the source point where vorticity is located, while $\mathbf{x}-\mathbf{x}_0$ is the vector from the source point to the field point. The cross product determines the direction of the induced velocity, and the denominator shows the decay of the induced contribution with distance.

In vortex-filament models, the vorticity is idealised as being concentrated along a curve $\mathcal{C}$. If the filament has circulation Γ, the volume distribution in Eq. (2.42) reduces to a line distribution, and the induced velocity becomes

$$\mathbf{V}_{\mathrm{ind}}(\mathbf{x}) = \frac{\Gamma}{4\pi}\int_{\mathcal{C}} \frac{d\mathbf{l}\times(\mathbf{x}-\mathbf{x}_0)}{|\mathbf{x}-\mathbf{x}_0|^3}. \tag{2.43}$$

In numerical vortex methods, continuous filaments are usually discretised into straight vortex segments. The total induced velocity is then obtained by superposition of the contributions of all the segments. Consider a straight segment of circulation Γ, oriented from endpoint $\mathbf{x}_1$ to endpoint $\mathbf{x}_2$, and a field point $\mathbf{x}$. Defining

$$\mathbf{r}_1 = \mathbf{x}-\mathbf{x}_1, \qquad \mathbf{r}_2 = \mathbf{x}-\mathbf{x}_2, \qquad \mathbf{r}_0 = \mathbf{x}_2-\mathbf{x}_1, \tag{2.44}$$

the induced velocity of the finite straight segment can be written as

$$\mathbf{V}_{\mathrm{ind}} = \frac{\Gamma}{4\pi}\frac{\mathbf{r}_1\times\mathbf{r}_2}{|\mathbf{r}_1\times\mathbf{r}_2|^2}\left[\mathbf{r}_0\cdot\left(\frac{\mathbf{r}_1}{|\mathbf{r}_1|}-\frac{\mathbf{r}_2}{|\mathbf{r}_2|}\right)\right]. \tag{2.45}$$

This expression is particularly useful in numerical models because it depends only on the segment endpoints, the field point, and the circulation strength [5].

The same result can also be expressed in scalar form. Using the perpendicular distance d from the field point to the vortex line and the endpoint angles β_1 and β_2 defined in Fig. 2.7, the tangential velocity induced around the vortex line is

$$V_{\mathrm{tan}} = \frac{\Gamma}{4\pi d}\left(\cos\beta_1 - \cos\beta_2\right). \tag{2.46}$$

This form makes the geometrical dependence of the finite-segment contribution explicit. In the limiting case of an infinitely long straight vortex filament, $\beta_1 \to 0°$ and $\beta_2 \to 180°$, and Eq. (2.46) reduces to

$$V_{\mathrm{tan}} = \frac{\Gamma}{2\pi d}. \tag{2.47}$$

The Biot–Savart law is therefore the basic tool used to compute the induced velocity of vortex filaments, vortex rings, and wake elements in vortex-based

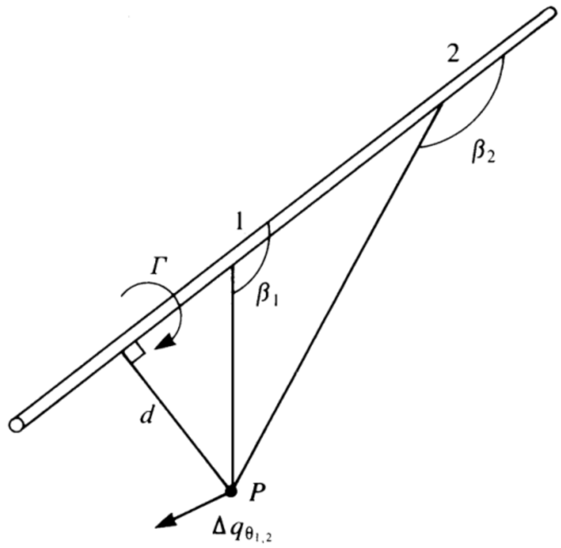


Figure 2.7: Velocity induced by a straight vortex filament [5].

aerodynamic methods. It also appears in source-and-doublet panel methods when constant-strength doublet panels are replaced, for induced-velocity evaluation, by their equivalent vortex-ring representation.

2.2.3 Vortex-Core Regularisation and Core Growth

The ideal Biot–Savart formulation reported in Eq. (2.43) becomes singular when the field point approaches the vortex filament. This singular behaviour is a consequence of representing the vorticity as concentrated along a line of zero thickness. Real vortices, however, possess a finite core in which vorticity is distributed over a non-zero area. As a result, the tangential velocity does not follow the potential-vortex law down to the vortex centre, but remains finite and usually reaches a maximum at a finite radial distance [5], [17].

For this reason, a core regularisation is introduced in which the Biot–Savart kernel is smoothed within a finite length scale. This length scale is commonly denoted by r_c and is referred to as the vortex-core radius. In numerical wake simulations, r_c plays a dual role: it represents, in a simplified way, the finite size of real vortex cores, and it prevents excessively large induced velocities when vortex elements pass close to one another or close to the lifting body [18], [19].

For an infinite rectilinear vortex, a generic regularised tangential velocity can be written as

$$V_{\tan} = \frac{\Gamma}{2\pi d} F\left(\frac{d}{r_c}\right), \tag{2.48}$$

where d is the distance from the vortex centreline and $F(d/r_c)$ is a damping function. The function must recover the potential-vortex behaviour far from the core,

$$F\left(\frac{d}{r_c}\right) \to 1 \qquad \text{for} \qquad d \gg r_c. \tag{2.49}$$

Several damping functions have been proposed in the literature. The Rankine model assumes solid-body rotation inside the core and potential-vortex behaviour outside it,

$$F(\bar{d}) = \begin{cases} \bar{d}^2, & \bar{d} \leq 1, \\ 1, & \bar{d} > 1, \end{cases} \qquad \bar{d} = \frac{d}{r_c}. \tag{2.50}$$

Although simple, this model introduces a discontinuity in the derivative of the tangential velocity at the core radius. Smoother models are therefore often preferred in wake calculations.

The Lamb–Oseen model is associated with the viscous diffusion of an initially concentrated vortex. If r_c is defined as the radial position of maximum tangential velocity, the tangential velocity can be written as

$$V_{\tan} = \frac{\Gamma}{2\pi d}\left[1 - \exp\left(-\alpha_{\text{Oseen}}\frac{d^2}{r_c^2}\right)\right], \qquad \alpha_{\text{Oseen}} = 1.25643. \tag{2.51}$$

The constant α_{Oseen}, often referred to as the Oseen constant, depends on the convention used to define the core radius. If r_c is instead interpreted as a diffusion length scale, the same model may appear with a different normalisation. This distinction is important because the expression "Lamb–Oseen core" is not always used with the same definition of r_c in the rotor-wake literature [18], [19], [20], [21].

A commonly used algebraic regularisation is the Scully-type core model,

$$F(\bar{d}) = \frac{\bar{d}^2}{1 + \bar{d}^2}, \tag{2.52}$$

which gives

$$V_{\tan} = \frac{\Gamma}{2\pi}\frac{d}{d^2 + r_c^2}. \tag{2.53}$$

This form is convenient because it can be introduced directly into the Biot–Savart expression by replacing the singular distance dependence with a regularised one [22], [23], [24].

A more general algebraic family is provided by the Vatistas model,

$$F(\bar{d}) = \frac{\bar{d}^2}{\left(1 + \bar{d}^{2n}\right)^{1/n}}, \tag{2.54}$$

The parameter n controls the sharpness of the transition between the inner-core region and the external potential-vortex behaviour. For $n = 1$, the Vatistas model reduces to the Scully-type formulation, while larger values of n produce a sharper transition [19], [25].

In addition to the choice of the damping function, the core radius may be allowed to evolve with wake age. This process is usually referred to as vortex-core growth. A common diffusion-based model assumes that the square of the core radius grows linearly in time,

$$r_c^2(t) = r_{c,0}^2 + 4\alpha_{\text{Oseen}}\delta\nu t, \tag{2.55}$$

where $r_{c,0}$ is the initial core radius, ν is the molecular kinematic viscosity, and δ is the turbulent diffusion parameter. If the wake age is expressed as an azimuthal age $\zeta = \Omega t$, Eq. (2.55) becomes

$$r_c^2(\zeta) = r_{c,0}^2 + \frac{4\alpha_{\text{Oseen}}\delta\nu}{\Omega}\zeta. \tag{2.56}$$

The initial core radius $r_{c,0}$ defines the core size when a vortex element is generated and is often prescribed as a fraction of a blade chord, a local panel dimension, or another reference length. Rotor near-wake studies, such as those by Young, provide useful guidance for estimating the initial or reference core size, but they do not remove the dependence on the specific rotor geometry, loading condition, wake age, and measurement or discretisation scale [26].

The parameter δ is particularly important in rotor-wake simulations. For a purely laminar Lamb–Oseen diffusion model, $\delta = 1$. However, rotor tip vortices often exhibit a faster apparent core growth than that predicted by molecular viscosity alone. This additional spreading may be associated with unresolved turbulent mixing, vortex instabilities, interaction with neighbouring wake elements, and numerical diffusion introduced by the discretisation. Therefore, δ should not be interpreted as a universal physical constant, but as an effective parameter representing unresolved diffusion mechanisms within the adopted vortex model [19], [27], [28].

A frequently used empirical approach is based on Squire's hypothesis, in which the effective diffusion depends on the vortex Reynolds number,

$$Re_v = \frac{\Gamma}{\nu}, \tag{2.57}$$

through a relation of the form

$$\delta = 1 + a_1 Re_v, \tag{2.58}$$

where a_1 is an empirical coefficient. Gupta reports different values of this coefficient for different applications,

$$a_1 = \begin{cases} 2 \cdot 10^{-4}, & \text{for rotors,} \\ 5 \cdot 10^{-5}, & \text{for wings.} \end{cases} \tag{2.59}$$

These values illustrate the empirical nature of the model: the effective diffusion depends on the flow configuration, Reynolds number, vortex type, and calibration database [19]. This is especially relevant for hover simulations, where the wake remains close to the rotor for several revolutions and the induced-velocity field can be highly sensitive to the assumed core growth. In this context, δ becomes an actual modelling variable: increasing it tends to spread the core faster and can improve numerical robustness, but excessive diffusion may also smooth out physically relevant induced-velocity gradients.

Core growth can also be affected by wake deformation. As vortex filaments are convected, they may stretch or contract under the induced velocity field. If a vortex segment changes from an initial length l_0 to a length l, a simple incompressible vortex-tube argument gives

$$r_c^2 l = r_{c,0}^2 l_0, \quad \rightarrow \quad r_c = r_{c,0}\sqrt{\frac{l_0}{l}}. \tag{2.60}$$

This relation shows that stretching tends to reduce the core radius, while contraction tends to increase it. Therefore, viscous diffusion and filament stretching may act as competing mechanisms: diffusion spreads the core, whereas stretching can locally contract it [28], [29].

A simple combined model can be written as

$$r_c(\zeta) = \left(r_{c,0}^2 + \frac{4\alpha_{\text{Oseen}}\delta\nu}{\Omega}\zeta\right)^{1/2}\left(\frac{l_0}{l(\zeta)}\right)^{1/2}. \tag{2.61}$$

This expression should be interpreted as a modelling closure rather than as an exact solution of the vorticity transport equation. More detailed approaches may apply the strain correction incrementally or introduce a dependence on the full wake-age history of the vortex element [18], [28], [29].

The choice of vortex-core model and core-growth parameters affects both the physical behaviour and the numerical stability of wake simulations. A small core radius approaches the ideal singular vortex solution, but may generate very large induced velocities and numerical instabilities. A large core radius gives a smoother and more robust computation, but may excessively damp the induced-velocity field and alter the predicted wake roll-up. Similarly, increasing δ may stabilise hover simulations by accelerating core spreading, but it can also reduce physically meaningful velocity gradients if not properly calibrated.

At present, no universal analytical formula exists for selecting $r_{c,0}$, δ, the damping function, or the stretching correction for all rotor geometries, operating conditions, discretisations, and wake ages. Existing models are empirical or semi-empirical and must be selected consistently with the physical problem and the numerical method. In practice, the vortex-core parameters are therefore chosen through modelling assumptions and sensitivity analyses, seeking a compromise between physical fidelity and numerical stability [18], [19], [26].

2.3 Lifting-Line Modelling and Blade-Element Closure

The vortex-based concepts introduced in the previous sections provide the physical basis for representing lifting flows through bound circulation and trailing vorticity. For rotor and propeller applications, this idea is particularly useful because the blade can be replaced by a radial distribution of bound circulation, while the wake is represented through the vortex system shed from the blade. However, a vortex model alone does not determine the circulation distribution. A closure relation is required to connect the local circulation to the aerodynamic forces generated by the blade sections.

The present section introduces the lifting-line representation adopted as the basis of the VKI-BELL solver and then describes its sectional closure through the Kutta–Joukowski theorem and blade-element aerodynamics. In this way, the induced velocity is provided by the lifting-line wake model, whereas the sectional loads are obtained from airfoil polar data.

2.3.1 Lifting-Line Representation

Prandtl's lifting-line theory was originally developed to describe finite wings of sufficiently large aspect ratio [5], [30], [31]. Its main idea is to replace the lifting surface by a line carrying a bound circulation distribution. For a finite wing, this circulation is usually written as a function of the spanwise coordinate,

$$\Gamma_b = \Gamma_b(y), \tag{2.62}$$

and the lifting line is commonly placed along the quarter-chord line, consistently with the classical aerodynamic-centre location of thin airfoil theory.

The vortex representation must be consistent with Kelvin's theorem and the Helmholtz vortex theorems introduced in Sec. 2.2. In particular, a bound vortex cannot be treated as an isolated filament ending inside the fluid. It must be connected to trailing vortices and to the wake system. In the classical steady interpretation, the starting vortex is assumed to have been convected far downstream, so that the remaining near-field structure is represented by a horseshoe vortex composed of a bound vortex segment and two trailing legs, as illustrated in Fig. 2.8.

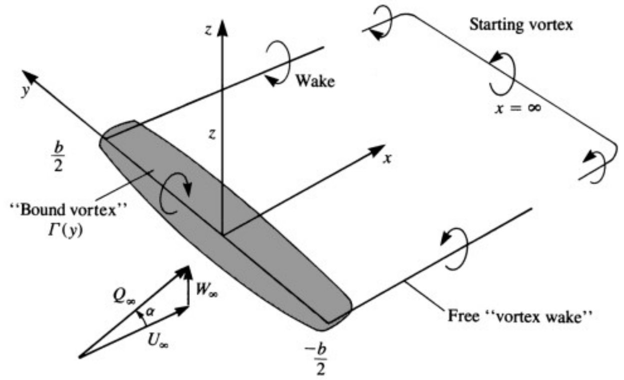


Figure 2.8: Prandtl lifting-line representation of a finite lifting surface [5].

The trailing vorticity is determined by the spatial variation of the bound circulation. If adjacent portions of the lifting line carry different circulation values, the wake must contain the circulation difference required to connect them. In continuous form, this relation can be written as

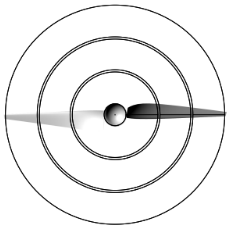
$$\gamma_t(y) = -\frac{d\Gamma_b}{dy}, \tag{2.63}$$

where the sign depends on the adopted orientation convention. This equation expresses the central idea that the wake vorticity is not prescribed independently, but is linked to the circulation distribution along the lifting line.

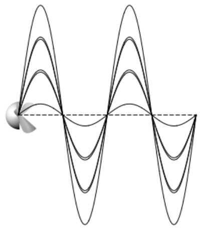
For propellers and rotors, the spanwise coordinate is replaced by the radial coordinate of the blade. Each blade is then represented by a radial distribution of bound circulation,

$$\Gamma_b = \Gamma_b(r). \tag{2.64}$$

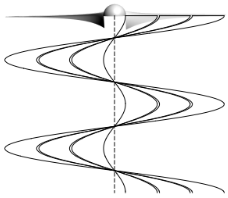
The classical lifting-line formulation for wings assumes trailing vorticity convected downstream along the freestream direction, but rotor applications require the wake to follow the combined effect of axial convection and blade rotation. The resulting helical vortex system can be discretised into straight vortex filaments, allowing the induced velocity to be evaluated through the Biot–Savart law and the lifting-line concept to be extended to rotating blades [32]. A schematic representation is shown in Fig. 2.9.



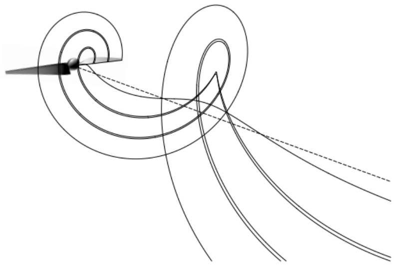
(a) **Front View**



(b) **Side View**



(c) **Top View**



(d) **Isometric View**

Figure 2.9: Views of the lifting-line representation of a rotating blade [32].

In numerical lifting-line methods, the continuous circulation distribution is discretised into a finite number of bound vortex segments. Each segment is associated with a circulation value $\Gamma_{b,i}$, and the local aerodynamic state is evaluated at a corresponding control point $\mathbf{x}_{\text{cp},i}$, usually located along the bound segment. The discrete counterpart of Eq. (2.63) is obtained from the difference between neighbouring bound-circulation values,

$$\Gamma_{\text{trail},i} = \Gamma_{b,i+1} - \Gamma_{b,i}, \tag{2.65}$$

up to the adopted sign convention for the trailing vortex orientation.

The induced velocity at a control point is obtained by superposing the contribution of all vortex segments,

$$\mathbf{V}_{\text{ind}}(\mathbf{x}_{\text{cp},i}) = \sum_{j=1}^{N_{\text{seg}}} \mathbf{V}_{\text{ind},j}(\mathbf{x}_{\text{cp},i}), \tag{2.66}$$

where the individual segment contribution is evaluated through the finite-segment Biot–Savart expression introduced in Eq. (2.45). The ideal vortex representation remains singular, and practical implementations must therefore define a consistent treatment of self-induced contributions and vortex-core regularisation.

2.3.2 Sectional Aerodynamic State

The lifting-line model provides the induced velocity generated by the vortex system, but the circulation distribution still requires a sectional aerodynamic closure. This closure is obtained by treating each blade element as a locally two-dimensional airfoil section subject to an effective relative velocity [2], [6], [7], [33], [34].

Consider a blade section located at radius r, with local chord $c(r)$ and pitch angle $\theta(r)$. The local relative velocity is decomposed into an effective axial component V_{ax} and an effective tangential component $V_{\tan}$. These components include the imposed operating condition and the induced velocity evaluated at the blade section, as illustrated in Fig. 2.10. The relative velocity magnitude is

$$V_{\mathrm{rel}} = \sqrt{V_{\mathrm{ax}}^2 + V_{\tan}^2}. \tag{2.67}$$

The local inflow angle is defined as

$$\varphi = \operatorname{atan2}\left(V_{\mathrm{ax}}, V_{\tan}\right), \tag{2.68}$$

and the local angle of attack is

$$\alpha = \theta - \varphi. \tag{2.69}$$

The sectional lift and drag coefficients generally depend on angle of attack, Reynolds number, and Mach number. In the low-fidelity framework considered here, compressibility effects are neglected and the Reynolds-number dependence is not explicitly updated at each section. The airfoil polar data are therefore treated primarily as functions of the local angle of attack,

$$C_l = C_l(\alpha), \qquad C_d = C_d(\alpha). \tag{2.70}$$

These data may be obtained from experiments, airfoil databases, or two-dimensional viscous-inviscid solvers such as XFOIL [35], [36]. This allows sectional viscous effects to be included at negligible computational cost, although the result depends on the quality and range of validity of the available polar data.

2.3.3 Kutta–Joukowski and Blade-Element Closure

The link between the force-based blade-element description and the vortex-based lifting-line description is provided by the Kutta–Joukowski theorem. For an incompressible inviscid flow around a lifting section, the lift per unit span is proportional to the bound circulation associated with that section [5], [37], [38],

$$L' = \rho V_{\mathrm{rel}} \Gamma_b. \tag{2.71}$$

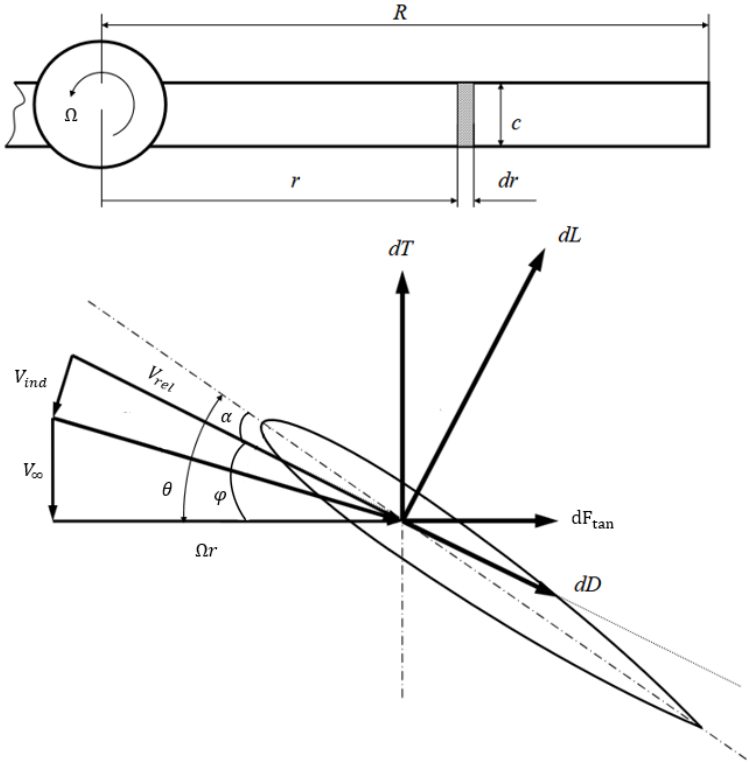


Figure 2.10: Local kinematics and force decomposition of a blade section.

The same sectional lift is provided by the blade-element relation,

$$L' = \frac{1}{2}\rho V_{\mathrm{rel}}^2 c C_l(\alpha). \tag{2.72}$$

Combining Eqs. (2.71) and (2.72) gives the local bound circulation,

$$\Gamma_b = \frac{1}{2}V_{\mathrm{rel}} c C_l(\alpha). \tag{2.73}$$

This equation is the sectional closure of the lifting-line problem. The circulation determines the wake and the induced velocity; the induced velocity modifies V_{rel} and α; the airfoil polar then determines $C_l(\alpha)$, and therefore updates the circulation. The problem is consequently coupled and generally nonlinear.

The sectional drag is obtained from the same polar data as

$$D' = \frac{1}{2}\rho V_{\mathrm{rel}}^2 c C_d(\alpha). \tag{2.74}$$

For a blade element of radial length dr,

$$dL = L'\,dr, \qquad dD = D'\,dr. \tag{2.75}$$

The lift force is perpendicular to the local relative velocity, while the drag force is aligned with it, as shown in Fig. 2.10. By projecting these forces onto the axial and tangential directions, the differential thrust and tangential force are

$$dT = dL\cos\varphi - dD\sin\varphi, \tag{2.76}$$

$$dF_{\text{tan}} = dL\sin\varphi + dD\cos\varphi. \tag{2.77}$$

The corresponding torque contribution is

$$dQ = r\,dF_{\text{tan}}. \tag{2.78}$$

Finally, for a propeller with N_b blades, the total thrust and torque are obtained by integrating the elemental contributions along the blade span,

$$T = N_b \int_{r_{\text{root}}}^{r_{\text{tip}}} dT, \tag{2.79}$$

$$Q = N_b \int_{r_{\text{root}}}^{r_{\text{tip}}} dQ. \tag{2.80}$$

The shaft power associated with the aerodynamic torque is

$$P = \Omega Q, \tag{2.81}$$

where Ω is the angular velocity magnitude of the propeller. The propulsive efficiency is

$$\eta = \frac{TV_\infty}{P}, \tag{2.82}$$

where V_∞ is the axial freestream velocity. These quantities provide the link between the sectional lifting-line solution and the global propeller performance.

2.4 Boundary-Integral Formulation for the Potential Problem

The objective of this section is to express the solution of the potential-flow problem introduced in Sec. 2.1 through an equivalent integral representation over the boundary of the fluid domain. Under the assumptions of incompressible, inviscid, and irrotational outer flow, the perturbation potential satisfies Laplace's equation,

$$\nabla^2\phi = 0 \qquad \text{in } \mathcal{V}. \tag{2.83}$$

A direct solution of Eq. (2.83) would require the potential field to be determined throughout the fluid domain. The boundary-integral approach avoids this volume formulation by representing the perturbation potential in terms of quantities defined only on the boundary surfaces. This transformation is obtained through Green's identities and provides the theoretical basis for source-and-doublet panel methods [4], [5]. The mathematical derivation of Green's identities is reported in Appendix A.2; only the resulting representation is used here.

2.4.1 Green Representation of the Perturbation Potential

Let $\mathbf{x}$ be the field point where the perturbation potential is evaluated, and let $\mathbf{x}_0$ denote a source point on the boundary of the fluid domain. The fundamental solution of the three-dimensional Laplace equation is

$$G(\mathbf{x}, \mathbf{x}_0) = \frac{1}{4\pi r}, \qquad r = |\mathbf{x} - \mathbf{x}_0|\,. \tag{2.84}$$

This function satisfies

$$\nabla_0^2 G(\mathbf{x}, \mathbf{x}_0) = -\delta(\mathbf{x} - \mathbf{x}_0), \tag{2.85}$$

where the differential operator is applied with respect to the source coordinates $\mathbf{x}_0$. Applying Green's third identity to the harmonic perturbation potential ϕ, the potential at an interior point of the fluid domain can be written as

$$\phi(\mathbf{x}) = \int_{\partial\mathcal{V}} \left[G(\mathbf{x}, \mathbf{x}_0) \frac{\partial \phi}{\partial n_{\mathcal{V}}}(\mathbf{x}_0) - \phi(\mathbf{x}_0) \frac{\partial G}{\partial n_{\mathcal{V}}}(\mathbf{x}, \mathbf{x}_0) \right] dS_0, \tag{2.86}$$

where $n_{\mathcal{V}}$ denotes the outward normal to the fluid domain. Equation (2.86) states that the perturbation potential inside the domain is completely determined by the potential and its normal derivative on the boundary.

When the field point approaches the boundary, the singularity of the fundamental solution must be treated through a limiting process. The boundary-integral representation is then written in the more general form

$$E(\mathbf{x})\phi(\mathbf{x}) = \int_{\partial\mathcal{V}} \left[G(\mathbf{x}, \mathbf{x}_0) \frac{\partial \phi}{\partial n_{\mathcal{V}}}(\mathbf{x}_0) - \phi(\mathbf{x}_0) \frac{\partial G}{\partial n_{\mathcal{V}}}(\mathbf{x}, \mathbf{x}_0) \right] dS_0. \tag{2.87}$$

The coefficient $E(\mathbf{x})$ depends on the position of the field point with respect to the fluid domain. For a smooth boundary, it is commonly written as

$$E(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in \mathcal{V}, \\ \dfrac{1}{2}, & \mathbf{x} \in \partial\mathcal{V}, \\ 0, & \mathbf{x} \notin \overline{\mathcal{V}}. \end{cases} \tag{2.88}$$

For use in a source-and-doublet panel formulation, Eq. (2.87) is rewritten using the surface normals and sign convention adopted for the panel influence coefficients. With this convention, the potential induced by a constant-strength source panel is written with a negative source contribution, while the potential induced by a constant-strength doublet panel is written as the negative normal derivative, with respect to the source coordinates, of the elementary source potential. Therefore, the elementary source and doublet contributions are expressed as

$$\phi_\sigma(\mathbf{x}) = -\frac{1}{4\pi}\int_{\mathcal{S}} \sigma(\mathbf{x}_0)\frac{1}{r}dS_0, \tag{2.89}$$

and

$$\phi_\mu(\mathbf{x}) = -\frac{1}{4\pi}\int_{\mathcal{S}} \mu(\mathbf{x}_0)\frac{\partial}{\partial n_0}\left(\frac{1}{r}\right)dS_0, \tag{2.90}$$

where n_0 is the panel normal used in the definition of the source and doublet influence coefficients, and the normal derivative is evaluated with respect to the source coordinates $\mathbf{x}_0$. Since

$$\nabla_{\mathbf{x}} G = -\nabla_{\mathbf{x}_0} G, \tag{2.91}$$

the doublet contribution may equivalently be expressed using the gradient with respect to the field coordinates. This convention is the one adopted for the analytical quadrilateral-panel expressions.

For a lifting configuration, the boundary of the fluid domain is decomposed as

$$\partial\mathcal{V} = \mathcal{S}_{\mathrm{B}} \cup \mathcal{S}_{\mathrm{w}} \cup \mathcal{S}_{\infty}, \tag{2.92}$$

where $\mathcal{S}_{\mathrm{B}}$ is the body surface, $\mathcal{S}_{\mathrm{w}}$ is the wake surface, and $\mathcal{S}_{\infty}$ is the far-field boundary. Since the perturbation potential and its gradient decay at infinity, the contribution of $\mathcal{S}_{\infty}$ vanishes.

On the body surface, the source strength is the quantity determined by the impermeability condition, whereas the body doublet strength represents the unknown perturbation-potential distribution of the source-and-doublet formulation. On the wake surface, no source distribution is assigned in the zero-thickness approximation. The wake contribution is instead represented by a doublet sheet whose strength is the potential jump across the wake,

$$\mu_{\mathrm{w}} = \Delta\phi_{\mathrm{w}} = [\phi]_{\mathcal{S}_{\mathrm{w}}} = \phi^{+} - \phi^{-}. \tag{2.93}$$

Using the panel convention of Eqs. (2.89) and (2.90), the source-and-doublet boundary representation can then be written as

$$E(\mathbf{x})\phi(\mathbf{x}) = -\frac{1}{4\pi}\int_{\mathcal{S}_{\mathrm{B}}} \sigma\frac{1}{r}dS_0 - \frac{1}{4\pi}\int_{\mathcal{S}_{\mathrm{B}}} \mu\frac{\partial}{\partial n_0}\left(\frac{1}{r}\right) dS_0 - \frac{1}{4\pi}\int_{\mathcal{S}_{\mathrm{w}}} \mu_{\mathrm{w}}\frac{\partial}{\partial n_0}\left(\frac{1}{r}\right) dS_0. \tag{2.94}$$

A numerical panel method is obtained by replacing the continuous source and doublet distributions in Eq. (2.94) with distributions over a finite number of surface elements, resulting in an algebraic linear system. The following subsection introduces the corresponding surface singularity elements and the influence coefficients used to construct this system.

2.4.2 Surface Singularity Distributions and Influence Coefficients

This subsection introduces the surface singularity elements used to discretise the boundary-integral formulation and to define the corresponding influence coefficients. These coefficients measure the potential induced at a field point by a unit-strength source or doublet distribution over a panel. The formulation follows the classical constant-strength quadrilateral panel expressions reported in [5], originally developed in the context of three-dimensional panel methods by Hess and Smith [39].

Let $\mathcal{S}$ be a surface element on which singularities are distributed, and let $\mathbf{x}$ be a field point. A generic source point on the surface is denoted by $\mathbf{x}_0$, and

$$r = |\mathbf{x} - \mathbf{x}_0| \tag{2.95}$$

is the distance between the field and source points. A continuous source distribution of strength $\sigma(\mathbf{x}_0)$ generates the potential

$$\phi_\sigma(\mathbf{x}) = -\int_{\mathcal{S}} \sigma(\mathbf{x}_0)G(\mathbf{x},\mathbf{x}_0)\ dS_0, \tag{2.96}$$

while a continuous doublet distribution of strength $\mu(\mathbf{x}_0)$ generates

$$\phi_\mu(\mathbf{x}) = -\int_{\mathcal{S}} \mu(\mathbf{x}_0)\frac{\partial G}{\partial n_0}(\mathbf{x},\mathbf{x}_0)\ dS_0. \tag{2.97}$$

The sign in Eq. (2.97) depends on the adopted convention for the doublet strength and surface normal. Here, the derivative is taken with respect to the source coordinates, consistently with Eq. (2.90).

The simplest three-dimensional panel element is obtained by assuming a flat quadrilateral panel with constant singularity strength, as shown in Fig. 2.11. The analytical formulas are written in a local panel coordinate system, where the panel lies in the plane $z = 0$. Its vertices are

$$\mathbf{x}_1 = (x_1, y_1, 0), \quad \mathbf{x}_2 = (x_2, y_2, 0), \quad \mathbf{x}_3 = (x_3, y_3, 0), \quad \mathbf{x}_4 = (x_4, y_4, 0), \tag{2.98}$$

and the field point is $\mathbf{x}(x, y, z)$. For a panel embedded in a global coordinate system, the field point and the panel vertices are first transformed into this local frame.

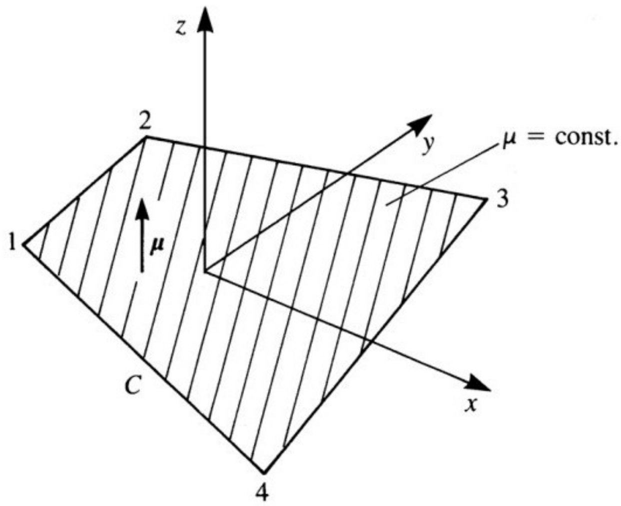


Figure 2.11: Constant quadrilateral doublet panel aligned with the local z-axis [5].

For a constant-strength source panel, the potential at $\mathbf{x}$ is

$$\phi_\sigma(x, y, z) = -\frac{\sigma}{4\pi} \int_S \frac{dS}{\sqrt{(x - x_0)^2 + (y - y_0)^2 + z^2}}. \tag{2.99}$$

For a constant-strength doublet panel oriented along the local z-direction, the potential is

$$\phi_\mu(x, y, z) = -\frac{\mu}{4\pi} \int_S \frac{z \, dS}{[(x - x_0)^2 + (y - y_0)^2 + z^2]^{3/2}}. \tag{2.100}$$

The sign convention in Eqs. (2.99) and (2.100) follows the panel formulation adopted in [5]. To write the analytical result compactly, define for each edge k of the quadrilateral

$$\Delta x_k = x_{k+1} - x_k, \qquad \Delta y_k = y_{k+1} - y_k, \qquad d_k = \sqrt{\Delta x_k^2 + \Delta y_k^2}, \qquad \mathbf{x}_5 \equiv \mathbf{x}_1. \tag{2.101}$$

The edge slope is

$$m_k = \frac{\Delta y_k}{\Delta x_k}, \tag{2.102}$$

and, for each vertex k,

$$r_k = \sqrt{(x - x_k)^2 + (y - y_k)^2 + z^2}, \tag{2.103}$$

$$e_k = (x - x_k)^2 + z^2, \qquad h_k = (x - x_k)(y - y_k). \tag{2.104}$$

The logarithmic contribution can be written as

$$\mathcal{L} = \sum_{k=1}^{4} \frac{(x - x_k)\Delta y_k - (y - y_k)\Delta x_k}{d_k} \ln\left(\frac{r_k + r_{k+1} + d_k}{r_k + r_{k+1} - d_k}\right), \tag{2.105}$$

while the angular contribution is

$$\Theta = \sum_{k=1}^{4} \left[\tan^{-1}\left(\frac{m_k e_k - h_k}{z r_k}\right) - \tan^{-1}\left(\frac{m_k e_{k+1} - h_{k+1}}{z r_{k+1}}\right)\right]. \tag{2.106}$$

Using these definitions, the analytical potential of the constant-strength source panel becomes

$$\phi_\sigma(x, y, z) = -\frac{\sigma}{4\pi}\left(\mathcal{L} - |z|\Theta\right). \tag{2.107}$$

The analytical potential of the constant-strength doublet panel is

$$\phi_\mu(x, y, z) = \frac{\mu}{4\pi}\Theta. \tag{2.108}$$

As the field point approaches the panel surface from the two sides, the doublet potential has the limiting behaviour

$$\phi_\mu(x, y, 0^\pm) = \mp\frac{\mu}{2}, \tag{2.109}$$

where 0^+ denotes the side toward which the local panel normal is directed and 0^- denotes the opposite side. In numerical implementations, equivalent limiting forms or robust arctangent evaluations are required when an edge is nearly parallel to the local y-axis and the slope m_k becomes ill-conditioned [40].

The analytical expressions of the constant-strength source and doublet panels provide the elementary contributions required to discretise the boundary-integral formulation. Once the body and wake surfaces are divided into panels, the potential induced at each field point by a unit-strength singularity distribution defines an influence coefficient.

Let $\mathbf{x}_i$ be the i-th field point and let $\mathcal{S}_j$ be the j-th panel. The source potential influence coefficient is defined as the potential induced at $\mathbf{x}_i$ by a unit-strength source distribution over panel j,

$$A_\phi^{ij} = \phi_{\sigma,j}(\mathbf{x}_i)\Big|_{\sigma_j=1}. \tag{2.110}$$

Using the analytical expression of the quadrilateral source panel, Eq. (2.107), this coefficient becomes

$$A_\phi^{ij} = -\frac{1}{4\pi}\left(\mathcal{L}_{ij} - |z_i|\Theta_{ij}\right), \tag{2.111}$$

where $\mathcal{L}_{ij}$ and Θ_{ij} are evaluated after expressing the field point $\mathbf{x}_i$ in the local coordinate system of panel j.

Similarly, the doublet potential influence coefficient is defined as the potential induced at $\mathbf{x}_i$ by a unit-strength doublet distribution over panel j,

$$B_\phi^{ij} = \phi_{\mu,j}(\mathbf{x}_i)\Big|_{\mu_j=1}. \tag{2.112}$$

Using Eq. (2.108), this gives

$$B_\phi^{ij} = \frac{1}{4\pi}\Theta_{ij}. \tag{2.113}$$

When the field point lies on the surface, the limiting behaviour of the doublet influence must be treated consistently with Eq. (2.109).

2.4.3 Discrete Boundary-Integral Formulation

The influence coefficients introduced above allow the continuous source-and-doublet boundary representation to be written as a finite-dimensional algebraic relation. At a given time level, the body and wake surfaces are approximated by a finite number of panels,

$$\mathcal{S}_\mathrm{B} \simeq \bigcup_{j=1}^{N_\mathrm{B}} \mathcal{S}_{B,j}, \qquad \mathcal{S}_\mathrm{w} \simeq \bigcup_{k=1}^{N_\mathrm{w}} \mathcal{S}_{\mathrm{w},k}, \tag{2.114}$$

where N_B is the number of body panels and N_w is the number of wake panels. A control point $\mathbf{x}_i$ is associated with each body panel, and the boundary-integral equation is enforced at these control points.

In a source-and-doublet formulation, the source strength on the j-th body panel is denoted by σ_j, while the corresponding body doublet strength is denoted by μ_j. Wake panels contribute through their doublet strengths, denoted by $\mu_{\mathrm{w},k}$. The wake doublet influence coefficient at the i-th body control point is denoted by C_ϕ^{ik}.

With the adopted convention, the body doublet strength represents the value of the perturbation potential on the corresponding body panel. For a smooth body surface, the limiting coefficient of the boundary-integral equation is $E_i = 1/2$. Therefore,

$$\sum_{j=1}^{N_\mathrm{B}} B_\phi^{ij}\mu_j = -\sum_{j=1}^{N_\mathrm{B}} A_\phi^{ij}\sigma_j - \sum_{k=1}^{N_\mathrm{w}} C_\phi^{ik}\mu_{\mathrm{w},k}. \tag{2.115}$$

where

$$B_\phi^{ij} = \begin{cases} -\dfrac{1}{2}, & i = j, \\ \dfrac{1}{4\pi}\Theta_{ij}, & i \neq j. \end{cases} \tag{2.116}$$

Here, A_ϕ^{ij} is the source potential influence coefficient of body panel j on control point i, B_ϕ^{ij} is the body-doublet coefficient including the limiting self-influence contribution, and C_ϕ^{ik} is the wake doublet influence coefficient. Equation (2.115) is the algebraic counterpart of the continuous boundary-integral formulation. At this stage, it is still a representation formula: the source strengths and the wake doublet strengths must be specified through the boundary conditions and the wake closure.

In matrix form, Eq. (2.115) becomes

$$\mathbf{B}_\phi\boldsymbol{\mu} = \mathbf{RHS}, \tag{2.117}$$

where the entries of the right-hand side are

$$RHS_i = -\sum_{j=1}^{N_\mathrm{B}} A_\phi^{ij}\sigma_j - \sum_{k=1}^{N_\mathrm{w}} C_\phi^{ik}\mu_{\mathrm{w},k}. \tag{2.118}$$

The algebraic system in Eq. (2.117) becomes closed once the source strengths σ_j and the wake doublet strengths $\mu_{\mathrm{w},k}$ have been specified.

The source strengths are obtained from the impermeability condition imposed on the body surface, while the wake doublet strengths are assigned through the Kutta condition and the wake history. Once these contributions have been evaluated, the only remaining unknowns are the body doublet strengths collected in $\boldsymbol{\mu}$.

For a moving body, the normal component of the relative velocity between the fluid and the body must vanish on the body surface,

$$\mathbf{n}_i \cdot (\nabla\Phi_i - \mathbf{V}_{B,i}) = 0, \tag{2.119}$$

where $\mathbf{n}_i$ is the outward normal at the i-th control point and $\mathbf{V}_{\mathrm{B},i}$ is the local velocity of the body surface. Using the decomposition $\Phi = \Phi_\infty + \phi$, this condition becomes

$$\mathbf{n}_i \cdot \nabla\phi_i = \mathbf{n}_i \cdot (\mathbf{V}_{B,i} - \mathbf{V}_{\infty,i}), \tag{2.120}$$

where $\mathbf{V}_{\infty,i} = \nabla\Phi_{\infty,i}$. With the convention in which the body source strength represents the normal derivative of the perturbation potential, the source strength at the i-th body panel is

$$\sigma_i = \mathbf{n}_i \cdot (\mathbf{V}_{B,i} - \mathbf{V}_{\infty,i}) = \mathbf{n}_i \cdot (\mathbf{V}_0 + \boldsymbol{\Omega} \times \mathbf{r}_i - \mathbf{V}_{\infty,i}). \tag{2.121}$$

The sign of Eq. (2.121) depends on the adopted convention for the source strength and for the surface normal. Once this convention is fixed, the source strengths are known quantities and form the body-source contribution to the right-hand side of Eq. (2.117).

The remaining closure required for lifting flows is provided by the Kutta condition. In a doublet formulation, the circulation shed from the trailing edge is represented by the doublet strength assigned to the wake panels. In its usual discrete form, the strength of a newly generated wake panel is related to the jump of body doublet strength at the trailing edge (TE),

$$\mu_{\mathrm{w}} = \mu_{\mathrm{upper,TE}} - \mu_{\mathrm{lower,TE}}, \tag{2.122}$$

up to the sign convention adopted for the wake orientation and for the definition of the doublet strength.

In an explicit unsteady formulation, the wake doublets appearing in the current linear system are known from the previous wake history. For example, the first wake panel used at time level n may be assigned as

$$\mu^n_{\mathrm{w},1} = \mu^{n-1}_{2N_c} - \mu^{n-1}_1. \tag{2.123}$$

Here, $2N_c$ and 1 are the chordwise indeces of the upper and lower trailing-edge panels, respectively. The wake-panel orientation and the corresponding coefficient C^{ik}_ϕ must be defined consistently with this potential-jump convention.

After solving the body problem at time level n, the strength of the newly shed wake panel to be used at the next time level is updated and the previously shed wake panels retain the doublet strengths assigned when they were released,

$$\mu_{\mathrm{w},k+1}^{n+1} = \mu_{\mathrm{w},k}^{n}. \tag{2.124}$$

Thus, the wake doublet strengths are known contributions determined by the previous lifting history and by the Kutta condition.

Once the linear system in Eq. (2.117) has been solved, the body doublet strengths $\boldsymbol{\mu}$ are known. Together with the prescribed source strengths and the wake doublet strengths, they determine the perturbation potential associated with the source-and-doublet representation. The velocity field can then be recovered from the spatial derivatives of the potential, and the pressure field follows from Bernoulli's equation.

The evaluation of surface velocities, pressure coefficients, aerodynamic loads, and wake convection belongs to the post-processing and time-marching stages of the numerical method. These aspects depend on the specific discretisation and implementation choices adopted in VKI-SDPM and are therefore discussed in Chapter 5. Consequently, the theoretical formulation presented here is considered complete once the linear system for the unknown body doublet strengths has been obtained and solved.

Chapter 3

VKI-BELL: Formulation and Implementation

3.1 Introduction to VKI-BELL

VKI-BELL is a MATLAB-based code inspired by the work of van Garrel [41]. It is intended to capture the dominant induced-velocity effects of a rotating lifting system while preserving a relatively compact and computationally efficient formulation.

The solver is based on a lifting-line representation of the blades coupled with a sectional aerodynamic closure. This class of methods is well established in the literature and has been shown to be suitable for fast steady analyses, while retaining the flexibility required for future extensions [18], [42], [43].

With respect to BEMT-based approaches, it provides a more explicit representation of the three-dimensional vortex-induced velocity field and of the mutual interaction among blade sections, blades, and wake elements [42]. Compared with a surface panel method, it is less geometrically general because the blade is represented by a lifting line rather than by a finite-thickness surface, and the present implementation does not include unsteady time marching.

The following sections describe the solver workflow, the rotor-geometry construction, the wake model, the vortex-segment assembly, the sectional aerodynamic closure, the fixed-point solution strategy, and the post-processing of rotor loads.

3.2 Solver Workflow and Setup

The purpose of this section is to provide a high-level map of the VKI-BELL execution logic; the individual modelling and numerical blocks are then discussed in detail in the following sections.

The VKI-BELL implementation is organised around a configuration-driven workflow. The user defines the case to be analysed and the execution mode in run_BELL.m, which acts as the entry point of the code. The actual simulation sequence is then managed by main_BELL.m, where the main blocks of the solver are executed in a fixed order.

The simulation setup is collected in a configuration structure, denoted here as cfg, whose components are listed in Eq. (3.1).

$$\mathrm{cfg} = \left\{ \begin{array}{l} \text{rotor and blade geometry} \\ \text{operating conditions} \\ \text{polar-data settings} \\ \text{wake settings} \\ \text{solver parameters} \\ \text{post-processing options} \end{array} \right\}. \tag{3.1}$$

The general workflow of VKI-BELL is schematically illustrated in Fig. 3.1.

The rotor geometry is first generated from the configuration file, including the lifting-line discretisation of each blade.

The core aerodynamic solution is computed in run_solver.m. For a given prescribed wake, the solver iteratively updates the bound-circulation distribution until the sectional aerodynamic closure and the vortex-induced velocity field are mutually consistent [18], [41].

An additional outer loop may be activated to rebuild the wake using an updated estimate of the mean induced velocity and the empirical wake-shape laws. This strategy leads to a so-called semi-prescribed wake treatment [42].

After convergence, the sectional quantities are post-processed to obtain radial load distributions and integral rotor performance quantities. The main output of the code is the results structure, which stores the initial configuration together with the circulation distribution, the sectional data, the convergence history, and the integrated loads.

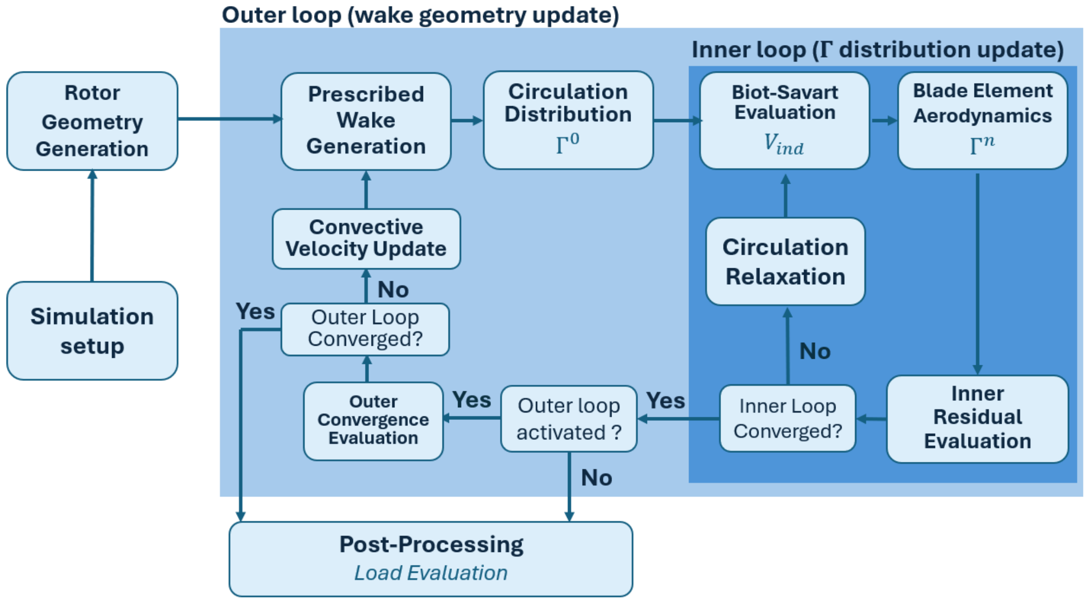


Figure 3.1: Block diagram of the VKI-BELL workflow.

3.3 Rotor Geometry and Lifting-Line Discretisation

The rotor-geometry block converts the configuration inputs into the discrete blade geometry used by the aerodynamic solver. This operation is performed in build_rotor_geometry.m, which constructs the reference blade, generates the complete multi-blade rotor, defines the local panel quantities, and stores the resulting data in the `bladeGeom` structure.

The blade is discretised along the radial direction into N_r panels. The radial panel edges are generated between the hub and tip radii, which are specified in the configuration file. Two radial distributions are currently available: uniform spacing and cosine spacing. The chord and twist distributions provided in the configuration are interpolated at the radial edge stations.

The pitch-axis line is first defined from the hub position, $\mathbf{x}_{\text{hub}}$, and the user-defined spanwise direction $\hat{\mathbf{m}}$, while the leading-edge and trailing-edge points are obtained by placing the chord around this axis. For a generic radial station, this operation can be written as

$$\begin{aligned} \mathbf{x}_r &= \mathbf{x}_{\text{hub}} + r\ \hat{\mathbf{m}} \\ \mathbf{x}_{LE} &= \mathbf{x}_r + \bar{x}_{PA}\ c(r)\ \hat{\mathbf{l}}_c \\ \mathbf{x}_{TE} &= \mathbf{x}_r + (1 - \bar{x}_{PA})\ c(r)\ \hat{\mathbf{l}}_c, \end{aligned} \tag{3.2}$$

where $\bar{x}_{PA}$ is the pitch-axis location along the chord, expressed as a chord fraction, $c(r)$ is the local chord, and $\hat{\mathbf{l}}_c$ is the chordwise unit vector related to the radial station. This is obtained by representing the local chord as a straight segment once it has been rotated around $\hat{\mathbf{m}}$ by the sum of the local twist angle θ_{tw} and collective pitch θ_c.

The panel geometry nomenclature and construction follow the convention adopted by Van Garrel in [41], as reported in Fig. 3.2. Each blade panel is described by four geometric corner points,

$$\mathbf{x}_1 = \mathrm{LE}_{\mathrm{inner}}, \quad \mathbf{x}_2 = \mathrm{TE}_{\mathrm{inner}}, \quad \mathbf{x}_3 = \mathrm{TE}_{\mathrm{outer}}, \quad \mathbf{x}_4 = \mathrm{LE}_{\mathrm{outer}} \tag{3.3}$$

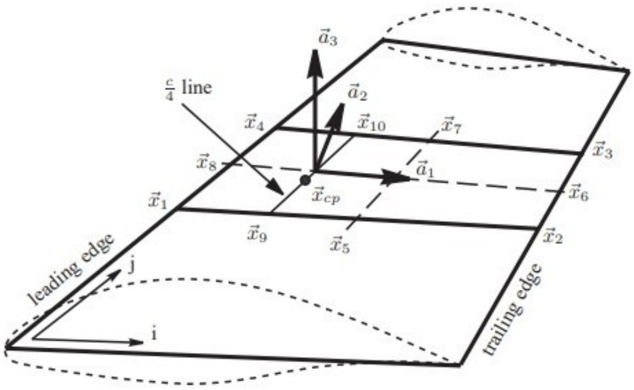


Figure 3.2: Geometric nomenclature for blade panels [41].

Additional midpoint and quarter-chord points are then computed and stored. In particular, the lifting-line segment of the panel is defined by the two quarter-chord points

$$\mathbf{x}_9 = 0.75\ \mathbf{x}_1 + 0.25\ \mathbf{x}_2, \qquad \mathbf{x}_{10} = 0.75\ \mathbf{x}_4 + 0.25\ \mathbf{x}_3, \tag{3.4}$$

and the corresponding bound segment unit vector is

$$\mathbf{dl}_j = \frac{\mathbf{x}_{10,j} - \mathbf{x}_{9,j}}{|\mathbf{x}_{10,j} - \mathbf{x}_{9,j}|}. \tag{3.5}$$

This segment represents the discrete lifting line associated with the blade panel. The control point at which the aerodynamic quantities are evaluated is placed halfway along the quarter-chord segment.

For each panel, a local frame is defined and used later for velocity projection and sectional aerodynamic evaluation. With reference to Fig. 3.2, the panel chordwise unit vector, $\hat{\mathbf{l}}$, is obtained from the midpoint of the leading-edge side to the midpoint of the trailing-edge side. The local normal unit vector, $\hat{\mathbf{n}}$, is then computed from the chordwise direction and the bound-segment unit vector. In compact form,

$$\hat{\mathbf{l}}_j = \frac{\mathbf{x}_{6,j} - \mathbf{x}_{8,j}}{|\mathbf{x}_{6,j} - \mathbf{x}_{8,j}|}, \qquad \hat{\mathbf{n}}_j = \frac{\hat{\mathbf{l}}_j \times \mathbf{dl}_j}{\left|\hat{\mathbf{l}}_j \times \mathbf{dl}_j\right|}, \qquad \text{for} \quad j = 1, ..., N_r \tag{3.6}$$

The panel area is also evaluated from the cross product between the diagonal midpoint vectors.

The complete rotor is obtained by first defining a reference blade and then replicating it through azimuthal rotations for the remaining blades.

The airfoil profile associated with each panel is assigned during the geometry-building stage. The present implementation supports either a constant airfoil along the blade or a spanwise assignment over selected radial intervals. The actual polar data are loaded separately, but the airfoil identifier allows each control point to be connected with the correct sectional aerodynamic data during the solver iterations.

3.4 Wake Modelling

The discrete vortex geometry that forms the wake is used by VKI-BELL to compute the induced velocity at the blade control points.

This geometry is constructed before the fixed-point iterations and is kept fixed throughout each inner loop. Therefore, the vortex-system geometry is prescribed, while the vortex strengths are updated consistently with the current bound-circulation distribution.

The wake is represented by a structured lattice of nodes attached to the blade trailing edge. The wake-node positions are stored as

$$\mathbf{x}_{\mathrm{w}} = \mathbf{x}_{\mathrm{w}}(b, k, w), \tag{3.7}$$

where b is the blade index, k is the radial-edge index, and w is the wake-age index.

Once the wake-node lattice has been generated, adjacent radial edges and wake-age rows are connected to form quadrilateral wake panels. For a generic panel associated with blade b, radial-edge station k, and wake-age interval w, the geometric vertex ordering convention is

$$\begin{aligned} \mathbf{P}_{1,\mathrm{w}} &= \mathbf{x}_{\mathrm{w}}(b, k, w), \\ \mathbf{P}_{2,\mathrm{w}} &= \mathbf{x}_{\mathrm{w}}(b, k, w+1), \\ \mathbf{P}_{3,\mathrm{w}} &= \mathbf{x}_{\mathrm{w}}(b, k+1, w+1), \\ \mathbf{P}_{4,\mathrm{w}} &= \mathbf{x}_{\mathrm{w}}(b, k+1, w). \end{aligned} \tag{3.8}$$

This convention is consistent with the one used for the blade panels in Eq. (3.3).

With this convention and a consistent circulation assignment, the blade trailing-edge contribution and the first wake-panel edge, at $w = 1$, can cancel algebraically, enforcing the Kutta-consistent connection between blade and wake. The following rows, $w > 1$, are generated downstream by combining the rotor azimuthal motion with axial convection.

The wake azimuthal spacing is defined by the number of wake segments per revolution, N_{seg}, as

$$\Delta\psi = \frac{2\pi}{N_{\text{seg}}}. \tag{3.9}$$

Starting from the trailing-edge location, each wake node is rotated by one azimuthal increment and convected with a velocity composed of a free-stream contribution and an induced-velocity contribution,

$$\mathbf{V}_{\text{conv}} = \mathbf{V}_\infty + \mathbf{V}_{\text{ind}}. \tag{3.10}$$

The use of the induced contribution depends on the operating condition. In axial flight, the free-stream velocity already provides a finite axial convection of the wake. In that case, the induced-velocity estimate may be set to zero or used only to obtain a more representative initial wake pitch. In hover, $\mathbf{V}_\infty = 0$; therefore, the induced contribution is necessary to prevent the wake geometry from collapsing onto the rotor plane.

When an induced contribution is prescribed, its initial value is computed from an actuator-disc-type guess based either on an initial thrust estimate T_0 or on an initial thrust coefficient C_T^0 [3], [43]. The magnitude is then computed as

$$\mathrm{V}_{\text{ind}}^0 = \begin{cases} \sqrt{\dfrac{T_0}{2\rho\pi R^2}}, & \text{if } T_0 \text{ is prescribed,} \\ \Omega R\sqrt{\dfrac{C_T^0}{2}}, & \text{if } C_T^0 \text{ is prescribed.} \end{cases} \tag{3.11}$$

3.4.1 Wake Update Strategy

Although VKI-BELL does not implement a time-marching loop, the prescribed wake geometry can still be progressively adjusted through an outer relaxation procedure, as illustrated in Fig. 3.1.

In the outer loop, the induced-velocity estimate is updated after convergence of the inner fixed-point solution. For hover calculations, the outer loop is always enabled, whereas in axial flight its activation is left to the user. At the n-th outer loop step, the target scalar estimate is obtained as the mean magnitude of the axial component of the induced velocity over all blades and blade panels,

$$\tilde{\mathrm{V}}_{\text{ind}}^n = \frac{1}{N_b N_r} \sum_{b=1}^{N_b} \sum_{j=1}^{N_r} \left| \mathbf{V}_{\text{ind}}^n(b,j) \cdot \hat{\mathbf{e}}_{\text{rot}} \right|, \tag{3.12}$$

where $\hat{\mathbf{e}}_{\text{rot}}$ is the user-defined rotor axis.

The current induced-velocity estimate is then relaxed towards this target before rebuilding the wake,

$$\mathrm{V}_{\text{ind}}^{n+1} = \mathrm{V}_{\text{ind}}^{n} + \omega_{\text{w}} \left(\tilde{\mathrm{V}}_{\text{ind}}^{n} - \mathrm{V}_{\text{ind}}^{n} \right), \tag{3.13}$$

where ω_{w} is the wake relaxation factor. The relaxed value can then be used in Eq. (3.10) to reconstruct the wake geometry before the next inner fixed-point solution. Thus, in the semi-prescribed treatment the wake remains frozen during each inner circulation iteration but is progressively updated between outer iterations [5], [42].

Two additional wake-shape corrections can be selected in the current implementation, both acting on the convected wake nodes after Eq. (3.10) and before the wake panels are assembled. The first is a McCormick-type radial contraction [6], [44], in which the radial position of each wake point is multiplied by a contraction factor depending on the axial distance,

$$k_c(d_{\text{ax}}) = k_{c,\infty} + (1 - k_{c,\infty}) \exp\left(-\frac{d_{\text{ax}}}{R} \right), \tag{3.14}$$

where d_{ax} is the orthogonal axial distance from the rotor plane and

$$k_{c,\infty} = \sqrt{\frac{1}{2}\left(1 + \sqrt{\max(1 - C_T, 0)} \right)}, \tag{3.15}$$

The corrected radial distance is then obtained by applying the contraction factor $k_c(d_{\text{ax}})$ to the original radial position.

The McCormick correction is smooth and monotonic in axial distance, and, being derived under moderate disc loading, it produces only a weak contraction for the thrust coefficients typical of edgewise rotors and propellers.

For hover, where the wake remains close to the rotor plane over many revolutions, a more pronounced radial contraction is required. The second option is an empirical Landgrebe-type model [6], [45], [46], in which every wake filament, including the tip filament, contracts gradually with wake age according to the same exponential law,

$$k_c(\psi_{\text{w}}) = k_{\text{Lan}} + (1 - k_{\text{Lan}}) \exp\left[-(0.145 + 27\, C_T)\psi_{\text{w}} \right], \tag{3.16}$$

where ψ_{w} is the wake azimuthal age and $k_{\text{Lan}} = 0.78$ is the asymptotic tip-vortex contraction ratio. The tip filament contracts directly toward $k_{\text{Lan}} \cdot R$, whereas an inner filament contracts proportionally to $k_c(\psi_{\text{w}})$, so that the whole trailing sheet contracts consistently rather than only the outermost filament.

The Landgrebe model can be complemented by an empirical axial settling law for the tip filament, again expressed as a function of wake age and of the thrust coefficient,

$$d_{\text{tip}}(\psi_{\text{w}}) = d_{\text{tip}}(0) + R\left[k_1 \min(\psi_{\text{w}}, 2\pi/N_b) + k_2 \max(\psi_{\text{w}} - 2\pi/N_b, 0) \right], \tag{3.17}$$

where the value $2\pi/N_b$ represents the wake age at the first blade passage and

$$k_1 = -0.25\left(\frac{C_T\ 2\pi\ R}{N_b\ c} + 0.001\,\theta_{\text{tw}}\right), \qquad k_2 = -\left(1.41 + 0.0141\,\theta_{\text{tw}}\right)\sqrt{\frac{C_T}{2}}, \tag{3.18}$$

where θ_{tw} is the average blade twist between root and tip [46]. This law reproduces the axial descent of the tip vortex observed experimentally in hover, with a slower settling rate before the first blade passage and a faster rate afterwards. The inner filaments are not subject to this axial law and continue to be convected according to Eq. (3.10).

3.4.2 Induced Velocity Computation and Regularisation

During the vortex-segment assembly stage, each panel is decomposed into its four oriented edges. Shared edges are identified through a unique topological representation and their circulation contributions are summed algebraically following Eqs. (3.3) and (3.8). The output of this operation is the finite set of vortex segments used for the induced-velocity calculation,

$$\texttt{vort\_struct} = \{\mathbf{x}_{1,s}, \mathbf{x}_{2,s}, \Gamma_s, r_{c,s}\}_{s=1}^{N_s}\,, \tag{3.19}$$

where $\mathbf{x}_{1,s}$ and $\mathbf{x}_{2,s}$ are the segment endpoints, Γ_s is the net segment circulation, and $r_{c,s}$ is the vortex-core radius assigned to the segment. This assembly procedure also stores the topological origin of each active segment, which is used to define the induction mask needed to remove the self-induced contribution of each bound segment at its corresponding control point.

The induced velocity is obtained by applying the finite-segment Biot–Savart kernel to the active segment list. The implementation is vectorised over both control points and vortex segments. A finite-core regularisation based on the Lamb–Oseen model is applied through the segment core radius $r_{c,s}$. The core radius of a wake segment grows with wake age through a core-spreading law introduced in Subsec. 2.2.3,

$$r_c(\zeta) = \sqrt{r_{c0}^2 + 4\,\alpha_{\text{Oseen}}\,\nu\,\delta\,\frac{\zeta}{\Omega}}, \tag{3.20}$$

where r_{c0} is the initial core radius, ζ is the wake age of the segment, Ω is the angular velocity, ν is the kinematic viscosity, $\alpha_{\text{Oseen}} = 1.25643$ is the Oseen constant, and δ is the diffusion parameter. The core-spreading law is primarily responsible for regularising the induced velocity close to the root and tip vortex filaments, where the wake would otherwise remain at an almost constant, small core radius throughout its length; it is also a prerequisite for the tip axial settling law of Eq. (3.17), for which an artificially thin and non-growing core would otherwise produce an unrealistically large induced velocity as the tip vortex passes close to the following blade.

3.5 Sectional Aerodynamic Closure

The sectional aerodynamic closure provides the relation between the local flow conditions at the blade control points and the bound-circulation distribution. In VKI-BELL, this operation is performed at each fixed-point iteration after the induced velocity has been evaluated from the current vortex system. The output of the aerodynamic closure is a target circulation distribution, denoted here as $\tilde{\Gamma}$, which is then used by the iterative solver to update the bound circulation.

The aerodynamic data are prescribed through two-dimensional airfoil polars, either computed with software such as XFOIL [36] or taken from literature sources such as Abbott [35]. For each airfoil used along the blade, the corresponding polar file contains the angle of attack, the lift coefficient, and the drag coefficient, the latter including viscous contributions,

$$\texttt{polar\_file.dat} = [\alpha,\ C_l,\ C_d]\,. \tag{3.21}$$

The polar data are loaded before the solver iterations and are associated with each blade panel through the airfoil identifier stored in `bladeGeom.airfoilName`. During the sectional aerodynamic evaluation, the local angle of attack is used to interpolate the corresponding values of $C_l(\alpha)$ and $C_d(\alpha)$.

For each blade b and panel j, the relative velocity at the control point is computed as

$$\mathbf{V}_{\text{rel}}(b,j) = \mathbf{V}_\infty - \boldsymbol{\Omega} \times \mathbf{x}_{\text{CP}}(b,j) + \mathbf{V}_{\text{ind,CP}}(b,j), \tag{3.22}$$

where $\mathbf{x}_{\text{CP}}$ is the position vector of the control point with respect to the hub, $\boldsymbol{\Omega}$ is the rotor angular-velocity vector, and $\mathbf{V}_{\text{ind,CP}}$ is the induced velocity evaluated at the same control point.

The relative velocity is then projected onto the local sectional frame. The tangential and normal components are

$$V_{\text{tan}} = \mathbf{V}_{\text{rel}} \cdot \hat{\mathbf{l}}, \qquad V_{\text{norm}} = \mathbf{V}_{\text{rel}} \cdot \hat{\mathbf{n}}, \tag{3.23}$$

where $\hat{\mathbf{l}}$ and $\hat{\mathbf{n}}$ are the local chordwise and normal directions defined in Eq. (3.6). The local angle of attack is then evaluated as

$$\alpha = \operatorname{atan2}\left(V_{\text{norm}}, V_{\text{tan}}\right), \tag{3.24}$$

implemented through the two-argument arctangent function in order to preserve the correct quadrant. The resulting angle is then used to interpolate the airfoil polar assigned to the corresponding panel. If the angle of attack falls outside the tabulated range, as may occur during the first inner iterations, different options are available for evaluating the aerodynamic coefficients, including linear extrapolation, clamping, and a flat-plate approximation [2], [47]. When the flat-plate option is selected, the aerodynamic coefficients are computed as

$$C_l = 2\sin(\alpha)\cos(\alpha), \qquad C_d = 2\sin^2(\alpha). \tag{3.25}$$

For compactness, the pair of indices (b, j), identifying blade b and panel j, is hereafter replaced by a single panel index p. Equivalently, for a rotor with N_r panels per blade, one may write $p = (b-1)N_r + j$. All quantities carrying the subscript p are therefore evaluated at the control point of the corresponding blade panel.

The interpolated lift coefficient is used to compute the target bound circulation through the Kutta–Joukowski theorem, following the formulation adopted in [18], [41]. For each blade panel, the circulation in the aerodynamic convention is written as

$$\tilde{\Gamma}_p = \frac{1}{2} C_l(\alpha_p) \frac{\left[V_{\mathrm{tan},p}^2 + V_{\mathrm{norm},p}^2\right] dS_p}{\sqrt{\left[(\mathbf{V}_{\mathrm{rel},p} \times \mathbf{dl}_p) \cdot \hat{\mathbf{n}}_p\right]^2 + \left[(\mathbf{V}_{\mathrm{rel},p} \times \mathbf{dl}_p) \cdot \hat{\mathbf{l}}_p\right]^2}}, \tag{3.26}$$

where dS_p is the blade-panel area and $\mathbf{dl}_p$ is the bound lifting-line segment from Eq. (3.5). The quantity $\tilde{\Gamma}_p$ represents the circulation value returned by the sectional aerodynamic closure for panel p and used within the iterative circulation update.

The drag coefficient does not enter the circulation update. It is nevertheless stored together with the local flow quantities and is used later during the load evaluation and post-processing stage. The aerodynamic closure therefore returns both the target circulation and a set of sectional quantities, including relative-velocity components, angle of attack, lift and drag coefficients, and circulation values.

3.6 Solution Strategy and Post-Processing

Once the blade geometry, the wake model, and the sectional aerodynamic closure have been defined, the remaining problem consists of finding a bound-circulation distribution that is consistent with the velocity field induced by the same vortex system. VKI-BELL solves this coupled non-linear problem through a fixed-point iterative procedure implemented in `run_solver.m`. The same non-linear problem could also be solved with more advanced iterative methods, such as the Newton–Raphson method, but these are not included in the current implementation.

Because the wake geometry is prescribed and frozen for the whole inner loop, the mapping from the bound-circulation vector to the induced velocity at the control points is linear. The vortex-segment assembly already expresses the net segment circulation as a linear combination of the panel circulations; combining this coefficient matrix with the Biot–Savart kernel evaluated once for the current wake geometry yields an influence matrix $\mathbf{I}_{\mathrm{BS}}$ such that

$$\mathbf{V}_{\mathrm{ind,CP}}^m = \mathbf{I}_{\mathrm{BS}}\, \Gamma_{\mathrm{ar}}^m, \tag{3.27}$$

where m denotes the inner-loop iteration index and Γ^m_{ar} is the bound-circulation array obtained from Γ^m_p through the same panel-index mapping $p = (b-1)N_r + j$ introduced in Sec. 3.5. The matrix $\mathbf{I}_{\mathrm{BS}}$ depends only on the current wake and blade geometry and is therefore assembled once per outer iteration. With $\mathbf{I}_{\mathrm{BS}}$ available, every inner iteration reduces to a matrix–vector product, which is considerably cheaper than repeating the full vortex-segment assembly and Biot–Savart evaluation at each step.

For a prescribed wake geometry, the inner solution loop starts from an initial circulation distribution Γ^0_{ar}. The current implementation supports either a zero initial circulation or an aerodynamic initialisation from Eq. (3.26) using the relative velocity without the induced contribution. At a generic inner iteration m, the solution sequence can be summarised as

1. the bound-circulation distribution Γ^m_p is assigned to the blade panels;
2. the induced velocity $\mathbf{V}^m_{\mathrm{ind,CP}}$ is evaluated at the blade panel control points through the influence matrix of Eq. (3.27);
3. the local relative velocity, angle of attack, and aerodynamic coefficients are computed;
4. the target circulation distribution $\tilde{\Gamma}^m_p$ is obtained from the sectional aerodynamic closure;
5. the residual is evaluated;
6. if the convergence criterion is not met, the current circulation is relaxed towards the target distribution $\tilde{\Gamma}^m_p$, and the updated value Γ^{m+1}_p is assigned before restarting the cycle from step 1.

The convergence of the inner loop is monitored through a residual computed on the circulation distribution,

$$\mathcal{R}^m(\Gamma) = \frac{\max_p \left|\tilde{\Gamma}^m_p - \Gamma^m_p\right|}{\left|\mathrm{mean}_p(\tilde{\Gamma}^m_p)\right|}, \tag{3.28}$$

where the numerator measures the maximum discrepancy between the target and current circulation distributions, and the denominator provides a characteristic circulation scale based on the absolute mean value of the target circulation. The inner fixed-point solution is considered converged when

$$\mathcal{R}^m(\Gamma) < \varepsilon_\Gamma, \tag{3.29}$$

where ε_Γ is the tolerance specified in the configuration file.

Before the next inner iteration, the circulation is updated by

$$\Gamma_p^{m+1} = \Gamma_p^m + \omega_\Gamma \left(\tilde{\Gamma}_p^m - \Gamma_p^m\right), \tag{3.30}$$

where ω_Γ is the circulation relaxation factor. This relaxation is especially needed during the first iterations, when the wake geometry, induced velocity field, and sectional aerodynamic state may still be poorly matched.

The circulation obtained in Eq. (3.30) is assigned to the bound-vortex system following the sign convention adopted in the vortex-segment assembly

$$\boldsymbol{\Gamma}_p^{m+1} = \Gamma_p^{m+1}\ \mathbf{dl}_p. \tag{3.31}$$

If the outer loop is activated, the converged induced-velocity field is used to compute the new wake geometry. The influence matrix $\mathbf{I}_{\mathrm{BS}}$ is then reassembled for the updated wake geometry, and steps 1–6 are repeated.

The outer loop is stopped when the updated mean induced velocity differs from the previous outer-loop estimate by less than a prescribed relative tolerance,

$$\frac{\left|\mathrm{V}_{\mathrm{ind}}^{\mathrm{n}+1} - \mathrm{V}_{\mathrm{ind}}^{n}\right|}{\left|\mathrm{V}_{\mathrm{ind}}^{n}\right|} < \varepsilon_{\mathrm{outer}}. \tag{3.32}$$

If this condition is not satisfied, the induced-velocity estimate is relaxed and the wake geometry is rebuilt before starting a new inner fixed-point solution.

After convergence, the aerodynamic loads are computed in `compute_loads.m`. The force acting on each blade panel is evaluated as the sum of a Kutta–Joukowski lift contribution and a drag contribution,

$$\mathbf{L}_p = \rho\left[\mathbf{V}_{\mathrm{rel},p} \times \boldsymbol{\Gamma}_p\right], \qquad \mathbf{D}_p = \frac{1}{2}\rho\left|\mathbf{V}_{\mathrm{rel},p}\right|^2 S_p C_d(\alpha_p)\frac{\mathbf{V}_{\mathrm{rel},p}}{\left|\mathbf{V}_{\mathrm{rel},p}\right|} \tag{3.33}$$

$$\mathbf{F}_p = \mathbf{L}_p + \mathbf{D}_p. \tag{3.34}$$

Here, ρ is the air density, S_p is the panel area, and $C_d(\alpha_p)$ is the drag coefficient obtained from the sectional polar lookup using the converged sectional value of α_p.

The sectional thrust and aerodynamic torque contributions are then obtained by projecting the panel force onto the rotor axis and by taking the moment about the same axis,

$$\Delta T_p = \mathbf{F}_p \cdot \hat{\mathbf{e}}_{\mathrm{rot}}, \qquad \Delta Q_p = \left[(\mathbf{x}_{\mathrm{CP},p} - \mathbf{x}_{\mathrm{hub}}) \times \mathbf{F}_p\right] \cdot \hat{\mathbf{e}}_{\mathrm{rot}}. \tag{3.35}$$

The global rotor loads are obtained by summing all of the $N_b N_r$ panel contributions

$$T = \sum_{p=1}^{N_b N_r} \Delta T_p, \qquad Q = \sum_{p=1}^{N_b N_r} \Delta Q_p. \tag{3.36}$$

The non-dimensional performance coefficients are evaluated by default using the rotor-disc area and the tip speed $V_{\text{tip}} = \Omega R$,

$$C_T = \frac{T}{\rho \pi R^2 V_{\text{tip}}^2}, \qquad C_Q = \frac{Q}{\rho \pi R^3 V_{\text{tip}}^2}, \tag{3.37}$$

The complete output is collected in the `results` structure, which contains the configuration setup, blade geometry, wake geometry, circulation distribution, and sectional data.

Chapter 4

VKI-BELL: Sensitivity Analysis and Validation

This chapter validates VKI-BELL implementation described in Ch. 3. The test case selected for validation is the Caradonna–Tung test case [12]. This represents a classical benchmark for numerical solvers within rotorcraft applications [48], [49], [50].

Firstly the chapter defines the experimental setup of the original test case and reports the sensitivity studies used to select the numerical framework for this test case. Then it moves to the comparison of VKI-BELL against the experimental measurements and the in-house BEMT solver from VKI (VKI-BEMT) for a code-to-code comparison. Finally the results obtained are discussed and the conclusions regarding the validity of VKI-BELL are drawn.

4.1 Caradonna–Tung Test Case (NASA-TM-81232)

The experimental configuration considered in the Caradonna–Tung study is a two-bladed model helicopter rotor tested in hover flight. The experiment was carried out in the Army Aeromechanics Laboratory hover test facility, which consisted of a large chamber equipped with special ducting designed to reduce room recirculation effects. The rotor was located near the centre of the chamber and was mounted on a column containing the drive shaft. A hot-wire sensor was placed below the rotor plane in order to measure the tip-vortex trajectories.

The rotor consisted of two cantilever-mounted, rectangular, untwisted, and untapered blades. The blade sections were based on the symmetric NACA 0012 airfoil, and each blade had an aspect ratio of $AR = 6$. A schematic of the model and the experimental setup is shown in Fig. 4.1.

The original report presents data for several collective pitch settings, over a range of rotor speeds, including low-subsonic and transonic tip-speed regimes. For the present validation, the configuration with collective pitch of 8° and rotational speed of 1250 rpm is chosen for availability and comparability of the experimental results. The main geometrical and operating information of the test case is summarised in Tab. 4.1.

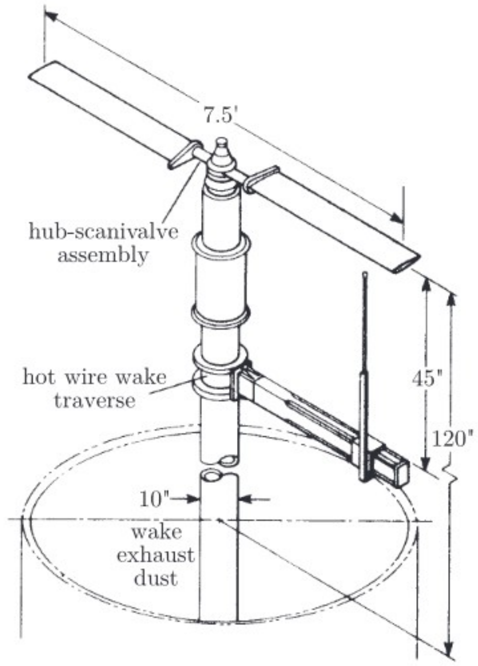


Figure 4.1: Schematic of the experimental setup for the Caradonna-Tung test case [12].

Table 4.1: Parameters used to configure VKI-BELL and VKI-BEMT for the Caradonna–Tung test case.

Parameter	**Symbol**	**Value**
Rotor radius	R	1.143 m
Number of blades	N_b	2
Blade chord (constant)	c	0.1905 m
Aspect ratio	AR	6
Blade twist	θ_{tw}	0°
Airfoil section	—	NACA 0012
Collective pitch	θ_c	8°
Rotor speed	Ω	1250 rpm

4.2 Sensitivity Analysis and Numerical Setup

The discretisation independence study is carried out on the spanwise panel count N_r, the azimuthal wake discretisation N_{seg}, and the prescribed-wake length N_{w}. Following the aforementioned order, first the radial mesh is varied while the other two parameters are kept fixed and set to conservative values. Once the parameter is selected from the sensitivity results, the next one is varied, and so on until the analysis is complete. Once the discretisation parameters are chosen, the analysis of the vortex regularisation is carried out. For the whole study, the sensitivity is measured in terms of relative differences of thrust coefficient between two consecutive refinements, evaluated as follows

$$\Delta\% = \frac{\left|C_T^k - C_T^{k-1}\right|}{\left|C_T^{k-1}\right|} \cdot 100 \tag{4.1}$$

Generally, only refinements yielding relative differences below 0.5% are considered, and the final value is selected among them.

The first sensitivity analysis is about the spanwise discretisation, which is the only discretisation parameter directly associated with the blade geometry. In fact, in the current implementation, VKI-BELL does not allow any chordwise discretisation. The selected spanwise spacing law is the cosine-type. The initial wake parameters are set to $N_{\mathrm{seg}} = 36$, placing nodes every 10°, and a total number of revolutions of $N_{\mathrm{w}} = 10$.

In Fig. 4.2 the mesh sensitivity is reported. The 0.5% threshold is reached around 20 panels; nevertheless a monotonic behaviour is preserved for the rest of the sweep, therefore a more conservative value of $N_r = 30$ is chosen and kept constant.

Following this, Figure 4.3 shows the sensitivity to the azimuthal discretisation. Here, the threshold is reached around $N_{\mathrm{seg}} = 36$; however, a plateau is reached beyond 60 segments per turn, and $N_{\mathrm{seg}} = 72$ is chosen as a conservative value.

Finally, the remaining free parameter is the wake length expressed as the number of revolutions. As reported in Fig. 4.4, $N_{\mathrm{w}} = 9$ is enough to stay under 0.5% but a clearer plateau develops for higher values. Therefore the selected value is $N_{\mathrm{w}} = 15$.

Having defined the full discretisation, the sensitivity to regularisation parameters is analysed. VKI-BELL uses a prescribed-wake geometry; therefore, the viscous parameters are needed for the stability of the fixed-point convergence rather than for wake convection. For this purpose, both the initial core size $r_{c,0}$ and the diffusion parameter δ are varied within reference values reported in the literature. For helicopter rotors in hover, the vortex-core size is typically set at 10% of the reference chord, while δ is in the range between 100 and 1000 [6], [19]. Variations within these ranges are therefore explored in the analysis.

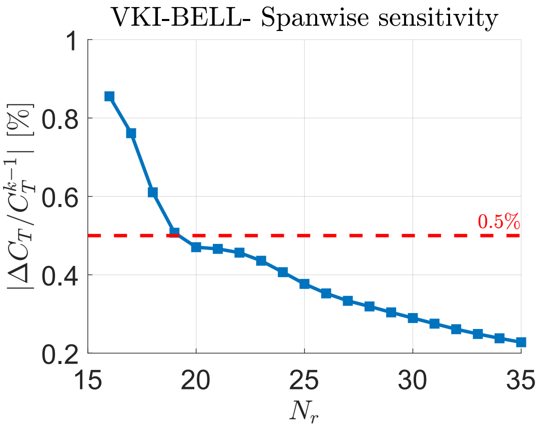


Figure 4.2: VKI-BELL spanwise mesh sensitivity.

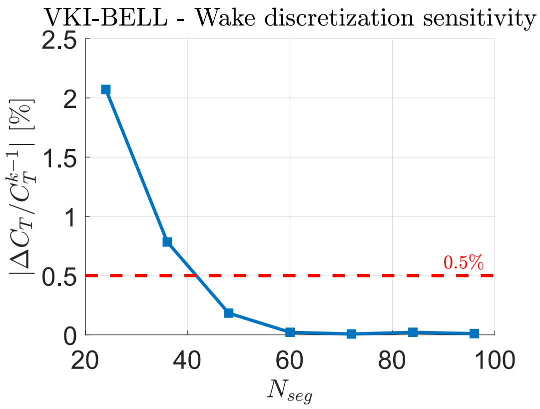


Figure 4.3: VKI-BELL wake discretisation sensitivity.

From Figure 4.5, it can be seen that while the δ parameter has no effective influence on the thrust coefficient, the dependence on $r_{c,0}$ tends to be quite relevant. The declared threshold is closely reached by $r_{c,0} = 0.01$ m but a more suitable value is found to be $r_{c,0} = 0.005$ m and therefore selected. The value for δ is set to 100 to avoid introducing unnecessary numerical viscosity.

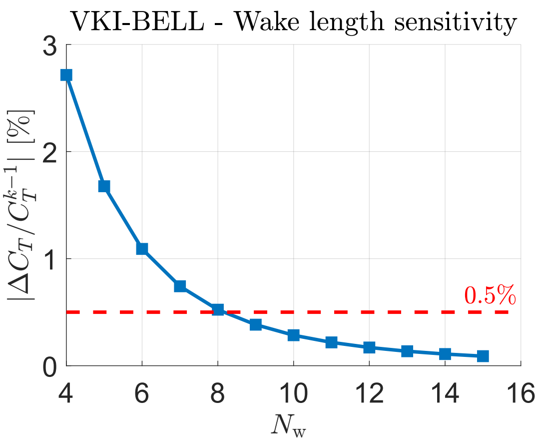


Figure 4.4: VKI-BELL wake length sensitivity.

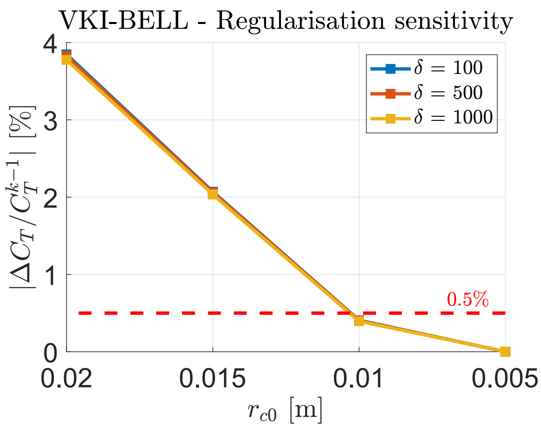


Figure 4.5: VKI-BELL sensitivity to vortex core regularisation.

In conclusion, the results of the sensitivity studies together with additional numerical settings are summarised in Table 4.2. The Landgrebe models are chosen for both the wake contraction and tip axial trajectory with the experimental value of C_T used as input; however, the validity and effects of these wake-model choices are discussed in greater detail in the following section.

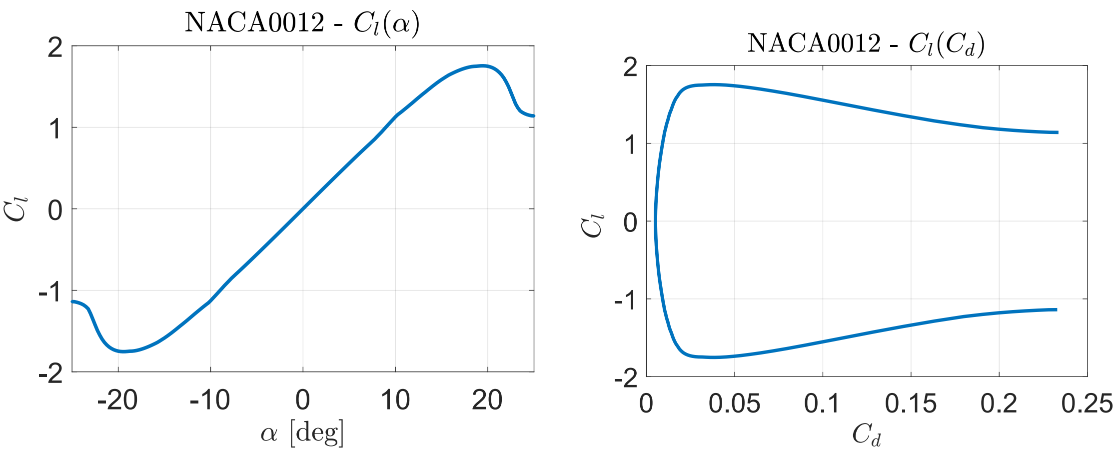
Table 4.2: Numerical setup for VKI-BELL validation runs.

Parameter	**Selected**
Spanwise panels	30
Wake segments per turn	72
Wake revolutions	15
Wake regularisation	$r_{c,0} = 5 \cdot 10^{-3}$[m], $\delta = 100$
Wake-contraction model	Landgrebe
Tip-vortex axial-settling model	Landgrebe
Outer loop iterations	Enabled

To conclude the numerical setup, choices regarding VKI-BEMT and the polar data are briefly reported below.

Uniform radial discretisation is the only option available in VKI-BEMT and the value of $N_r = 30$ is set for a more consistent comparison between the two solvers. The BEMT theoretical background can be found in App. A.4.

Both solvers use the same NACA0012 airfoil data tables. These are obtained via XFOIL including viscous contributions at a constant value of $Re = 10^6$, while at the tip the estimated value reaches $Re_{\text{tip}} \approx 2.3 \cdot 10^6$. No compressibility effects are considered. For a more accurate simulation, each section should use airfoil data evaluated at the local $Re(r)$ and $M(r)$. However, this implementation would introduce only second-order effects, which are not of primary interest in the present validation [2]. The airfoil data are reported in Fig. 4.6 in terms of $C_l(\alpha)$ and $C_l(C_d)$.



(a) Lift curve $C_l(\alpha)$ for NACA0012 airfoil profile.

(b) Polar curve $C_l(C_d)$ for NACA0012 airfoil profile.

Figure 4.6: NACA0012 aerodynamic curves used by VKI-BELL and VKI-BEMT for the code-to-code comparison.

4.3 Results and Discussion

The results used for the comparison in the validation below concern global thrust prediction, tip-vortex trajectory, and spanwise load distribution. Some numerical and physical considerations about the models involved are reported when deemed necessary for the interpretation of the results.

Before comparing the obtained results, the numerical convergence of VKI-BELL is discussed. As reported in Tab. 4.2, the outer loop is activated to reconcile the wake geometry with the predicted mean induced-velocity field. For this reason, the solver computes both inner and outer iterations. For both, a residual threshold is set to $\varepsilon = 10^{-4}$, and relaxation parameters are set to $\omega_\Gamma = 0.1$ and $\omega_\mathrm{w} = 0.9$. This leads to an average of 40 iterations per inner loop and 6 outer iterations. Figure 4.7 reports the residual evolution in the last outer iteration and a quasi-linear convergence behaviour can be observed. As already stated in Ch. 3, alternative iterative strategies may be implemented to solve the non-linear problem. However, the solution time does not currently justify implementing a more complex non-linear solver.

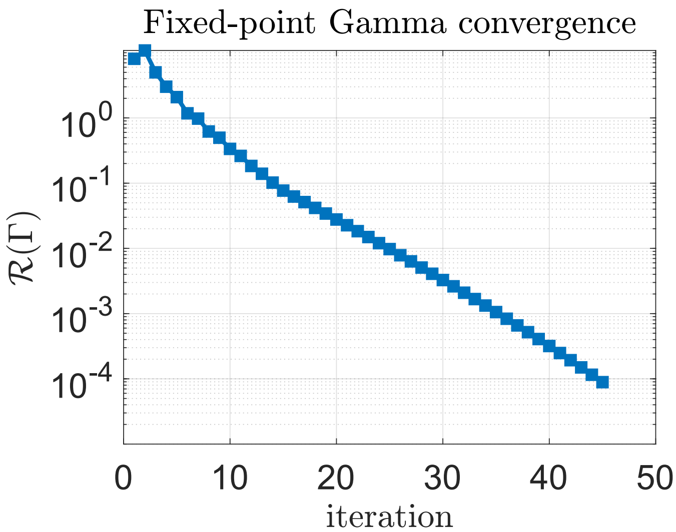


Figure 4.7: Fixed-point convergence for VKI-BELL hover simulation at last outer-loop iteration.

4.3.1 Thrust Coefficient Prediction and Simulation Analysis

Table 4.3 reports a comparison between the thrust coefficients predicted by the two solvers and the experimental measurement. It can be seen that VKI-BELL provides a closer estimate than VKI-BEMT, but with both solvers still overpredicting it. The other comparison available is done by looking at the simulation times; VKI-BEMT is approximately 25 times faster than VKI-BELL. This is mainly because of the differences in structures of the two solvers: VKI-BELL computes the induced velocity matrix $\boldsymbol{I}_{\mathrm{BS}}$ at the beginning of each outer iteration and then iterates with it, and the stiffness of the problem influences the convergence rate through the relaxation factors. In contrast, VKI-BEMT requires only one iterative loop without computing any mutual influence between sections and the wake contribution enters only through the momentum-derived induction factors [2]. However, excluding the matrix computation, the average inner-loop time for VKI-BELL is about 0.10 s, comparable with VKI-BEMT whole execution time.

Table 4.3: Results for the hover configuration.

Source	C_T	ΔC_T [%]	Wall-clock time [s]
Experiment	0.00459	—	—
VKI-BELL	0.00519	+13.07	10.71
VKI-BEMT	0.00574	+25.05	0.42

4.3.2 Tip-Vortex Trajectory

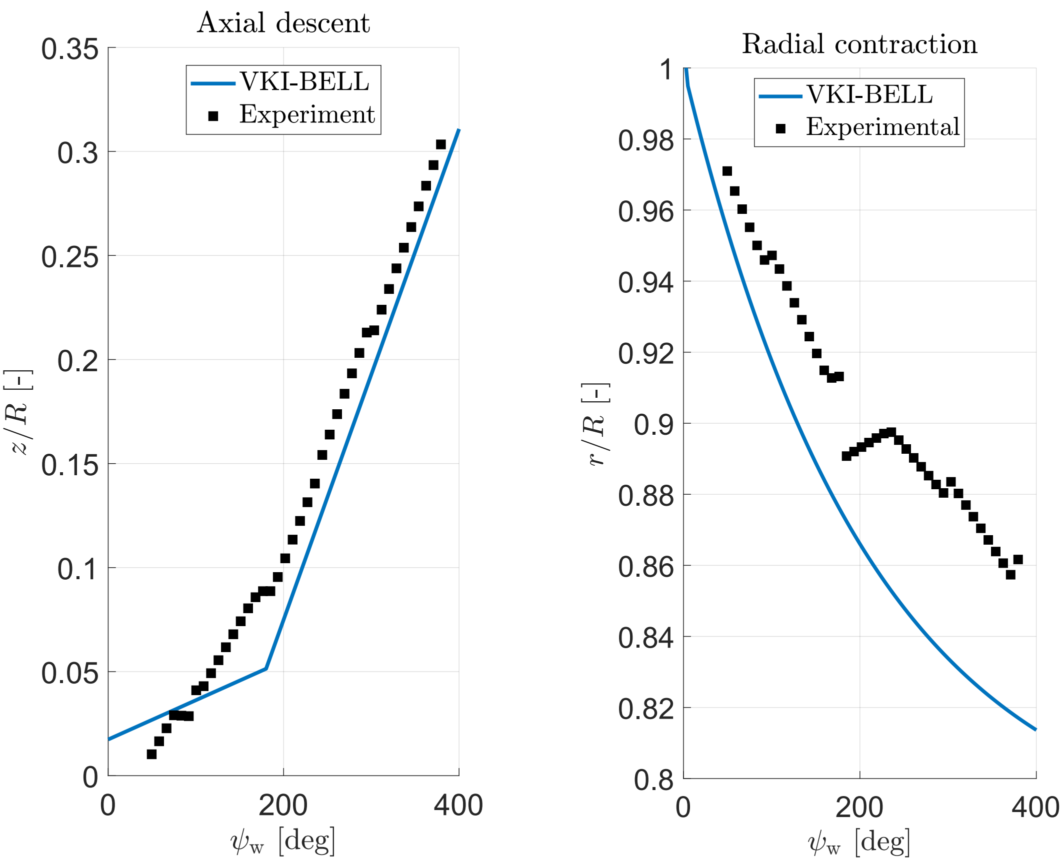
The wake representation for VKI-BELL relies on the Landgrebe models combined with a wake relaxation update. Test runs have proven that a pure relaxation strategy prevents the solver from reaching a physical solution in hover configuration. This may be caused by the lack of a local convection strategy that convects each filament of the wake with the local induced velocity. However, even such a solution would still be an approximation since it is known from theory that the flow accelerates downstream of the rotor disc in propulsive applications. If the wake is not placed in a physically correct way the induced-velocity field negatively affects the blade loading. To address this issue, the Landgrebe empirical model is adopted for the axial trajectory of the tip vortex, which represents the most influential structure of the wake system [45], [46].

The results of such implementation are reported in Fig. 4.8a and 4.8b, where the wake trajectory is decomposed into its axial and radial components, respectively, and compared against the measured results.

For the axial descent the main difference is noticeable around $\psi_{\mathrm{w}} = 180°$, where the computed position is almost half of the measured one. This location corresponds to the passage of the tip-vortex right under the opposite blade, as shown in Fig. 4.9. This anticipates a higher local induction near the tip region than in the physical case. In this situation stabilisation with a bigger core radius would excessively damp the induced field in the remaining part of the domain as well.

The reason behind such a discrepancy between the experiment and the numerical results lies in the empirical coefficients k_1 and k_2 used in the Landgrebe relation in Eq. (3.17), and the experimental value of C_T used in the computation of the wake trajectories. The empirical model was originally formulated for medium- to highly loaded rotors whereas the Caradonna–Tung rotor generates relatively low thrust compared with the reference rotors used in [45]. This therefore results in an underestimated axial displacement of the tip vortex.

A similar issue with prescribed-wake representation is reported in the original test case [12]. On that occasion, the prescribed model used was the one formulated by Kocurek [51] and implemented in the numerical solver reported in [52]. The resulting global thrust was overestimated by about 20% with respect to the measured one. The solution then adopted was to prescribe the tip trajectory by interpolating the experimental measurements, therefore obtaining a better estimate of the sectional and integral loads.



(a) Axial descent.

(b) Radial contraction.

Figure 4.8: Comparison of the tip-vortex axial trajectory and radial contraction between VKI-BELL predictions and experimental measurements.

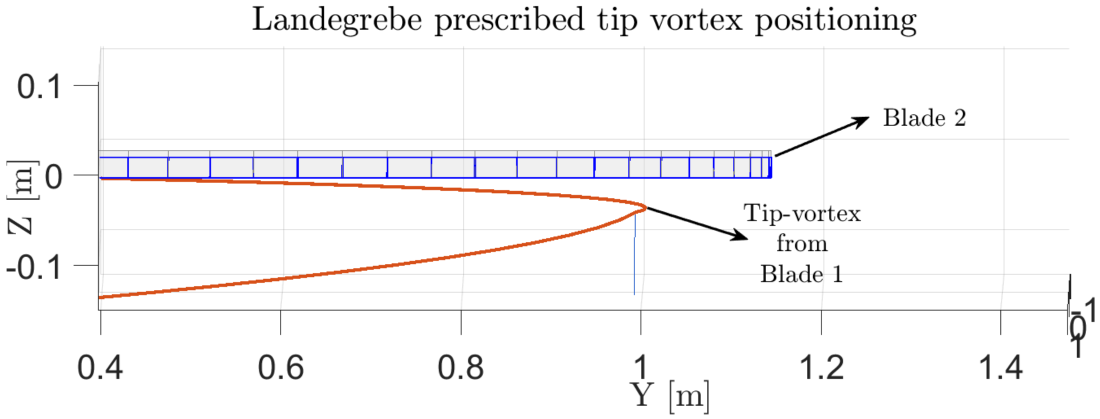


Figure 4.9: Snapshot of the tip vortex location in the near wake region following the Landgrebe tip-vortex trajectory.

4.3.3 Comparison of Spanwise Distributions

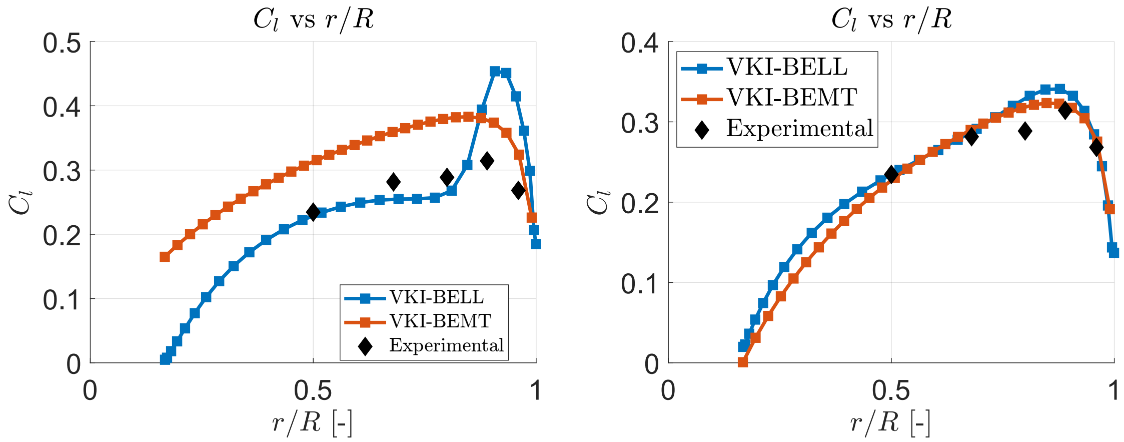
In the previous section the issue related to the positioning of the tip vortex has been introduced. In this section the effects are quantified in terms of load distributions. Even though a solution was proposed by Caradonna in [12], reproducing it is beyond the scope of the present work; however, a simpler diagnostic approach is later proposed to isolate the source of the discrepancy.

The validation proceeds with the analysis of the spanwise loading distributions. Experimental data are available for the lift coefficient spanwise distribution obtained by integration of pressure measurements at the experimental radial stations. Figure 4.10a reports the results for VKI-BELL and VKI-BEMT, compared against the experimental data. VKI-BEMT overpredicts the general trend, which is then reflected in the thrust coefficient overestimate. VKI-BELL captures most of the spanwise trend but in proximity of the tip a peak is detected. This peak results from the excessive proximity of the tip vortex, which modifies the blade loading near the tip.

To verify the source of the discrepancy, a diagnostic test is performed using the same numerical setup but disabling the Landgrebe tip-vortex trajectory. At this point as said in the previous section, VKI-BELL cannot reach a physically consistent solution without such a model, therefore an additional axial inflow equal to 2% of the tip speed (≈ 3 m/s) is imposed. This numerical artifact helps the development of the wake relaxation and was inspired by the work of Verstraete in [48] where a similar solution is applied to solvers with free-wake capabilities to more easily convect the starting vortices downstream in hover configuration.

The results from this study are reported in Fig. 4.10b. The same inflow was applied to VKI-BEMT for consistency even though, because of its formulation, it does not present any such problems. It can be seen that an improved coherence in the distribution is obtained by simply moving the tip vortex downstream, away from the rotor disc. This is also reflected in a better estimate for the global thrust coefficient that gets closer to the experimental value reported in Tab. 4.4. VKI-BEMT results are getting closer not because of the removal of an anomaly but because of the shift in the rotor operating point.

In any case, an additional comment on the inherently different nature of the two solvers can be made by looking at the tip of the blade. In VKI-BEMT a tip-loss correction is implemented to model the finite blade effect. VKI-BELL does not require such correction because of its three-dimensional formulation; therefore the drop at the tip is more physical and not imposed.



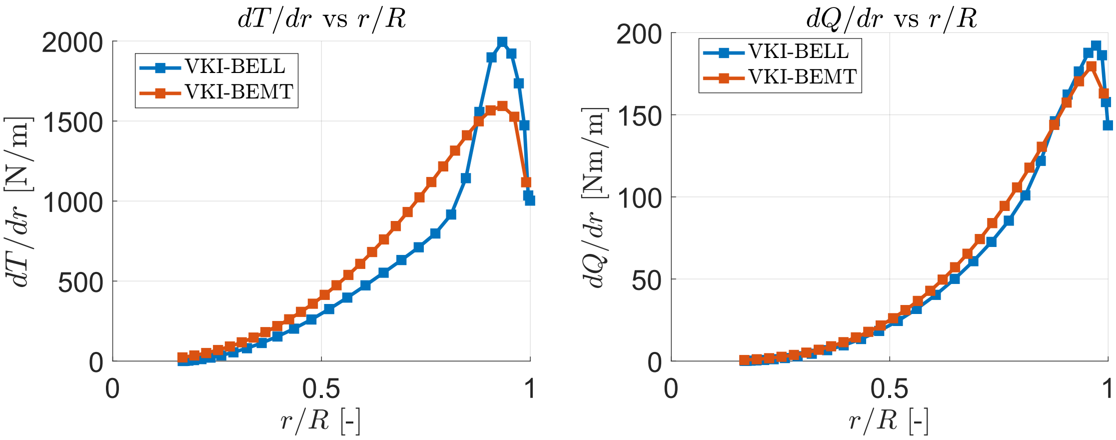
(a) Hover condition ($V_\infty = 0$). **(b)** Inflow condition ($V_\infty = 0.02\ V_{\text{tip}}$).

Figure 4.10: Spanwise lift-coefficient distribution $C_l(r/R)$ for hover and small inflow conditions.

Table 4.4: Results for the small inflow diagnostic test.

Source	C_T	ΔC_T [%]	Wall-clock time [s]
Experiment	0.00459	—	—
VKI-BELL	0.00489	+6.54	7.12
VKI-BEMT	0.00468	+1.96	0.38

To complete the comparison between the two solvers the radial distribution of thrust and torque are reported in Fig. 4.11 in the original hover condition. For VKI-BELL it can be seen that the thrust distribution is affected by the presence of the tip vortex in a similar manner to the $C_l(r/R)$ distribution. This is reasonable since lift is the main contribution to the thrust in this hover condition at low collective pitch $\theta_c = 8°$. Instead, the torque is less affected by it since the tangential component of the lift is relatively small at this angle, and the contribution is mainly drag-related.



(a) Thrust radial distribution dT/dr. **(b)** Torque radial distribution dQ/dr.

Figure 4.11: Thrust and torque radial distributions comparison between VKI-BELL and VKI-BEMT in hover condition.

4.4 Conclusions and Remarks

Overall, VKI-BELL reproduces the qualitative and, to a reasonable extent, the quantitative behaviour of the Caradonna–Tung rotor in hover, for both the integrated thrust and the spanwise loading. The overall accuracy is increased with respect to VKI-BEMT, even though local inconsistencies are still present and are related to the wake geometry. The computational cost is higher but remains reasonably low and within the range of solvers of the same class reported in [1].

The present validation does not include a proper axial-flight condition, but the diagnostic results with a small imposed inflow look promising if compared with those obtained with VKI-BEMT.

At the same time the validation also exposes the main limitations of the method in this operating condition. In hover configuration, the wake-relaxation strategy does not yield a physically consistent wake geometry. Therefore, VKI-BELL has to rely on empirical or semi-analytical models to overcome such limitation, leading to the results of the previous section.

A natural extension of the code would be to adopt a free-wake formulation which would increase the computational cost and would move VKI-BELL beyond the low-fidelity range of methods [1]. An alternative would be to extend the library of empirical wake models and data-fitted relations available in the solver, further extending its range of validity while retaining low computational cost.

Nevertheless, this solution highlights both the advantages and the limitations of these simplified solvers such as VKI-BELL and VKI-BEMT. Their main strength is their low computational cost, which is achieved by relying on external models and data sources for computationally expensive aspects of the solution, such as wake modelling. Improving their accuracy therefore requires broader and more accurate external models and databases. The desire for a more independent and versatile solver motivated the development of the higher-fidelity VKI-SDPM, presented and validated in the following chapters.

Chapter 5

VKI-SDPM: Formulation and Implementation

5.1 Introduction to VKI-SDPM

The VKI-SDPM solver is a MATLAB-based code for the source–doublet representation of the perturbation potential over a closed body surface. Within this approach, the source distribution directly accounts for the thickness contribution, whereas doublet singularities represent the lifting part of the solution. The theoretical background is reported in Chapter 2.

The computational framework is capable of treating both rotating and fixed aerodynamic configurations, such as propellers, rotors, and wings. In this context, the body motion is prescribed and the wake may be treated either as a prescribed wake determined by the freestream velocity or as a fully convected free wake.

The chapter focuses on the numerical formulation and implementation choices that define VKI-SDPM as a working solver. The solver architecture and simulation setup are first introduced, followed by the geometry and surface discretisation. The discrete potential problem is then described through the influence matrices, boundary conditions, and linear-system solution. The wake representation and evolution are subsequently presented, and the complete time-marching sequence is summarised before discussing the evaluation of surface velocities, pressures, and aerodynamic loads.

A practical code reference is provided in Appendix C, where the expected configuration-file format, saved `.mat` outputs, and a table with the main routines are reported.

5.2 Solver Architecture and Simulation Setup

The VKI-SDPM execution is organised around the driver Main_SDPM.m, which coordinates the complete simulation sequence from case selection to post-processing. The present section introduces the simulation configuration, the temporal discretisation, and the high-level workflow. The individual numerical modules are discussed in the following sections.

5.2.1 Simulation Configuration and Temporal Discretisation

The simulation setup creates a MATLAB structure containing the test-case inputs and the settings used by the subsequent solver stages. The main groups of configuration parameters are summarised in Table 5.1.

Table 5.1: Main groups of parameters stored in the VKI-SDPM configuration structure.

Configuration group	**Main contents**
Case identification	Test case, simulation class, and geometry
Mesh definition	Panel numbers and mesh-spacing laws
Flow and kinematics	$\mathbf{V}_\infty(t)$, $\mathbf{\Omega}(t)$, and reference physical values
Wake definition	Wake length, wake mode, and time-integration method
Regularisation model	Vortex-core model and core evolution
Output and storage	Save path and simulation-animation settings
Post-processing	Reference quantities and interpolation settings
Validation and plots	Data paths, quantity maps, and plot options

The time step and the total number of time steps are computed from the configuration data, depending on the simulation class:

$$\Delta t = \begin{cases} \dfrac{\Delta\psi\ \pi/180}{|\bar{\mathbf{\Omega}}|}, & \text{rotor simulation,} \\ \dfrac{c_{\text{ref}}/N_c}{V_{\text{ref}}}, & \text{wing simulation,} \end{cases}$$
$$N_t = \begin{cases} K_{\text{w}}\,\dfrac{360^\circ}{\Delta\psi}, & \text{rotor simulation,} \\ K_{\text{w}}\ N_c, & \text{wing simulation.} \end{cases} \tag{5.1}$$

Here, $\Delta\psi$ is the azimuthal increment in degrees, $\bar{\mathbf{\Omega}}$ is the mean rotational speed,

N_c is the number of chordwise panels per half-surface, c_{ref} and V_{ref} are the reference chord and velocity, and K_{w} is the prescribed wake length, expressed as a number of rotor revolutions in rotor cases or as a multiple of N_c in wing cases.

5.2.2 Overall Computational Workflow

The global execution sequence is schematically represented in Fig. 5.1.

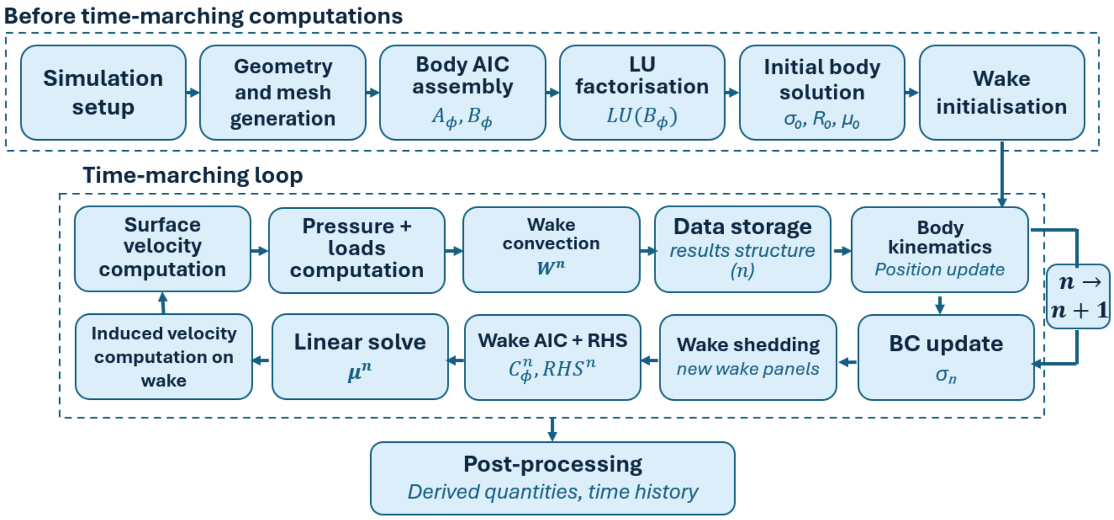


Figure 5.1: Block diagram of the VKI-SDPM workflow.

The workflow starts with the simulation setup and geometry generation. The body influence matrices $\mathbf{A}_\phi$ and $\mathbf{B}_\phi$ are then assembled. Since these matrices depend only on the body geometry for the rigid configurations considered here, they are computed before the time-marching loop, and $\mathbf{B}_\phi$ is factorised once. The body source distribution and wake state are subsequently initialised.

During the time-marching loop, the body kinematics, source vector, wake strengths, wake influence matrix, and right-hand side are updated. The body doublet distribution is obtained from the stored factorisation. The wake representation is then updated, the velocity at the wake nodes is computed according to the selected wake model, and the wake is convected. The surface velocity, pressure field, aerodynamic loads, and solution histories are evaluated or stored before advancing to the next time step. The complete sequence is collected in Section 5.6.

5.3 Geometry and Surface Discretisation

The VKI-SDPM code does not include a general-purpose CAD-like geometry generator. Instead, each validation case is associated with a dedicated geometry-generation function, which provides the upper and lower surface vertex grids and the geometric metadata required by the common assembly routines.

Consequently, adding a new test case requires writing a dedicated geometry function that follows the same output interface, while the subsequent assembly, indexing, panel construction, and diagnostic routines remain unchanged.

5.3.1 Geometry Assembly and Mesh Spacing

For rotor and propeller simulations, the reference blade geometry is generated by the selected test-case function and then replicated azimuthally around the declared rotor axis by assemble_propeller.m. For wing simulations, the semi-span geometry is converted into a complete wing by assemble_wing.m. In both cases, the final panel structure is generated by build_panel_geometry.m. The wing is therefore treated as a single component, whereas a rotor is treated as a set of components, one per blade.

The resulting panel geometry can be schematically represented as

$$\mathcal{G}_B = \left\{\mathbf{x}_{\mathrm{CP},k}, \mathbf{P}_{1,k}, \mathbf{P}_{2,k}, \mathbf{P}_{3,k}, \mathbf{P}_{4,k}, \hat{\mathbf{n}}_k, \hat{\mathbf{l}}_k, \hat{\mathbf{m}}_k, S_k, \mathcal{I}_k\right\}_{k=1}^{N_\mathrm{B}}, \tag{5.2}$$

where $\mathbf{x}_{\mathrm{CP},k}$ is the panel collocation point, $\mathbf{P}_{1,k}$–$\mathbf{P}_{4,k}$ are the panel vertices, $\hat{\mathbf{n}}_k$ is the outward normal, $\hat{\mathbf{l}}_k$ and $\hat{\mathbf{m}}_k$ are the local tangential directions, S_k is the panel area, $\mathcal{I}_k$ contains the topological indices of the panel, and N_B is the total number of body panels.

The case-specific geometry function defines the physical shape of the lifting surface, whereas the spacing laws determine how this shape is sampled in the chordwise and spanwise directions. The same spacing options are then propagated to the common panel-construction routine.

The chordwise mesh is generated from the non-dimensional coordinate $\xi_i = x_i/c$, with $i = 0, \ldots, N_c$. Three chordwise spacing laws are available:

$$\xi_i = \begin{cases} \dfrac{i}{N_c}, & \text{uniform spacing,} \\ \dfrac{1}{2}\left[1 - \cos\left(\dfrac{\pi i}{N_c}\right)\right], & \text{cosine spacing,} \\ 1 - \cos\left(\dfrac{\pi i}{2N_c}\right), & \text{sine spacing.} \end{cases} \tag{5.3}$$

The cosine law clusters nodes near both the leading edge and the trailing edge, whereas the sine law clusters nodes mainly near the leading edge and produces a coarser distribution toward the trailing edge.

For the spanwise spacing a distinction between wings and rotors is necessary. In rotor and propeller cases, the spanwise stations are defined along the radial coordinate from the inner station r_{in} to the blade tip R

$$r_j = \begin{cases} r_{\text{in}} + \dfrac{j}{N_r}\left(R - r_{\text{in}}\right), & \text{uniform spacing,} \\ r_{\text{in}} + \dfrac{1}{2}\left[1 - \cos\left(\dfrac{\pi j}{N_r}\right)\right]\left(R - r_{\text{in}}\right), & \text{cosine spacing,} \end{cases} \qquad j = 0, \ldots, N_r. \tag{5.4}$$

In wing cases, the test-case generator first defines the semi-span $b/2$, whereas the assembly routine reconstructs the complete span. After this assembly, N_r denotes the total number of spanwise panels of the complete wing and the spanwise stations are defined from the left tip to the right tip

$$\eta_j = \begin{cases} b\left(\dfrac{j}{N_r} - \dfrac{1}{2}\right), & \text{uniform spacing,} \\ -\dfrac{b}{2}\cos\left(\dfrac{\pi j}{N_r}\right), & \text{cosine spacing,} \end{cases} \qquad j = 0, \ldots, N_r. \tag{5.5}$$

The full-span cosine law clusters panels near the two wing tips.

5.3.2 Panel Topology, Vertex Ordering, and Local Frames

The topology of each panel is identified by one component label and three local indices:

$$\mathcal{I}_k = \left\{\text{body\_idx}_k,\ \text{surf}_k,\ c_i,\ s_j\right\}, \tag{5.6}$$

where body_idx_k is the body/component index, surf_k is the surface label, c_i is the contour index, and s_j is the spanwise strip index. The surface label divides the closed lifting body into three portions:

$$\text{surf}_k = \begin{cases} 1, & \text{left cap or root cap,} \\ 2, & \text{main lifting surface,} \\ 3, & \text{right cap or tip cap.} \end{cases} \tag{5.7}$$

For rotor blades, the left cap corresponds to the inner or root station and the right cap corresponds to the tip station. For assembled wings, the left cap corresponds to the negative-span tip and the right cap to the positive one. Thus, the main surface and the cap surfaces are three portions of the same closed panel topology ordered according to the following convention for each body_idx_k

$$\text{left/root cap} \rightarrow \text{main lifting surface} \rightarrow \text{right/tip cap}. \tag{5.8}$$

At each spanwise station, the lower and upper surface nodes are merged into one contour array ordered from the ventral TE to the dorsal TE

$$\mathrm{TE}_{\mathrm{ventr}} \to \mathrm{LE} \to \mathrm{TE}_{\mathrm{dors}},$$

Therefore, the chordwise index $c_i = 1, \ldots, 2N_c + 1$ and the spanwise index $s_j = 1, \ldots, N_r$. Consequently, the chordwise distinction between ventral and dorsal side can be written as

$$\begin{cases} 1 \leq c_i \leq N_c, & \text{ventral side,} \\ N_c + 1 \leq c_i \leq 2N_c + 1, & \text{dorsal side.} \end{cases} \tag{5.9}$$

Cap stations have $s_j = 0$. The cap panels are generated using the same chordwise contour structure, with a midline contour introduced as the average of corresponding vertices.

For each panel the vertex ordering is

$$\begin{cases} \mathbf{P}_{1,k} = \mathbf{x}(c_i, s_j), & \mathbf{P}_{2,k} = \mathbf{x}(c_i, s_j + 1), \\ \mathbf{P}_{3,k} = \mathbf{x}(c_i + 1, s_j + 1), & \mathbf{P}_{4,k} = \mathbf{x}(c_i + 1, s_j). \end{cases} \tag{5.10}$$

The two caps use the midline to identify the two parallel rows of panels and their orientation. However, all of the ordering rules preserve the same clockwise convention when the panel is viewed from outside the body facing the outward normal vector.

The outward normal is obtained from the diagonal cross product:

$$\hat{\mathbf{n}}_k = \frac{(\mathbf{P}_{4,k} - \mathbf{P}_{2,k}) \times (\mathbf{P}_{3,k} - \mathbf{P}_{1,k})}{|(\mathbf{P}_{4,k} - \mathbf{P}_{2,k}) \times (\mathbf{P}_{3,k} - \mathbf{P}_{1,k})|}, \qquad S_k = \frac{1}{2} \left| (\mathbf{P}_{4,k} - \mathbf{P}_{2,k}) \times (\mathbf{P}_{3,k} - \mathbf{P}_{1,k}) \right|. \tag{5.11}$$

The panel local frame is completed by the two tangent unit vectors

$$\hat{\mathbf{l}}_k = \frac{[(\mathbf{P}_{4,k} - \mathbf{P}_{1,k}) + (\mathbf{P}_{3,k} - \mathbf{P}_{2,k})]}{|(\mathbf{P}_{4,k} - \mathbf{P}_{1,k}) + (\mathbf{P}_{3,k} - \mathbf{P}_{2,k})|}, \qquad \hat{\mathbf{m}}_k = \hat{\mathbf{n}}_k \times \hat{\mathbf{l}}_k. \tag{5.12}$$

For main-surface panels, $\hat{\mathbf{l}}_k$ follows the direction of increasing contour index. It therefore points from trailing edge to leading edge on the ventral side and from leading edge to trailing edge on the dorsal side.

At the leading edge and trailing edge of the cap surfaces, the outer and midline nodes may coincide. In this case, the quadrilateral panel collapses into a triangular panel. The code detects this condition through vertex-distance checks and changes the centroid formula accordingly:

$$\mathbf{x}_{\mathrm{CP},k} = \begin{cases} (\mathbf{P}_{1,k} + \mathbf{P}_{2,k} + \mathbf{P}_{3,k} + \mathbf{P}_{4,k})/4, & \text{regular quadrilateral,} \\ (\mathbf{P}_{1,k} + \mathbf{P}_{3,k} + \mathbf{P}_{4,k})/3, & \mathbf{P}_{1,k} \equiv \mathbf{P}_{2,k}, \\ (\mathbf{P}_{1,k} + \mathbf{P}_{2,k} + \mathbf{P}_{3,k})/3, & \mathbf{P}_{3,k} \equiv \mathbf{P}_{4,k}. \end{cases} \tag{5.13}$$

The collapsed panels are therefore retained in the surface topology, but their collocation point is evaluated as the centroid of the corresponding triangular panel.

Several consistency checks are executed to prevent inconsistent panel orientations and other geometric errors that would compromise the simulation. The most relevant diagnostic is the mesh non-planarity measurement. This diagnostic does not define a hard error by itself, but it quantifies the mesh quality and supports the assessment of the subsequent computations. It is evaluated as

$$\varepsilon_{\mathrm{plan},k} = \frac{\max_{a=1,\dots,4} \left| \hat{\mathbf{n}}_k \cdot (\mathbf{P}_{a,k} - \mathbf{x}_{\mathrm{CP},k}) \right|}{\sqrt{S_k}} \cdot 100. \tag{5.14}$$

A panel is marked as planar if this value remains below the prescribed percentage tolerance. This diagnostic is especially useful for twisted blades, swept wings, and tapered geometries, where quadrilateral panels may become slightly warped.

5.4 Discrete Potential Problem

The potential problem is reduced to the algebraic system introduced in Chapter 2. The present section describes the evaluation of the influence coefficients, the assembly of the known contributions, and the solution strategy adopted in VKI-SDPM.

5.4.1 Influence Matrices and Kernel Evaluation

The aerodynamic influence-coefficient matrices are built from the same panel-influence logic: the target points are the body collocation points, whereas the singularity panels may belong either to the body surface or to the wake surface. Consequently, the body source matrix $\mathbf{A}_\phi$, the body doublet matrix $\mathbf{B}_\phi$, and the wake doublet matrix $\mathbf{C}_\phi^n$ are obtained as

$$\begin{aligned} &\mathbf{A}_\phi : \text{body source panels} \longrightarrow \text{body collocation points}, \\ &\mathbf{B}_\phi : \text{body doublet panels} \longrightarrow \text{body collocation points}, \\ &\mathbf{C}_\phi^n : \text{wake doublet panels at time step } n \longrightarrow \text{body collocation points}. \end{aligned} \tag{5.15}$$

$\mathbf{A}_\phi$ and $\mathbf{B}_\phi$ are body–body influence matrices, and they can be computed once before the time-marching loop. $\mathbf{C}_\phi^n$ is the wake–body influence matrix, therefore it must be computed at every step of the time loop since the wake evolves relative to the body.

The body source and doublet influence entries are defined as

$$A_\phi^{ji} = \phi_{\sigma,i} \left(\mathbf{x}_{\mathrm{CP},j},\ \sigma_i = 1 \right), \qquad i,j = 1,\dots,N_{\mathrm{B}}. \tag{5.16}$$

$$B_{\phi}^{ji} = \phi_{\mu,i}\left(\mathbf{x}_{\mathrm{CP},j},\ \mu_i = 1\right), \qquad i, j = 1, \ldots, N_{\mathrm{B}}. \tag{5.17}$$

Here N_{B} is the number of body panels, the row index j identifies the target body collocation point, while the column index i identifies the body singularity panel. Thus, one matrix column corresponds to the influence of one unit-strength panel on all body collocation points.

The same convention is used for the wake doublet influence matrix. If q denotes a wake panel and N_{w}^{n} is the number of wake panels active at time step n, the wake influence entries are written as

$$C_{\phi}^{jq,n} = \phi_{\mu_{\mathrm{w}},q}\left(\mathbf{x}_{\mathrm{CP},j}; \mu_{\mathrm{w},q} = 1\right), \qquad j = 1, \ldots, N_{\mathrm{B}}, \qquad q = 1, \ldots, N_{\mathrm{w}}^{n}. \tag{5.18}$$

In the code implementation, the body matrices are assembled by the vectorised routines compute_A_phi_LS_vectorial.m and compute_B_phi_LS_vectorial.m, whereas the wake influence matrix is assembled by compute_C_phi_vectorial.m. The analytical source-potential kernel is evaluated following Eq. (2.107) by influence_potential_source_vectorial.m, while influence_potential_doublet_vectorial.m is used for both body and wake doublet panels following Eq. (2.108).

The assembly is organised column-wise. For a generic singularity panel i, the code first extracts the panel centroid, vertices, unit normal, and local tangential directions. All body collocation points are then expressed in the local reference frame of panel i via the local transformation matrix

$$\tilde{\mathbf{T}}_i = \begin{bmatrix} \hat{\mathbf{l}}_i^{T} \\ \hat{\mathbf{m}}_i^{T} \\ \hat{\mathbf{n}}_i^{T} \end{bmatrix}. \tag{5.19}$$

The local target coordinates are therefore obtained as

$$\mathbf{x}_{\mathrm{CP},j}^{(loc)} = \tilde{\mathbf{T}}_i\left(\mathbf{x}_{\mathrm{CP},j} - \mathbf{x}_{\mathrm{CP},i}\right), \qquad j = 1, \ldots, N_{\mathrm{B}}. \tag{5.20}$$

The vertices of the singularity panel are also expressed in the same local frame. The analytical source or doublet kernel is then evaluated on the whole vector of target points.

For non-planar quadrilateral panels, the influence coefficients are evaluated on a local mean panel plane, as schematically shown in Fig. 5.2. This approximation makes the analytical panel formulas compatible with the slightly non-planar quadrilateral elements that may arise from twisted, tapered, or swept surfaces. For wake panels, the effect of this approximation is expected to be less critical as the panels are convected downstream, although close and strongly rolled-up wake regions may require attention.

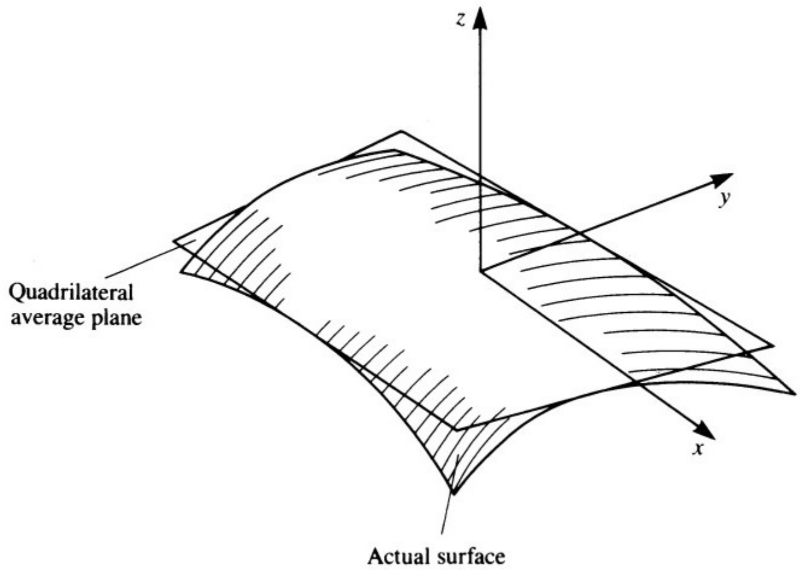


Figure 5.2: Example of mean-panel-plane approximation for warped panels [5].

5.4.2 RHS Assembly and Linear-System Solution

The known terms of the algebraic system form the RHS. The first contribution is the source term, which is evaluated by compute_sources.m. For each body panel k, the source strength is obtained from the local non-penetration condition, which can be written, according to the adopted normal convention, as

$$\sigma_k^n = -\hat{\mathbf{n}}_k^n \cdot \mathbf{V}_{\mathrm{kin},k}^n. \tag{5.21}$$

The kinematic velocity $\mathbf{V}_{\mathrm{kin},k}^n$ is evaluated at the panel collocation point as

$$\mathbf{V}_{\mathrm{kin},k}^n = \mathbf{V}_\infty^n - \boldsymbol{\Omega}^n \times \mathbf{x}_{\mathrm{CP},k}^n. \tag{5.22}$$

The second known contribution is the wake-induced potential term. At time step n, the current wake state provides both $\mathbf{C}_\phi^n$ and $\boldsymbol{\mu}_\mathrm{w}^n$. The wake strengths are assigned from the trailing-edge doublet difference and retained through the wake history, as described in Section 5.5.

The right-hand side is therefore

$$\mathbf{RHS}^n = -\mathbf{A}_\phi \boldsymbol{\sigma}^n - \mathbf{C}_\phi^n \boldsymbol{\mu}_\mathrm{w}^n. \tag{5.23}$$

At the initial step, the wake structure does not yet contain any wake panels. Therefore, the right-hand side reduces to the source term only

$$\mathbf{RHS}^0 = -\mathbf{A}_\phi \boldsymbol{\sigma}^0. \tag{5.24}$$

Once the components of the algebraic system are assembled, the solution is reduced to the computation of the body doublet vector at the n-th time step

$$\mathbf{B}_\phi \boldsymbol{\mu}^n = \mathbf{RHS}^n \tag{5.25}$$

The current implementation takes advantage of the fact that for rigid-body configurations, the body doublet matrix $\mathbf{B}_\phi$ is not rebuilt during the time-marching loop. Therefore, after the body doublet matrix has been assembled, the Lower-Upper (LU) factorisation has to be done only once

$$\mathbf{P}^\star \mathbf{B}_\phi = \mathbf{L}_\star \mathbf{U}_\star. \tag{5.26}$$

The matrices $\mathbf{L}^\star$, $\mathbf{U}^\star$, and $\mathbf{P}^\star$ are stored and reused for all subsequent body solves.

Therefore, for a generic right-hand side $\mathbf{RHS}^n$, the solution is obtained in two steps:

$$\mathbf{L}^\star \mathbf{y}^n = \mathbf{P}^\star \mathbf{RHS}^n \quad \rightarrow \quad \mathbf{U}^\star \boldsymbol{\mu}^n = \mathbf{y}^n. \tag{5.27}$$

This architecture reduces the computations repeated at each time step and allows affordable time marching for the rigid-body configurations considered in this work.

5.5 Wake Representation and Evolution

The wake model implemented in VKI-SDPM is based on an explicit shedding procedure from the trailing edge of the lifting surfaces. The theoretical role of the wake doublet sheet and its connection with the Kutta condition have already been introduced in Chapter 2. The present section focuses on the wake attachment, data structure, equivalent vortex-segment representation, convection models, and regularisation used for induced-velocity evaluation.

5.5.1 Wake Structure

The routine `initialize_wake.m` searches for the dorsal trailing-edge nodes in order to build the first row of wake nodes. These are identified as

$$\mathbf{P}_{\mathrm{TE}} = \{\mathrm{surf}_k = 2, \quad c_i = 2N_c + 1, \quad s_j = 1, \dots, N_r\}. \tag{5.28}$$

With the mask in Eq. (5.28), cap panels are excluded from wake shedding even though they belong to a lifting body.

In the wing case the shedding body is the wing itself, while for the rotor case there are as many shedding bodies as the number of blades. Therefore, a separate wake state is created for each component.

For each wake component, the wake grid has a row index i and a spanwise index j. The newest wake-node row is always stored at $i = 1$, close to the trailing edge, whereas the oldest row has the largest active value of i. The wake panels retain the same vertex-ordering convention as the body panels and therefore use the same local-frame construction and surface-related quantities.

The main quantities stored in the wake structure are represented similarly to those of the body-panel structure

$$\mathcal{W}_b = \left\{\mathbf{P}_{\mathrm{w},b}, \boldsymbol{\mu}_{\mathrm{w},b}, \hat{\mathbf{n}}_{\mathrm{w},b}, \hat{\mathbf{l}}_{\mathrm{w},b}, \hat{\mathbf{m}}_{\mathrm{w},b}, \mathbf{V}^{\mathrm{prev}}_{\mathrm{w},b}\right\}, \tag{5.29}$$

where the subscript b is the shedding component index. Here, $\mathbf{P}_{\mathrm{w},b}$ contains the wake-panel coordinates, $\boldsymbol{\mu}_{\mathrm{w},b}$ contains the wake-panel doublet strengths, $\hat{\mathbf{n}}_{\mathrm{w},b}$, $\hat{\mathbf{l}}_{\mathrm{w},b}$, and $\hat{\mathbf{m}}_{\mathrm{w},b}$ are the local wake-panel-frame unit vectors, and $\mathbf{V}^{\mathrm{prev}}_{\mathrm{w},b}$ stores the previous induced velocity field.

New wake panels are generated by create_wake_panels_exp.m. The indices of the previously existing wake panels are shifted from i to $i+1$, leaving the row $i = 1$ available for the newly shed trailing-edge nodes. The new row is obtained from the nodes selected by Eq. (5.28) and connected to the previous wake row.

The doublet strength assigned to the new wake row is obtained from the dorsal and ventral trailing-edge panels at the previous time step [5]:

$$\boldsymbol{\mu}^{n}_{\mathrm{w},1\mathrm{j}} = \mu^{n-1}_{2N_c+1,j} - \mu^{n-1}_{1,j}, \qquad j = 1, \ldots, N_r. \tag{5.30}$$

5.5.2 Prescribed- and Free-Wake Convection

The wake-convection model is selected in the simulation configuration through the field cfg.solver.wake_mode. Two options are implemented:

$$\mathsf{wake\_mode} \in \{\mathsf{prescribed}, \mathsf{free}\}. \tag{5.31}$$

In the prescribed-wake mode, the wake nodes are convected only by the freestream velocity. The wake-node update is therefore

$$\mathbf{x}^{n+1}_{\mathrm{w},i,j} = \mathbf{x}^{n}_{\mathrm{w},i,j} + \mathbf{V}^{n}_{\infty}\Delta t. \tag{5.32}$$

In the free-wake mode, the wake-node velocity contains both the freestream contribution and the induced velocity generated by the current body and wake singularities. The induced component is computed by compute_on_wake_velocities_vectorial.m.

The induced velocity used by the free-wake model includes three contributions:

$$\mathbf{V}^{n}_{\mathrm{ind,w}} = \mathbf{V}^{n}_{\sigma} + \mathbf{V}^{n}_{\mu} + \mathbf{V}^{n}_{\mu_{\mathrm{w}}} \tag{5.33}$$

The body-source contribution is evaluated by velocity_from_source_vectorial.m using the analytical expression for the source velocity that coincides with the potential doublet kernel of Eq. (2.108).

The body- and wake-doublet contributions to the induced velocity are evaluated through the constant doublet-panel vortex-ring analogy [5], for which a derivation is reported in App. A.3. According to this equivalence, a constant-strength doublet panel induces the same velocity field as a vortex ring distributed along its boundary, with

$$\mu = \Gamma. \tag{5.34}$$

This representation allows the induced velocity to be evaluated through the Biot–Savart law without requiring the analytical velocity formulation of a planar doublet panel. For this purpose, an additional data structure based on vortex filaments is implemented alongside the doublet-panel representation. This structure facilitates the evaluation of local wake properties and avoids repeated calculations on edges shared by adjacent panels. In particular, coincident segments are identified and their circulation strengths are summed algebraically, so that only the resulting net circulation is retained.

The resulting segment data are stored using a structure-of-arrays layout and include the segment endpoints, circulation strengths, and the variables required by the adopted Biot–Savart kernel regularisation. The induced velocity is evaluated by the routine `velocity_vortex_segment_vectorial.m`.

Once the induced velocity field has been evaluated, two time-integration options are available for the free-wake convection. The first option is the forward-Euler scheme

$$\mathbf{x}^{n+1}_{\mathrm{w},i,j} = \mathbf{x}^{n}_{\mathrm{w},i,j} + \left(\mathbf{V}^{n}_{\infty} + \mathbf{V}^{n}_{\mathrm{ind,w},i,j}\right)\Delta t. \tag{5.35}$$

The second one is the Euler–trapezoidal-type option

$$\mathbf{x}^{n+1}_{\mathrm{w},i,j} = \mathbf{x}^{n}_{\mathrm{w},i,j} + \frac{\Delta t}{2}\left(2\mathbf{V}^{n}_{\infty} + \mathbf{V}^{n}_{\mathrm{ind,w},i,j} + \mathbf{V}^{n-1}_{\mathrm{ind,w},i,j}\right). \tag{5.36}$$

At the first time step, the trapezoidal option falls back to Euler forward because no previous induced-velocity field is available [43]. After each convection step, `convect_wake.m` updates the wake-node coordinates and recomputes the wake-panel local frames.

5.5.3 Vortex-Core Regularisation and Core Growth

The vortex-segment velocity kernel requires the application of a regularisation factor, as introduced in Subsec. 2.2.3. The implemented options are

$$F = \begin{cases} 1, & \textsf{none}, \\ 1 - \exp\left[-\left(\dfrac{d}{r_c}\right)^2\right], & \textsf{lamb\_oseen}, \\ \dfrac{d^2}{\sqrt{d^4 + r_c^4}}, & \textsf{vatistas}. \end{cases} \tag{5.37}$$

The option none leaves the Biot–Savart expression unregularised, except for the machine-precision numerical cutoff inside the kernel. The options lamb_oseen and vatistas damp the induced velocity near the segment core. In free-wake simulations, this regularisation is important to avoid unrealistically large velocities when wake nodes approach vortex segments during wake roll-up.

Different core-growth models have been implemented following [18] and are applied by the routine compute_vortex_segment_core_radius.m.

The available options are

$$r_c = \begin{cases} r_{c,0}, & \textsf{constant}, \\ \dfrac{r_{c,0}}{\sqrt{1+\varepsilon}}, & \textsf{stretching}, \\ \sqrt{r_{c,0}^2 + 4\alpha_{\mathrm{Oseen}}\delta\nu\zeta}, & \textsf{wake\_age}, \\ \dfrac{\sqrt{r_{c,0}^2 + 4\alpha_{\mathrm{Oseen}}\delta\nu\zeta}}{\sqrt{1+\varepsilon}}, & \textsf{stretching\_age}. \end{cases} \tag{5.38}$$

Here, $r_{c,0}$ is the initial core radius, ε is the non-dimensional vortex strain computed from the change in segment length between two consecutive time steps, ζ is the segment age in physical time, ν is the kinematic viscosity, and δ is the diffusion parameter.

The model for the diffusion parameter can be chosen from the following options

$$\textsf{turbulent\_viscosity\_model} \in \{\textsf{constant}, \textsf{squire}, \textsf{stepped}\}\,. \tag{5.39}$$

The constant option keeps δ fixed; the squire option updates it based on the segment vortex Reynolds number, following [27] as in Eq. 2.58; stepped implements implements a step-law variation of δ that can be controlled by the number of steps and the variation between consecutive steps.

5.6 Time-Marching Procedure

The preceding sections describe the individual modules of VKI-SDPM. Their interaction within a complete simulation is summarised here. Before entering the time loop, the code generates the body geometry, assembles $\mathbf{A}_\phi$ and $\mathbf{B}_\phi$, factorises $\mathbf{B}_\phi$, and initialises the wake and the required history arrays.

At a generic time step n, the sequence can be summarised as follows:

Algorithm 1 VKI-SDPM time-marching procedure

1: Update the prescribed body kinematics and compute $\boldsymbol{\sigma}^n$
2: Update the active wake topology and assign $\boldsymbol{\mu}_{\mathrm{w}}^n$ from the previous trailing-edge solution
3: Assemble $\mathbf{C}_\phi^n$ and $\mathbf{RHS}^n$
4: Solve $\mathbf{B}_\phi \boldsymbol{\mu}^n = \mathbf{RHS}^n$ using the stored LU factors
5: Update the equivalent wake vortex segments and their regularisation variables
6: Evaluate the wake-node velocity according to the prescribed- or free-wake model
7: Convect the wake
8: Reconstruct the surface perturbation velocity, evaluate pressure and aerodynamic loads, and update the stored solution histories

At the initial step, no wake panels are active and the right-hand side is given by Eq. (5.24). From the subsequent steps onward, the wake contribution is included through Eq. (5.23). The body matrices and their LU factorisation remain unchanged throughout the simulation, whereas $\boldsymbol{\sigma}^n$, $\boldsymbol{\mu}_{\mathrm{w}}^n$, $\mathbf{C}_\phi^n$, and the wake geometry are updated at every time step.

5.7 Surface Velocity, Pressure, and Load Evaluation

After the body doublet system has been solved, VKI-SDPM reconstructs the surface perturbation velocity, evaluates the pressure field, and integrates the aerodynamic loads. These operations are performed inside the time loop after the current body doublet vector $\boldsymbol{\mu}^n$ and source vector $\boldsymbol{\sigma}^n$ are available. The corresponding quantities are then stored in the results structure and used by the post-processing routines.

5.7.1 Surface Perturbation Velocity Reconstruction

The surface perturbation velocity is reconstructed by `compute_surface_velocity.m`. The routine receives the panel structure, the current body doublet vector $\boldsymbol{\mu}^n$, and the current source vector $\boldsymbol{\sigma}^n$.

The reconstruction is performed from the surface gradients of the doublet strength and from the source strength [5]. For a generic panel k, the local velocity components are written as

$$u_k^n = \left.\frac{\partial \mu^n}{\partial l}\right|_k, \qquad v_k^n = \left.\frac{\partial \mu^n}{\partial m}\right|_k, \qquad w_k^n = \sigma_k^n. \tag{5.40}$$

Here, l and m denote the two local tangential directions. The normal component is given directly by the source strength, consistently with the imposed non-penetration condition.

The derivatives of μ are evaluated using finite differences. Since the panel spacing may be non-uniform in both chordwise and spanwise directions, the distance between collocation points is used explicitly in the finite-difference formulas. For an interior point, the centred non-uniform approximation used is

$$\left.\frac{\partial \mu}{\partial s}\right|_0 \simeq \frac{h_-^2 \mu_+ - h_+^2 \mu_- + \left(h_+^2 - h_-^2\right)\mu_0}{h_+ h_- \left(h_+ + h_-\right)}, \tag{5.41}$$

where s represents either the chordwise or spanwise surface direction. The quantities h_+ and h_- are the distances from the current collocation point to the forward and backward neighbouring collocation points, respectively.

At boundaries, one-sided formulas are used. The second-order forward approximation implemented is

$$\left.\frac{\partial \mu}{\partial s}\right|_0 \simeq \frac{-\mu_0 \left(h_2^2 - h_1^2\right) + h_2^2 \mu_1 - h_1^2 \mu_2}{h_1 h_2 \left(h_2 - h_1\right)}, \tag{5.42}$$

whereas the corresponding backward approximation is

$$\left.\frac{\partial \mu}{\partial s}\right|_0 \simeq \frac{\mu_0 \left(h_2^2 - h_1^2\right) - h_2^2 \mu_1 + h_1^2 \mu_2}{h_1 h_2 \left(h_2 - h_1\right)}. \tag{5.43}$$

Once the local components have been computed, they are transformed into the global frame. For cap panels, the current implementation keeps the original reconstruction based on the local panel frame stored in the geometry structure. The perturbation velocity is therefore obtained as

$$\mathbf{v}_{\mathrm{surf},k}^n = u_k^n \hat{\mathbf{l}}_k + v_k^n \hat{\mathbf{m}}_k + w_k^n \hat{\mathbf{n}}_k. \tag{5.44}$$

For the main lifting surface, the code uses a metric-aware transformation [40], [53]. The reason is that the chordwise and spanwise directions reconstructed from neighbouring collocation points are not necessarily orthogonal, as in the case of twisted, tapered, or swept geometries. Let $\hat{\mathbf{t}}_{l,k}$ and $\hat{\mathbf{t}}_{m,k}$ be the chordwise and spanwise unit tangents reconstructed from the neighbouring collocation points of panel k. In compact form,

$$\hat{\mathbf{t}}_{l,k} = \frac{\Delta \mathbf{x}_{l,k}}{\left|\Delta \mathbf{x}_{l,k}\right|}, \qquad \hat{\mathbf{t}}_{m,k} = \frac{\Delta \mathbf{x}_{m,k}}{\left|\Delta \mathbf{x}_{m,k}\right|}. \tag{5.45}$$

The increments $\Delta\mathbf{x}_{l,k}$ and $\Delta\mathbf{x}_{m,k}$ are obtained consistently with the finite-difference stencil used for the doublet gradients.

The global perturbation velocity is then reconstructed by imposing the three local projection conditions

$$\mathbf{v}^n_{\mathrm{surf},k} \cdot \hat{\mathbf{t}}_{l,k} = u^n_k, \qquad \mathbf{v}^n_{\mathrm{surf},k} \cdot \hat{\mathbf{t}}_{m,k} = v^n_k, \qquad \mathbf{v}^n_{\mathrm{surf},k} \cdot \hat{\mathbf{n}}_k = w^n_k. \tag{5.46}$$

These conditions lead to the local linear system

$$\tilde{\mathbf{G}}_k \mathbf{v}^n_{\mathrm{surf},k} = \mathbf{v}^n_k, \tag{5.47}$$

where

$$\tilde{\mathbf{G}}_k = \begin{bmatrix} \hat{\mathbf{t}}^T_{l,k} \\ \hat{\mathbf{t}}^T_{m,k} \\ \hat{\mathbf{n}}^T_k \end{bmatrix}, \qquad \mathbf{v}^n_k = \begin{bmatrix} u^n_k \\ v^n_k \\ w^n_k \end{bmatrix}. \tag{5.48}$$

The global surface perturbation velocity is therefore obtained as

$$\mathbf{v}^n_{\mathrm{surf},k} = \tilde{\mathbf{G}}_k^{-1} \mathbf{v}^n_k. \tag{5.49}$$

In the implementation, the determinant of the local metric matrix is checked before applying the transformation.

5.7.2 Pressure Evaluation and Aerodynamic Loads

The dimensional pressure is evaluated by \textsf {compute_pressure.m} using the unsteady Bernoulli equation. At each body panel k, the reference kinematic velocity is computed as

$$\mathbf{V}^n_{\mathrm{ref},k} = \mathbf{V}^n_\infty - \boldsymbol{\Omega}^n \times \mathbf{x}^n_{\mathrm{CP},k}. \tag{5.50}$$

The total velocity used in the pressure equation is then

$$\mathbf{V}^n_k = \mathbf{V}^n_{\mathrm{ref},k} + \mathbf{v}^n_{\mathrm{surf},k}. \tag{5.51}$$

The unsteady contribution is obtained from the time derivative of the body doublet strength. In the code, the history arrays $\boldsymbol{\mu}^n$, $\boldsymbol{\mu}^{n-1}$, and $\boldsymbol{\mu}^{n-2}$ are passed to compute_pressure.m. A second-order scheme is implemented to compute the time derivative at time step n

$$\left.\frac{\partial \phi}{\partial t}\right|^n_k = \left.\frac{\partial \mu}{\partial t}\right|^n_k = \begin{cases} 0, & \text{if } n = 1, \\ \dfrac{\mu^n_k - \mu^{n-1}_k}{\Delta t}, & \text{if } n = 2, \\ \dfrac{3\mu^n_k - 4\mu^{n-1}_k + \mu^{n-2}_k}{2\Delta t}, & \text{if } n \geq 3. \end{cases} \tag{5.52}$$

The pressure is then computed from the unsteady Bernoulli equation in the form

$$p_k^n = p_{\mathrm{ref}} + \frac{1}{2}\rho\left(\left|\mathbf{V}^n_{\mathrm{ref},k}\right|^2 - |\mathbf{V}_k^n|^2\right) - \rho \left.\frac{\partial \phi}{\partial t}\right|_k^n . \tag{5.53}$$

The output is the dimensional pressure vector $\mathbf{p}^n_{\mathrm{surf}}$. During the simulation, the pressure at each time step is stored in order to reconstruct its complete time history.

The pressure coefficient is evaluated by compute_Cp.m. The routine reads the dimensional pressure history stored in results.p_surf and converts it into

$$C^n_{p,k} = \frac{p_k^n - p_{\mathrm{ref}}}{q_{\mathrm{ref}}}. \tag{5.54}$$

The value of q_{ref} depends on the simulation class. For rotor cases, the default reference dynamic pressure is based on the local rotational speed:

$$q_{\mathrm{ref}} = \frac{1}{2}\rho\,(\Omega r)^2 . \tag{5.55}$$

For wing cases, the reference dynamic pressure is computed from the freestream velocity by default.

5.7.3 Force and Moment Integration

The aerodynamic loads are evaluated by compute_forces_moments.m. This routine is called after C_p has been computed.

For a body panel k, the elementary pressure force is computed as

$$\mathbf{F}_k^n = -C^n_{p,k} q_{\mathrm{ref}} S_k \hat{\mathbf{n}}_k, \tag{5.56}$$

where S_k is the panel area and $\hat{\mathbf{n}}_k$ is the outward panel normal in the global frame. The moment contribution is evaluated with respect to the user-specified reference point as

$$\mathbf{M}_k^n = (\mathbf{x}_{\mathrm{CP},k} - \mathbf{x}_{\mathrm{ref}}) \times \mathbf{F}_k^n. \tag{5.57}$$

If no reference point is provided, the routine uses the origin as a fallback.

The total force and moment are obtained by summing the panel contributions over each body and then over all bodies at each time step

$$\mathbf{F}_B^n = \sum_{k=1}^{N_{\mathrm{B}}} \mathbf{F}_k^n, \qquad \mathbf{M}_B^n = \sum_{k=1}^{N_{\mathrm{B}}} \mathbf{M}_k^n, \tag{5.58}$$

and

$$\mathbf{F}_{\mathrm{tot}}^n = \sum_{B=1}^{N_{\mathrm{body}}} \mathbf{F}_B^n, \qquad \mathbf{M}_{\mathrm{tot}}^n = \sum_{B=1}^{N_{\mathrm{body}}} \mathbf{M}_B^n. \tag{5.59}$$

The corresponding dimensional and non-dimensional quantities are stored in the loads structure, both as body-wise contributions and as total loads.

The force and moment coefficients are computed by default as

$$\mathbf{C}_F^n = \frac{\mathbf{F}_{\mathrm{tot}}^n}{F_{\mathrm{scale}}}, \qquad \mathbf{C}_M^n = \frac{\mathbf{M}_{\mathrm{tot}}^n}{M_{\mathrm{scale}}}. \tag{5.60}$$

where F_{scale} and M_{scale} are user-defined normalising quantities. By default, they are set to

$$\begin{aligned} F_{\mathrm{scale}} &= \begin{cases} \pi \rho V_{\mathrm{tip}}^2 R^2, & \text{rotor case,} \\ q_{\mathrm{ref}} S_{\mathrm{ref}}, & \text{wing case,} \end{cases} \\ M_{\mathrm{scale}} &= \begin{cases} \pi \rho V_{\mathrm{tip}}^2 R^3, & \text{rotor case,} \\ q_{\mathrm{ref}} S_{\mathrm{ref}} L_{\mathrm{ref}}, & \text{wing case.} \end{cases} \end{aligned} \tag{5.61}$$

Wing and rotor reference quantities can be defined by the user according to the needs of the simulation.

Rotor quantities. Rotor thrust and torque components are obtained by projection onto the declared rotor axis $\hat{\mathbf{e}}_{\mathrm{rot}}$:

$$T^n = \mathbf{F}_{\mathrm{tot}}^n \cdot \hat{\mathbf{e}}_{\mathrm{rot}}, \qquad Q^n = \mathbf{M}_{\mathrm{tot}}^n \cdot \hat{\mathbf{e}}_{\mathrm{rot}}, \tag{5.62}$$

Wing quantities. For wing simulations, lift and drag are obtained by projecting the total force onto the wind-frame directions

$$\hat{\mathbf{e}}_D = \frac{\mathbf{V}_\infty}{|\mathbf{V}_\infty|}, \qquad \hat{\mathbf{e}}_L = \frac{\hat{\mathbf{e}}_D \times \hat{\mathbf{e}}_y}{|\hat{\mathbf{e}}_D \times \hat{\mathbf{e}}_y|}. \tag{5.63}$$

The stored wing aliases are

$$L^n = \mathbf{F}_{\mathrm{tot}}^n \cdot \hat{\mathbf{e}}_L, \qquad D^n = \mathbf{F}_{\mathrm{tot}}^n \cdot \hat{\mathbf{e}}_D, \tag{5.64}$$

and

$$C_L^n = \frac{L^n}{F_{\mathrm{scale}}}, \qquad C_D^n = \frac{D^n}{F_{\mathrm{scale}}}. \tag{5.65}$$

Chapter 6

VKI-SDPM: Sensitivity Analysis

6.1 Purpose of the Sensitivity Analysis

The formulation and implementation of VKI-SDPM have been presented in the previous chapter. Before using the solver for the validation against the experimental data, it is necessary to assess whether the numerical solution is sufficiently insensitive to the main discretisation and modelling parameters adopted in the computations.

The objective of the present chapter is therefore to define a consistent set of parameters for the validation campaign of VKI-SDPM. The sensitivity analysis is carried out for both wing and rotor configurations, so that the main numerical parameters are assessed in the two classes of applications considered in this work.

For the wing simulations, the reference configuration is the rectangular straight wing from the Yip–Shubert test case [9]. For the rotor simulations, the reference configuration is taken from the NREL UAE Phase VI wind-turbine rotor test case [11]. These two cases are used here only as numerical benchmarks for the sensitivity campaign, while the detailed description of their geometry, operating conditions, and experimental data is provided in the corresponding validation sections of Ch. 7.

For both cases, the sensitivity analysis is organised around the main numerical parameters that can affect the computed loads and pressure distributions. First, the influence of the body-surface discretisation is examined by varying the chordwise resolution and the spanwise resolution of the mesh. In this way, the dependence of the solution on the number and distribution of panels can be quantified separately for wing and rotor simulations.

A separate part of the analysis treats the wake representation. Once a suitable body-surface discretisation has been selected, the sensitivity to wake-related parameters is investigated by considering the wake length, the prescribed- and free-wake formulations, and the vortex-core regularisation.

A final discussion of CPU time and parallel speed-up is included to quantify the computational performance of the solver and to estimate the parallel efficiency of the implementation.

The final outcome of the chapter is a set of numerical parameters retained for the validation cases. These parameters are not intended to be universally optimal for every possible configuration, but rather to provide a compromise between accuracy, numerical robustness, and computational cost for the test cases considered in this work.

6.1.1 Internal Numerical Verification Checks

Before performing the sensitivity analysis, a set of internal numerical checks is introduced to assess the consistency of the VKI-SDPM implementation. These checks are intended to verify that the discretised problem is assembled and solved in a numerically consistent way.

The first verification concerns the solution of the algebraic system. At each time step, once the source contribution and wake influence have been assembled, the unknown doublet-strength vector are obtained by solving the linear system associated with the discrete formulation. The residual of this system is monitored as

$$\varepsilon_{\mathrm{LS}}^{n} = \frac{|\mathbf{B}_{\phi}\boldsymbol{\mu}^{n} - \mathbf{RHS}^{n}|_{\infty}}{|\mathbf{RHS}^{n}|_{\infty}}, \tag{6.1}$$

where $\mathbf{B}_{\phi}$ is the body doublet influence matrix, while $\boldsymbol{\mu}^{n}$ and $\mathbf{RHS}^{n}$ are the unknown doublet strengths vector and the right-hand side vector at the n-th time step, respectively.

A further verification is based on the impermeability condition on the body surface. At each time step, after the solution has been obtained, the total velocity at each control point is reconstructed and projected onto the corresponding outward normal vector. The maximum residual of the non-penetration condition is evaluated as

$$\varepsilon_{\mathrm{BC}}^{n} = \frac{\max_{i} |\mathbf{V}_{i}^{n} \cdot \mathbf{n}_{i}|}{|\mathbf{V}_{\mathrm{ref}}|}, \tag{6.2}$$

where $\mathbf{V}_{i}^{n}$ is the total velocity at the i-th control point at time level n, $\mathbf{n}_{i}$ is the local unit normal, and $\mathbf{V}_{\mathrm{ref}}$ is the reference kinematic velocity.

Table 6.1 reports the values of the verification indicators for four representative cases considered in the sensitivity campaign. The reported values correspond to the maximum value of each indicator over the whole simulation. All residuals remain of the order of 10^{-14}, indicating that the algebraic system is correctly solved and that the impermeability condition is satisfied up to numerical precision. Therefore, these residuals are sufficiently small not to affect the sensitivity analysis.

Table 6.1: Representative internal verification indicators for VKI-SDPM.

Case	Wake mode	$\max_n [\varepsilon^n_{\mathrm{LS}}]$	$\max_n [\varepsilon^n_{\mathrm{BC}}]$
Wing	prescribed	$1.58481 \cdot 10^{-14}$	$1.1546 \cdot 10^{-14}$
Wing	free	$1.4239 \cdot 10^{-14}$	$1.4211 \cdot 10^{-14}$
Rotor	prescribed	$2.8422 \cdot 10^{-14}$	$2.4860 \cdot 10^{-14}$
Rotor	free	$3.1974 \cdot 10^{-14}$	$2.5315 \cdot 10^{-14}$

6.2 Surface Discretisation Sensitivity

The first sensitivity analysis concerns the discretisation of the body surface. The numerical solution depends on the resolution of the surface mesh used to describe the geometry, impose the boundary condition, and reconstruct the pressure field and aerodynamic loads.

The objective of this section is to quantify the influence of the body-surface discretisation on the main aerodynamic quantities used in the validation campaign. The chordwise and spanwise resolutions are varied separately in order to isolate their individual effects. A final combined refinement is then chosen for the subsequent validation campaign.

In order to isolate the effect of the body-surface discretisation, the wake parameters are kept fixed throughout this part of the analysis. The wake length is set to ten reference chords for the wing case and seven revolutions for the rotor case. For the rotor case with an azimuthal step of $\Delta\psi = 15°$. Moreover, all cases are run using the prescribed-wake formulation to exclude any regularisation effect on the results. All results reported in this section are evaluated after the loads have reached a steady value.

For each discretisation level, the solution is compared in terms of aerodynamic integral coefficients. For wing cases, the monitored quantity is the lift coefficient C_L, while for rotor cases it is the thrust coefficient C_T. The percentage variation is computed between two consecutive refinement levels. For a generic coefficient $C_\star$, it is defined as

$$\Delta C_\star = \frac{\left|C_\star^k - C_\star^{k-1}\right|}{\left|C_\star^{k-1}\right|} \cdot 100, \tag{6.3}$$

where the index k denotes the current refinement level.

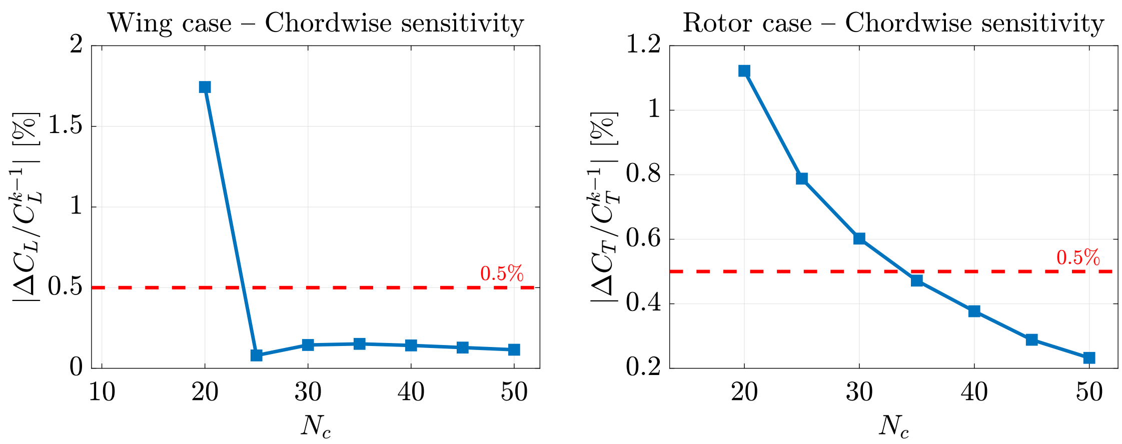
6.2.1 Chordwise discretisation

The chordwise discretisation controls the resolution of the airfoil section and directly affects the reconstruction of the surface velocity and pressure coefficient.

In this analysis, the number of chordwise panels is varied while keeping the spanwise resolution fixed at $N_r = 20$ panels, for both wing and rotor configurations.

Figure 6.1 shows the percentage variation of the aerodynamic coefficients with chordwise mesh refinement, where N_c denotes the number of panels on each half of the airfoil section.

Based on these results, the chordwise resolution retained for both the wing and rotor cases is set to $N_c = 40$ panels for both cases. For the wing case, at this resolution the consecutive variation remains below the selected threshold of 0.5%, with a stable trend. For the rotor case, the selected resolution satisfies the threshold with sufficient margin, making the choice reliable for the following simulations.



(a) Wing case chordwise sensitivity **(b)** Rotor case chordwise sensitivity

Figure 6.1: VKI-SDPM chordwise sensitivity

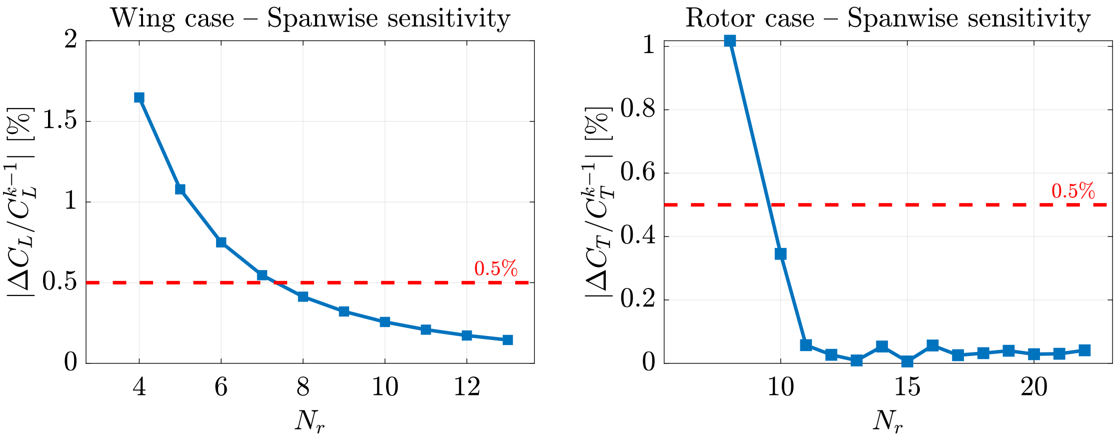
6.2.2 Spanwise discretisation

The second refinement study concerns the discretisation along the lifting direction. For wing simulations, this corresponds to the spanwise discretisation, while for rotor simulations it corresponds to the radial discretisation of each blade.

In this study, the spanwise resolution is varied while keeping the chordwise discretisation fixed at $N_c = 40$, the value chosen from the previous analysis.

Figure 6.2 illustrates the percentage variation of the aerodynamic coefficients as the spanwise discretisation is varied.

The selected spanwise resolution is $N_r = 10$ for the wing case and $N_r = 20$ for the rotor case. For the wing case, the selected value remains sufficiently below the prescribed threshold, providing a margin for the subsequent simulations. For the rotor case, the selected value provides a conservative compromise, since the coefficient variation remains below the threshold while avoiding numerical oscillations.



(a) Wing spanwise sensitivity. **(b)** Rotor spanwise sensitivity

Figure 6.2: VKI-SDPM spanwise sensitivity

6.2.3 Final Mesh Selection

After the separate chordwise and spanwise studies, a combined grid refinement is performed to verify that the selected mesh remains adequate when both discretisation directions are considered together. This step is necessary because the effects of chordwise and spanwise refinement are not completely independent: the pressure reconstruction, load integration, and wake release depend on the complete surface mesh.

The combined refinement levels are summarised in Tables 6.2 and 6.3. For each grid, the number of surface panels, the monitored aerodynamic coefficients, and the CPU time are reported. Their relative differences are computed as follows

$$\Delta C_\star = \frac{C_\star^{\text{finer}} - C_\star^{\text{baseline}}}{C_\star^{\text{baseline}}} \cdot 100, \qquad \Delta t_\star = \frac{t_\star^{\text{finer}} - t_\star^{\text{baseline}}}{t_\star^{\text{baseline}}} \cdot 100. \tag{6.4}$$

From Tables 6.2 and 6.3, the baseline meshes selected from the separate chordwise and spanwise analyses provide a reasonable compromise between accuracy and computational cost. Further mesh refinement produces limited variations in the monitored coefficients, while the CPU time increases substantially. Therefore, the baseline meshes are retained for the validation campaign, with the understanding that the remaining mesh-related uncertainty is small compared with the modelling and experimental differences expected from the validation.

Table 6.2: Combined surface-grid refinement for the wing case in VKI-SDPM.

Grid	N_c	N_r	Surface panels	C_L	ΔC_L [%]	CPU time [s]	Δt [%]
Baseline	40	10	1600	0.4641	/	98.1	/
Refined	60	15	3600	0.4626	−0.32	490.9	+400.4
Fine	80	20	6400	0.4623	−0.39	1729.9	+1663.4

Table 6.3: Combined surface-grid refinement for the rotor case in VKI-SDPM.

Grid	N_c	N_r	Surface panels	$C_T \times 10^{-2}$	ΔC_T [%]	CPU time [s]	Δt [%]
Baseline	40	20	3200	0.4969	/	68.0	/
Refined	60	30	7200	0.5018	+0.99	207.35	+204.9
Fine	80	40	12800	0.5039	+1.41	602.025	+785.3

6.3 Sensitivity to Wake Parameters

The sensitivity analysis is now extended to the wake representation. In VKI-SDPM, the wake affects the solution through its contribution to the potential field and to the induced velocity field. Therefore, even for a fixed body mesh, the computed aerodynamic loads depend on how the wake is discretised, convected, and regularised.

The present section focuses on three aspects of the wake model. First, the influence of wake length is considered. Second, prescribed- and free-wake formulations are compared, in order to assess the sensitivity of the solution to the adopted wake-evolution model. Finally, the vortex-core regularisation is analysed.

For all wake configurations, the selected meshes are the baselines from Tabs. 6.2 and 6.3. In this way, the variations observed in the aerodynamic quantities can be attributed to the wake treatment. The monitored quantities are again the lift coefficient for the wing case and the thrust coefficient for the rotor case when the analysis is focused on wake length and convection model. For the analysis of the regularisation parameters specific indicators are selected, such as the sectional normal-force coefficient at the blade tip, the tip-vortex axial trajectory, and the radial contraction.

6.3.1 Wake Length

In a potential-flow formulation, the wake represents the vorticity released from the trailing edge and contributes to the induced velocity field acting on the body. If the wake is too short, its contribution to the solution may be underestimated since it is not fully developed. On the other hand, an excessively long wake increases the computational cost and may provide only marginal changes in the aerodynamic loads once the far-wake contribution becomes sufficiently weak.

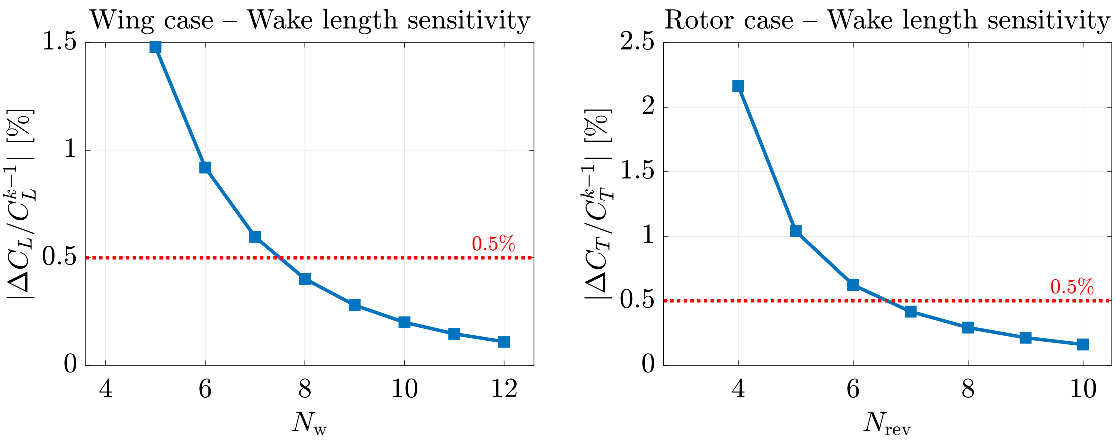
The definition of wake length depends on the type of configuration. For wing simulations, the wake length is expressed as a multiple of the reference chord,

$$l_{\mathrm{w}} = K_{\mathrm{w}} \; c_{\mathrm{ref}}, \tag{6.5}$$

where K_{w} is the nondimensional wake-length parameter. For rotor simulations, the wake length is more naturally expressed in terms of the number of revolutions N_{rev} retained in the computation.

The sensitivity to wake length is evaluated by progressively increasing K_{w}, or N_{rev}, while keeping the body mesh and the remaining wake parameters fixed. The relative variation of a monitored aerodynamic quantity is computed as in Eq. (6.3) with respect to consecutive wake lengths in the sequence.

Based on these results, the wake length retained for the validation campaign is $K_{\mathrm{w}} = 10$ for the wing cases and $N_{\mathrm{rev}} = 8$ for the rotor cases. These values provide a sufficient convergence of the monitored aerodynamic coefficients while avoiding the additional computational cost associated with a longer wake.



(a) Wing wake length sensitivity **(b)** Rotor wake length sensitivity

Figure 6.3: VKI-SDPM wake length sensitivity

6.3.2 Comparison between Prescribed- and Free-Wake Simulations

The second part of the wake sensitivity analysis concerns the comparison between prescribed- and free-wake formulations. This is used to determine how strongly the aerodynamic quantities depend on the selected wake-evolution model.

In the prescribed-wake approach, the wake is convected by the freestream only, avoiding the computation of the local induced velocity on the wake nodes. This reduces the computational cost but it neglects the deformation of the wake induced by body–wake and wake–wake interactions. Table 6.4 reports the values of the aerodynamic coefficients, the CPU time of the simulation and the relative differences between the prescribed- and free-wake simulations, defined as

$$\Delta C_{\text{rel}} = \frac{C_{\text{presc}} - C_{\text{free}}}{C_{\text{free}}} \cdot 100, \qquad \Delta t_{\text{rel}} = \frac{t_{\text{presc}} - t_{\text{free}}}{t_{\text{free}}} \cdot 100. \tag{6.6}$$

For the wing case, the prescribed-wake formulation is advantageous in terms of CPU time, reducing the CPU time by approximately 75% compared with the free-wake formulation. This reduction in computational cost is accompanied by a relative difference in the predicted lift coefficient of less than 0.2%. This can be explained by considering the wake representations in Fig. 6.4. The free-wake model allows wake roll-up, but its effects are less relevant since the freestream rapidly convects the wake downstream from the wing.

For the rotor simulations, the comparison shows a different trend. In Tab. 6.4 the relative difference between the prescribed- and free-wake simulations is larger but still around 5%. This can be explained by the more evident differences in the wake geometries reported in Fig. 6.5. In this case, the wake roll-up is concentrated closer to the rotor disk. For this reason, it has a greater influence on the computed results. In this case as well, the reduction in CPU time is considerable, amounting to approximately 73% compared with the free-wake simulation. However, unlike the wing case, the rotor case may require a trade-off between computational cost and wake-model fidelity. Based on the results of this comparison, the prescribed-wake formulation is retained for the wing validation cases, whereas the free-wake formulation is adopted for the rotor validation cases.

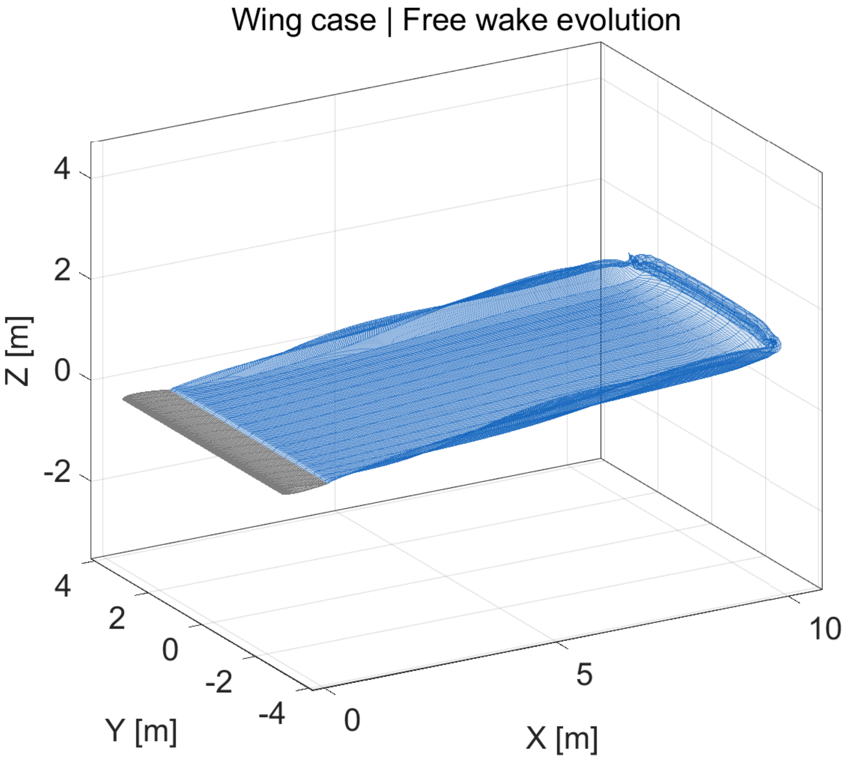
Table 6.4: Comparison between prescribed- and free-wake simulations.

Wing case

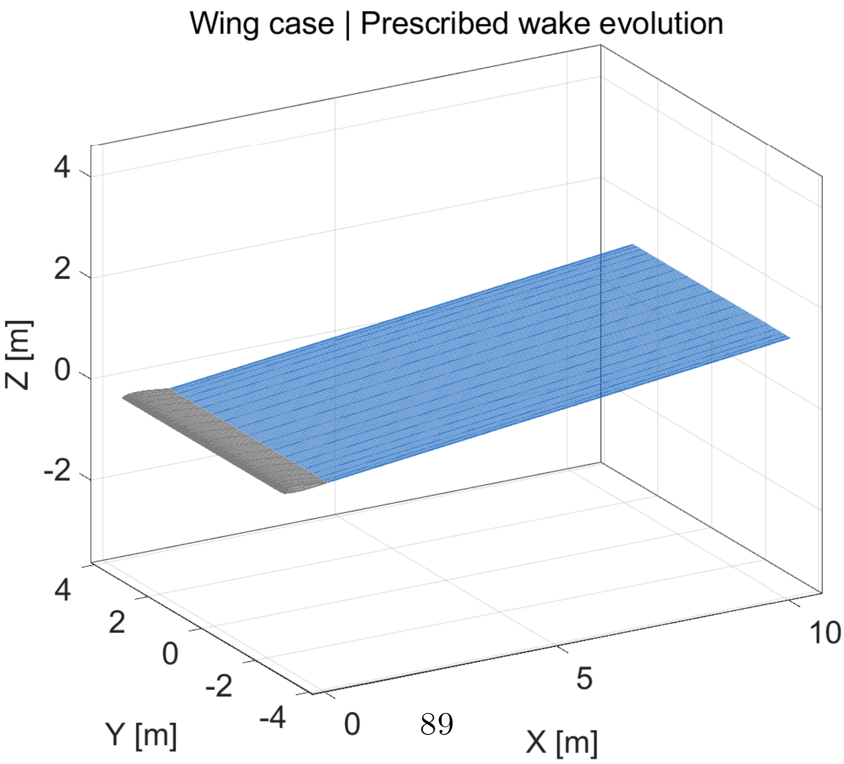
Wake	C_L	CPU [s]
Free	0.4633	408.3
Prescribed	0.4641	98.1
Δ (%)	+0.17	−76

Rotor case

Wake	$C_T \times 10^{-2}$	CPU [s]
Free	0.4733	257.3
Prescribed	0.4969	68.0
Δ (%)	+5	−73.6

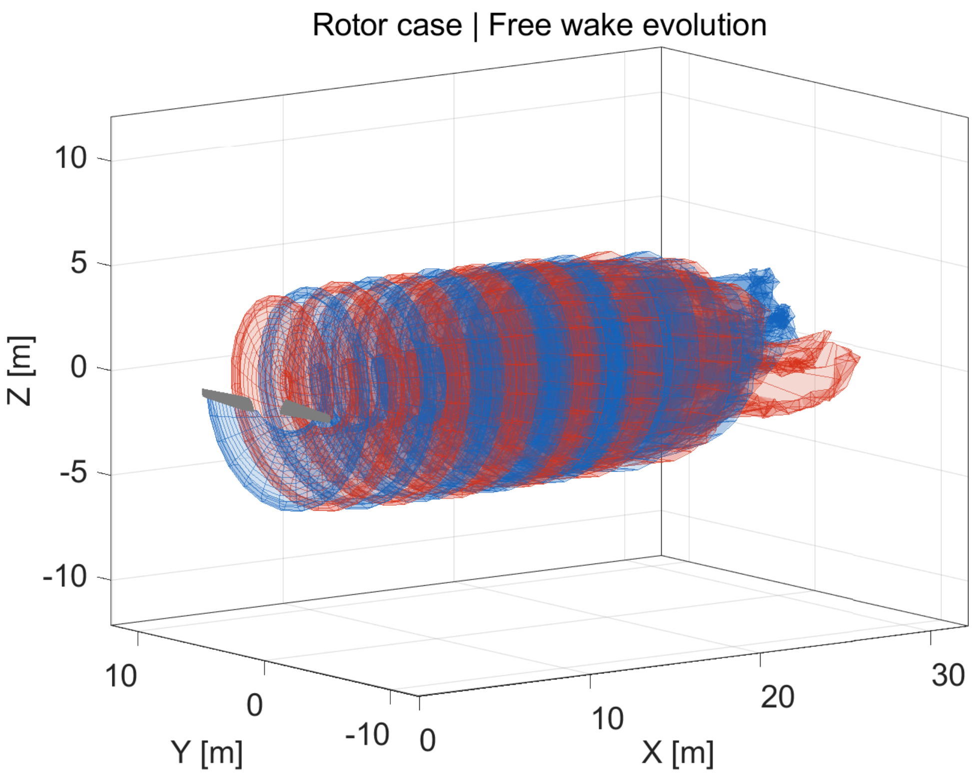


(a) Free wake evolution for the wing case

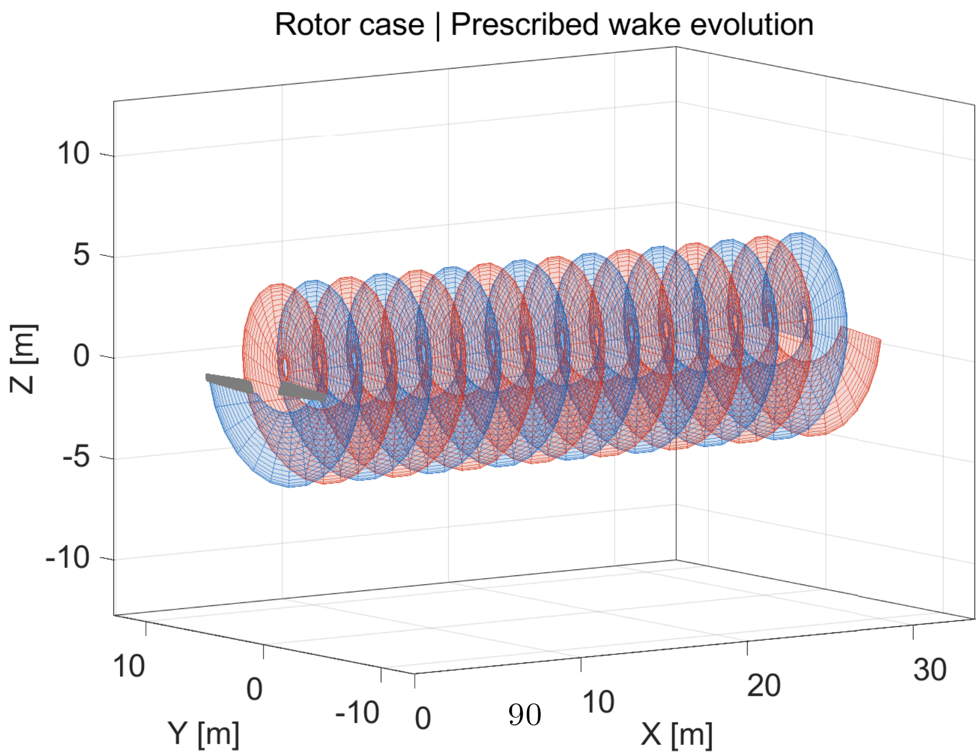


(b) Prescribed wake evolution for the wing case

Figure 6.4: Wake evolution model comparison for the wing case.



(a) Free wake evolution for the rotor test case.



(b) Prescribed wake evolution for the rotor test case

Figure 6.5: Wake evolution model comparison for the rotor case

6.3.3 Sensitivity to Regularisation Parameters

The final wake-related study concerns the regularisation parameters used in the Biot–Savart law. This includes the regularisation functions, such as the Lamb–Oseen and Vatistas models, the stretching and wake-age-based core-growth models, and the initial core size. Since preliminary tests did not show appreciable sensitivity to the regularisation parameters for the wing case, the analysis is restricted to the rotor case.

Different combinations of vortex-core models and core-growth laws are compared, including the Lamb–Oseen and Vatistas regularisations together with stretching-based and wake-age-based core-growth models. The comparison is performed by analysing the difference in values for the sectional normal force coefficient at the blade tip for different values of the initial core size for all regularisation models considered.

Figure 6.6 illustrates the results of this comparison for different values of $r_{c,0}$. It can be seen that the Lamb–Oseen model generally predicts lower loads, with the minimum value obtained when it is combined with the stretching-based core-growth model. However, the spread among the different model combinations remains within approximately 1%.

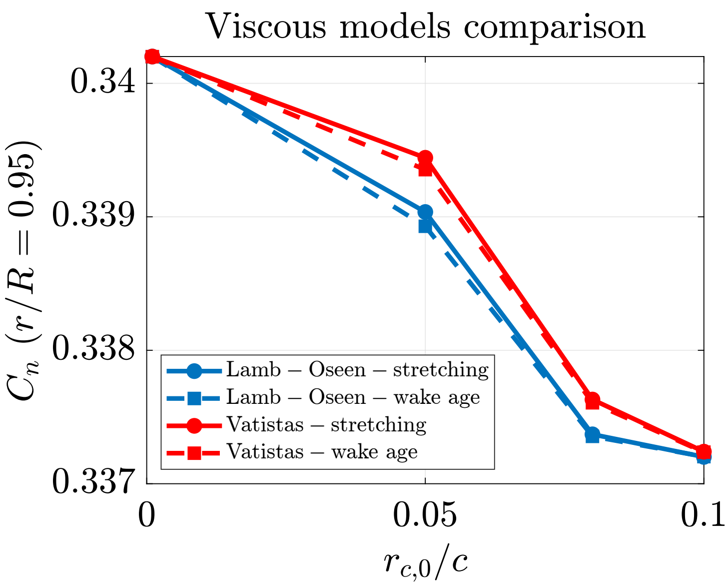


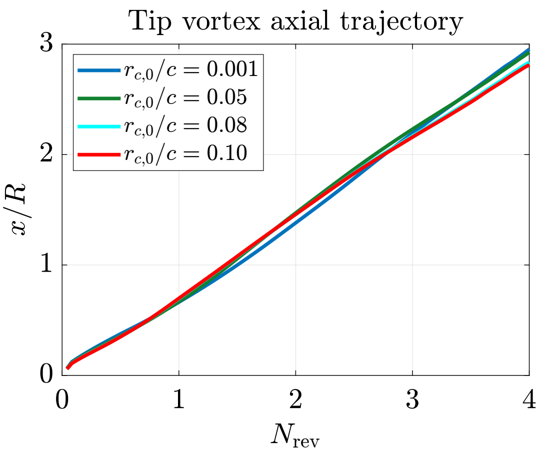
Figure 6.6: Comparison between different viscous models and core growth models at the blade tip.

To better assess the influence of the initial vortex-core size, its effects on the radial contraction and the axial trajectory of the tip vortices are analysed. The Lamb–Oseen model is combined with the wake age core-growth model to provide the most dissipative model combination considered in the comparison, as shown in Fig. 6.6. The upper limit of the initial core size is set by following Gupta [19], who indicates that a first estimate may be taken as 10% of the reference chord.

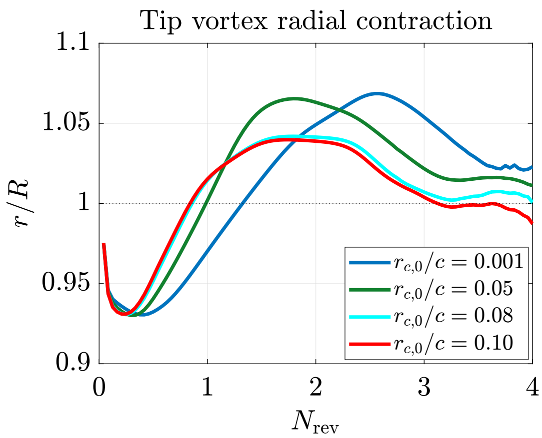
The tip-vortex axial trajectory is reported in Fig. 6.7a. No evident differences are observed for the different values of $r_{c,0}$. On the other hand, the radial contraction is more sensitive to the selected value of $r_{c,0}$, as reported in Fig. 6.7b. Higher values of $r_{c,0}$ tend to further smooth the wake-induced velocities and reduce the radial contraction or expansion of the wake.

For the validation campaign, the Lamb–Oseen regularisation with wake-age-based core growth and $r_{c,0} = 10\%\ c_{\mathrm{ref}}$ is retained. This choice follows the literature-based estimate for the initial core size and provides a stable regularisation of the near-wake induced velocities [18][19].

However, wake-dominated operating configurations, such as hover flight, may require different choices of the regularisation parameters, as will be discussed in Sec. 7.5.



(a) Tip vortex axial trajectory



(b) Tip vortex radial contraction

Figure 6.7: Tip vortex trajectories for the combined Lamb–Oseen wake-age regularisation model for different values of initial core size.

6.4 Parallel Computing Performance

The computational cost of VKI-SDPM is mainly associated with the repeated evaluation of influence coefficients and induced velocities. Several of these operations can be evaluated independently and are therefore suitable for parallel execution. For this reason, VKI-SDPM was implemented with a parallel-computing strategy. The effect of the number of MATLAB workers on the overall wall-clock time is therefore analysed in this section.

The parallel performance analysis is carried out for the free-wake rotor case, which represents the most computationally demanding configuration among those considered in the validation campaign. The analysis is carried out on a mesh with $N_c = 40$ chordwise panels and $N_r = 20$ radial panels and an azimuthal step of $\Delta\varphi = 15°$. The total number of time steps executed per run is 72, corresponding to three complete rotor revolutions. For each number of workers, the simulation is repeated three times and the median wall-clock time is reported; the minimum and maximum observed times are also included to characterise the run-to-run variability.

The simulations were executed on a single workstation, whose main hardware and software specifications are reported in Table 6.5. Parallel execution was performed using the MATLAB Parallel Computing Toolbox with local workers. No GPU acceleration was used.

Table 6.5: Computational environment used for the VKI-SDPM performance analysis.

Processor	13th Gen Intel(R) Core(TM) i9-13980HX
Physical cores / logical threads	24 / 32
RAM	16 GB
Operating system	Microsoft Windows 11 Home 10.0.26200 64-bit
MATLAB version	2024a
Parallelisation	MATLAB Parallel Computing Toolbox, local workers
GPU acceleration	Not used

Parallel speed-up and parallel efficiency are standard metrics used to quantify the benefit of executing a fixed computational workload on multiple processing units [54]. The speed-up measures the reduction of wall-clock time with respect to a reference execution, while the efficiency measures how close the obtained speed-up is to the ideal linear scaling. For a fixed problem size, the maximum achievable speed-up is limited by the fraction of the algorithm that cannot be parallelised, as expressed by Amdahl's law [55].

The parallel speed-up is defined as

$$S_{\mathrm{pw}} = \frac{t_1}{t_{\mathrm{pw}}}, \tag{6.7}$$

where t_1 is the representative wall-clock time obtained with one worker and t_{pw} is the corresponding time obtained with p_{work} workers [54]. The parallel efficiency is defined as

$$E_{\mathrm{pw}} = \frac{S_{\mathrm{pw}}}{p_{\mathrm{work}}} \cdot 100. \tag{6.8}$$

In the present analysis, the case $p_{\text{work}} = 1$ corresponds to the execution with one MATLAB local worker and is used as the reference time for the speed-up calculation. It should not be interpreted as a strictly serial implementation of the solver, since the parallel execution framework and MATLAB internal operations may still introduce overheads. The measured speed-up therefore quantifies the benefit of increasing the number of local workers with respect to the one-worker execution of the same workflow.

For comparison with the measured speed-up, Amdahl's law is also considered. If f_{pw} denotes the parallelisable fraction of the measured workload, the theoretical speed-up for a fixed problem size can be written as

$$S_{p_{\text{work}},\text{Amdahl}} = \frac{1}{(1 - f_{\text{pw}}) + \dfrac{f_{\text{pw}}}{p_{\text{work}}}}, \tag{6.9}$$

This relation shows that the speed-up does not increase indefinitely with p_{work}, since the serial fraction of the workflow and the parallel overhead progressively limit the benefit of adding more workers.

In the present analysis, f_{pw} is estimated from the measured data by inverting Eq. (6.9) at the maximum number of workers $N_{\max} = 24$, yielding

$$f_{\text{pw}} = \frac{1 - 1/S_{N_{\max}}}{1 - 1/N_{\max}}, \tag{6.10}$$

where $S_{N_{\max}}$ is the measured speed-up at $p_{\text{work}} = N_{\max}$. This estimate assumes that the Amdahl model holds throughout the measured range and provides an effective value of f_{pw} for the timed workflow.

The reported wall-clock times are measured over a timed window of 48 time steps, obtained after excluding the first 24 steps from the measurement. This warm-up window corresponds to the first rotor revolution, during which the wake is still developing and the number of active vortex segments is not yet representative of a regime computation. The results should be interpreted as the effective parallel performance of this timed workflow, rather than as the isolated performance of a single parallel kernel. The median time is indicated as t_{med}, while $t_{\min}$ and $t_{\max}$ provide an estimate of the variability among repeated runs.

The measured speed-up and efficiency are reported in Table 6.6. Each entry corresponds to the median of three independent runs performed with the same configuration; the minimum and maximum values are also reported to characterise the run-to-run variability.

Table 6.6: Parallel performance versus number of workers.

N_{work}	t_{med} [s]	t_{min} [s]	t_{max} [s]	Speed-up S_{pw}	Efficiency E_{pw} [%]
1	1092.43	1079.49	1104.03	1.000	100.0
2	588.10	583.89	603.33	1.858	92.9
4	335.98	335.54	340.77	3.252	81.3
6	276.30	276.14	281.45	3.954	65.9
8	249.10	248.42	252.46	4.386	54.8
12	227.00	224.67	228.96	4.813	40.1
16	219.59	215.42	220.27	4.975	31.1
24	218.57	213.84	219.66	4.998	20.8

The results show a clear reduction of the computational time when the number of workers is increased from one to four, with an efficiency that remains above 80%. The wall-clock time continues to decrease for larger numbers of workers, but the marginal gain progressively reduces. The minimum representative time is obtained with $p_{\text{work}} = 24$ workers, for which the measured speed-up is approximately $S_{\text{pw}} = 5.0$. It is noted that the difference in median wall-clock time between sixteen and twenty-four workers is less than 0.5%, which falls within the variability range observed across repeated runs, as indicated by the overlap between the respective t_{min} and t_{max} values. The saturation observed at twenty-four workers should therefore be interpreted as a plateau rather than as a definitive minimum.

The Amdahl-law fit provides an additional interpretation of the observed saturation. The estimated effective parallel fraction, $f_{\text{pw}} = 0.8346$, corresponds to an ideal asymptotic speed-up of

$$S_{\text{max}} = \frac{1}{1 - f_{\text{pw}}} \simeq 6.0. \tag{6.11}$$

The maximum measured speed-up, approximately equal to $S_{\text{pw}} \simeq 5$, is therefore consistent with the theoretical limit predicted by the fitted Amdahl model. The difference between the measured value and the asymptotic limit can be attributed to parallel overhead, memory-access limitations, and residual serial operations in the implementation. Since the timing was performed on the timed workflow described above, the fitted value of f_{pw} should be interpreted as an effective value for this workflow, not as the intrinsic parallel fraction of the individual solver kernels.

The measured speed-up is compared in Fig. 6.8 with the ideal linear behaviour $S_{\text{pw}} = N_{\text{work}}$ and with the Amdahl-law estimate. The ideal trend is not reached because only part of the measured workflow can be effectively parallelised, while the remaining operations are serial or affected by memory access, scheduling, and parallelisation overhead. For this reason, the measured curve progressively departs from the ideal scaling as the number of workers increases.

The same behaviour is observed in terms of parallel efficiency, reported in Fig. 6.9. The efficiency remains above 80% up to four workers, but decreases as the number of workers is further increased, with progressively diminishing benefits in terms of wall-clock-time reduction.

Different worker configurations can therefore be identified depending on the selected criterion. Up to four workers, the efficiency remains high and the scaling is close to the ideal behaviour. Between eight and twelve workers, the wall-clock time is further reduced, although at the cost of a lower parallel efficiency. The minimum measured time is obtained with twenty-four workers. For this reason, the validation campaign presented in this work is conducted using $p_{\text{work}} = 24$ workers.

These results also provide useful indications for future developments of VKI-SDPM. The parallel performance could be further improved by reducing the overhead associated with worker scheduling, memory access, and data transfer. This would likely require a more detailed optimisation of the data structures and of the operations performed inside the most expensive interaction loops. Nevertheless, at the present stage of development, physical validation of the solver takes priority over the full optimisation of its computational performance. The measured speed-up is therefore considered satisfactory for the analyses carried out in this work.

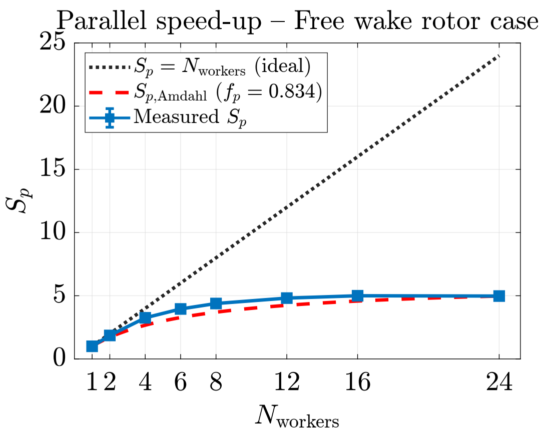


Figure 6.8: Parallel speed-up for the free-wake rotor case. The measured speed-up is compared with the ideal linear scaling and with the Amdahl-law estimate.

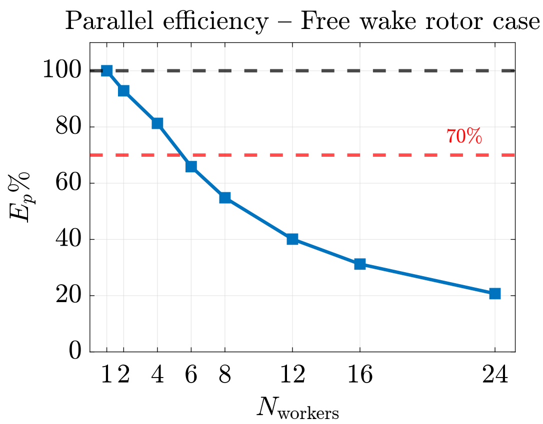


Figure 6.9: Parallel efficiency for the free-wake rotor case as a function of the number of MATLAB workers.

Chapter 7

VKI-SDPM: Validation

7.1 Validation Strategy

The purpose of this chapter is to assess the predictive capability of VKI-SDPM through comparison with experimental benchmark cases. The validation is not intended as a single global test of the solver, but rather as a sequence of targeted comparisons designed to examine different aspects of the numerical formulation such as the prediction of pressure distributions and aerodynamic coefficients, the modelling of fixed and rotating lifting surfaces, and the representation of both prescribed and free wakes.

A central aspect of the validation strategy is the consistency between the selected experimental conditions and the inviscid, potential-flow representation of the numerical method. Consequently, the comparisons are focused primarily on conditions in which the flow remains attached or only mildly affected by viscous phenomena. Cases dominated by separation, stall, or strong compressibility effects are not used as primary validation points. When such effects are present in the experimental datasets they are discussed in relation to VKI-SDPM. Therefore, the chapter provides both a quantitative assessment of the solver and a critical discussion of its present range of applicability.

The order of the validation cases follows a progression from simpler to more demanding configurations. The Yip–Shubert case is used first because it provides a benchmark for the straight wing configuration in the low-subsonic regime [9]. The Kolbe–Boltz case then extends the assessment to a swept and tapered wing [10]. This makes the case useful for assessing the capability of the solver to handle more demanding geometries. The NREL Phase VI experiment introduces a rotating, twisted, and tapered blade configuration representative of wind-turbine aerodynamics [11]. In addition, the availability of both axial and yawed operating conditions makes this case relevant for assessing the ability of the method to capture non-axisymmetric inflow conditions. Finally, the Caradonna–Tung hover-rotor case is used to assess the ability of the method to reproduce the aerodynamic behaviour under the challenging hover condition [12]. In this flight configuration, the wake evolution plays a central role in determining the induced velocity field and the resulting steady solution. The test case is therefore used as a benchmark to assess the effective free-wake capabilities of VKI-SDPM.

Each benchmark is treated independently in the following sections, which describe the experimental configuration, numerical setup, comparison quantities, and observed agreement and discrepancies.

7.2 Yip–Shubert Test Case (NASA TN D-8307)

The first validation case is based on the experimental measurements of Yip and Shubert on a finite semispan wing in subsonic flow. The experiment was originally designed to provide detailed pressure data on finite three-dimensional wings for the validation of computational aerodynamic methods [9].

The objective of this benchmark is to verify whether the solver can reproduce the main aerodynamic quantities associated with a finite straight wing under attached-flow conditions. In particular, the comparison focuses on surface pressure distributions, integral force and moment coefficients, and spanwise sectional loading.

7.2.1 Experimental Configuration

The experimental model consisted of a semispan wing tested in the Langley V/STOL tunnel. The wing was mounted on an adjustable turntable and connected to a support system that allowed the model to be rotated rearward, thereby imposing different effective sweep angles. A reflection plate was installed at the root to approximate the symmetry of a full-span wing, while appropriate wing-tip fairings were used to keep the tip aligned with the freestream as the wing was swept. The experimental setup is schematically illustrated in Fig. 7.1.

The wing had a constant NACA 0012 airfoil section, a nominal chord of $c = 1$ m, and a nominal semispan of $b/2 = 3$ m. The planform was rectangular, corresponding to a taper ratio $\lambda = 1.0$. The tested sweep angles ranged from 0° to 40°.

The experimental pressure distributions were obtained from 600 pressure orifices, with 60 taps installed at each of 10 spanwise pressure stations. The spanwise stations were non-uniformly distributed, with a higher concentration toward the outboard region. For the unswept configuration, their locations are defined by Eq. (7.1), where η is the non-dimensional semispan coordinate, $\eta = y/(b/2)$.

$$\eta = 0.05 + 0.95\ \cos\left[\frac{\pi(10-k)}{20}\right], \qquad 0 \leq k \leq 9 \tag{7.1}$$

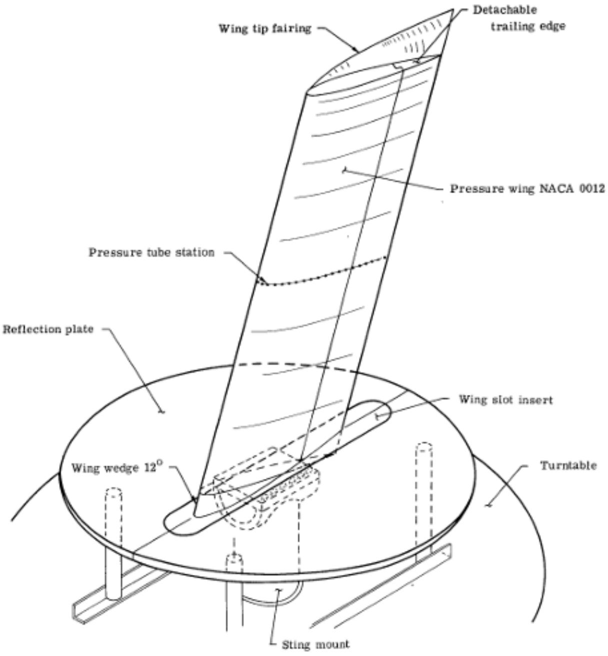


Figure 7.1: Schematic of the experimental setup used by Yip and Shubert [9].

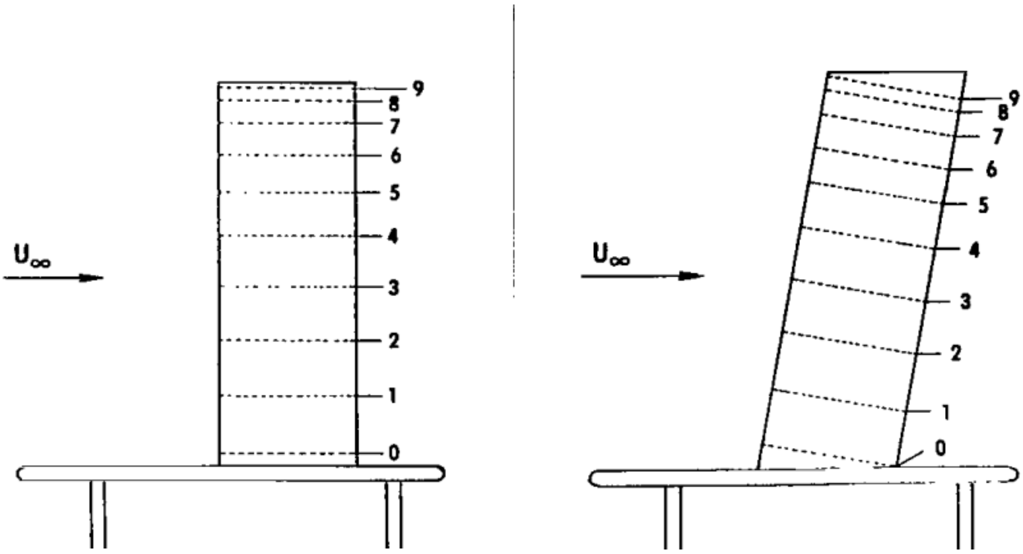
Because the swept configurations were obtained by rotating the straight model, the pressure-tap rows were also rotated, as shown in Fig. 7.2. This makes the reconstruction of purely chordwise pressure distributions less direct than in the unswept case. For this reason, the swept configurations are not used in the present benchmark, while the swept-wing validation is addressed in Sec. 7.3, where the pressure-tap layout is more suitable for a direct comparison [10].

A transition strip was installed on both upper and lower surfaces at $x/c = 0.05$. Therefore, the experimental data should be interpreted as fixed-transition measurements, whereas the present numerical model remains based on an inviscid potential-flow formulation.

Two corrections were applied to the angle of attack: the first accounts for jet-boundary effects following [56], while the second accounts for the wind-tunnel flow angularity and model misalignment, as reported in [9]. Accordingly, the simulations are run with the corrected values of incidence, and the selected experimental conditions are limited to the range

$$-5.91^\circ \leq \alpha \leq 19.35^\circ.$$

The freestream conditions were corrected for blockage effects due to the model and support system following [57], and these corrected values are also adopted in the simulations. For the unswept case, the resulting freestream dynamic pressure



(a) Straight wing configuration. (b) Swept wing configuration.

Figure 7.2: Pressure tap locations for the Yip–Shubert test case in straight and swept configurations [9].

is in the range

$$1460 \text{ Pa} \leq q_\infty \leq 1480 \text{ Pa}.$$

Under standard conditions, this corresponds to $M \approx 0.14$, which is sufficiently below the compressibility threshold.

7.2.2 Numerical Configuration

The VKI-SDPM numerical model reproduces the unswept Yip–Shubert wing as a full-span configuration and is discretised with $N_c = 80$ chordwise panels and $N_r = 20$ spanwise panels, as determined from the sensitivity analysis of Sec. 6.2. The chordwise and spanwise panel distributions follow sine and cosine spacing laws, respectively. The same discretisation is used for all angles of attack.

The NACA 0012 airfoil geometry is generated from the analytical formula reported in [35], with a modified last coefficient, resulting in the normalised thickness coordinate $\tilde{y}$ expressed as

$$\tilde{y} = 0.6 \left[0.2969 \sqrt{\frac{x}{c}} - 0.1260 \, \frac{x}{c} - 0.3516 \left(\frac{x}{c}\right)^2 + 0.2843 \left(\frac{x}{c}\right)^3 - 0.1036 \left(\frac{x}{c}\right)^4\right] \tag{7.2}$$

This modification ensures analytical closure at the TE and avoids manual clamping. The chordwise discretisation and the full mesh are reported in Figs. 7.3 and 7.4, respectively.

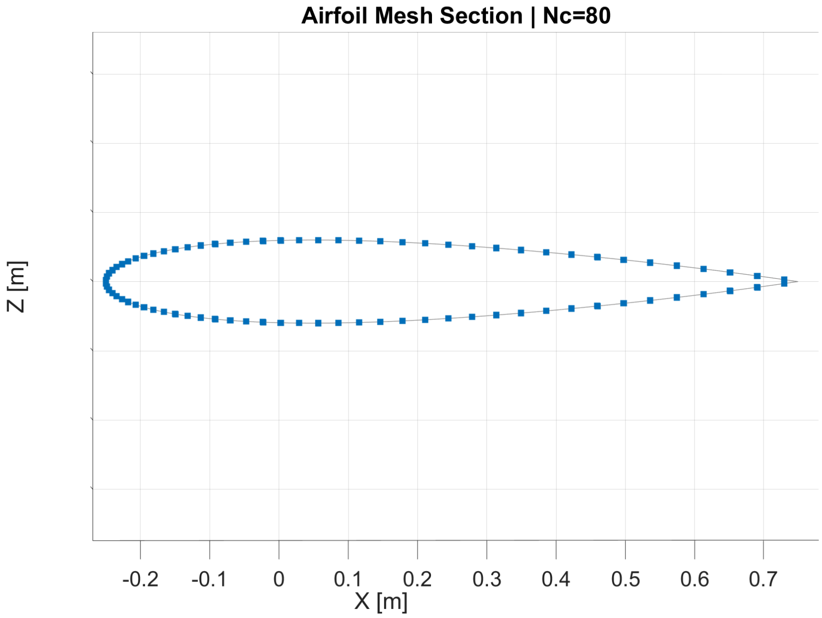


Figure 7.3: Numerical mesh for the airfoil profile NACA 0012. Here the blue markers represent the centroids of the surface panels. Their chordwise location is determined by a sine spacing law.

As a result of the analysis in Sec. 6.3, the wake is modelled as prescribed and no regularisation parameters need to be assigned. The wake length is set to

$$l_{\mathrm{w}} = K_{\mathrm{w}}\; c_{\mathrm{ref}} = 13\; c_{\mathrm{ref}},$$

where c_{ref} is the nominal chord. This value of the wake length was sufficient for the residual $\mathcal{R}(\mu)$, defined in Eq. (7.3), to reach the numerical threshold of $\varepsilon_{\mu} = 10^{-4}$ for every simulated condition.

$$\mathcal{R}(\mu) = \frac{\max\left(|\boldsymbol{\mu}^{n} - \boldsymbol{\mu}^{n-1}|\right)}{\max\left(|\boldsymbol{\mu}^{n-1}|\right)} \tag{7.3}$$

7.2.3 Comparison Quantities and Coefficient Conventions

The validation is based on the comparison between experimental and numerical values for different aerodynamic coefficients.

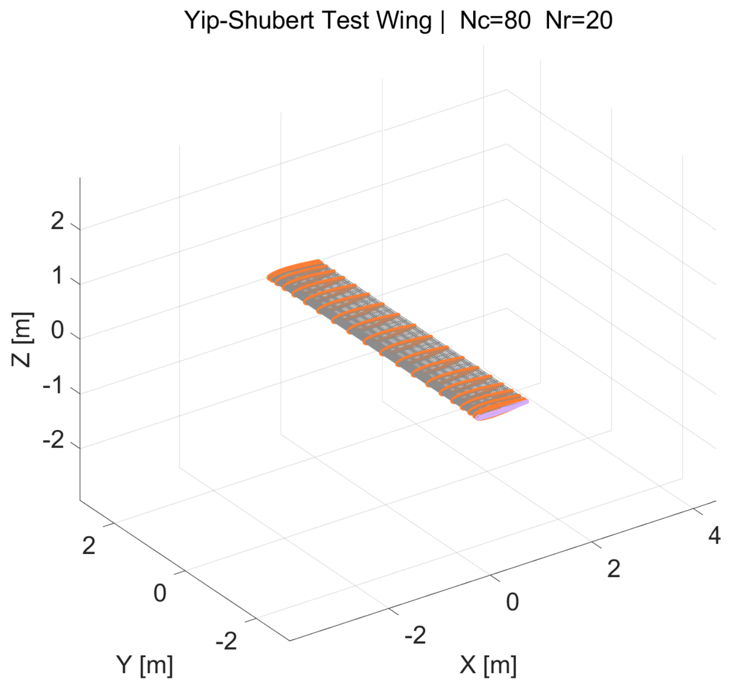


Figure 7.4: Surface mesh for the Yip–Shubert test case validation campaign.

The experimental values of C_p are compared with numerical ones by linearly interpolating the results at the two closest spanwise stations. The convention adopted in the report is to plot the pressure-coefficient distributions against $\sqrt{x/c}$, rather than x/c, in order to expand the leading-edge region. Since the experimental pressure data were extracted by digitising the published plots, the numerical pressure distributions are also presented using the same chordwise convention.

The authors also state that some pressure data points affected by instrumentation malfunction or tube blockage were corrected by linear interpolation between adjacent values [9]. Since the present comparison relies on digitised pressure distributions from the published plots, this represents an additional source of uncertainty that should be considered when interpreting local discrepancies.

For this validation, the total normal-force coefficient C_N is obtained, as in the experimental case, by integrating the sectional normal-force coefficient C_n along the semispan,

$$C_N = \int_0^1 C_n \; d\eta, \tag{7.4}$$

where C_n is obtained from the chordwise pressure integration,

$$C_n = \int_0^1 (C_{p,low} - C_{p,up})\, d\left(\frac{x}{c}\right). \tag{7.5}$$

The pitching-moment coefficients are also compared using the same experimental convention. The total pitching-moment coefficient C_M and the sectional pitching-moment coefficient C_m are defined about the leading edge, and C_m is evaluated using the following expression:

$$C_m = \sum_{i=1}^{n-1} \left[\frac{(x_{i+1}^2 - x_i^2) + (z_{i+1}^2 - z_i^2)}{c^2}\right] \left(\frac{C_{p,i+1} + C_{p,i}}{4}\right). \tag{7.6}$$

Here, n is the number of pressure taps per station, i denotes the i-th pressure tap, and c is the chord. Eq. (7.6) accounts for both the moment contribution along the z-axis and the one along the x-axis [9].

The simulations were performed for almost all the available angles of attack for the unswept test case, but only the C_p distributions indicated in Tab. 7.1 are reported in the thesis.

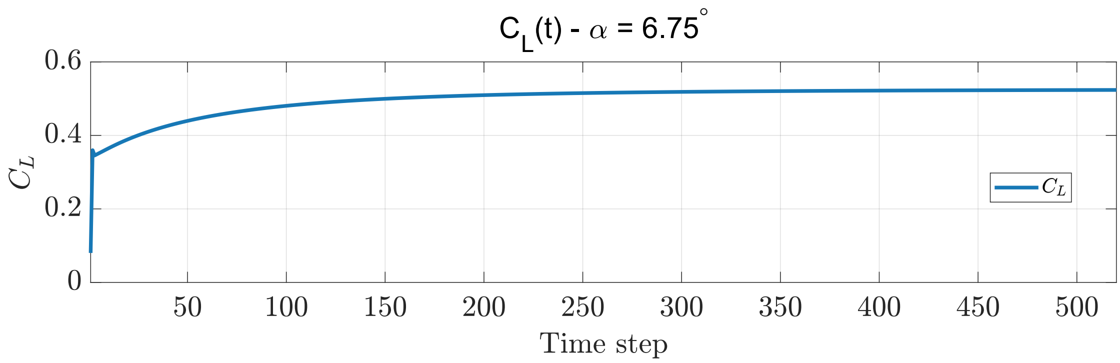
Table 7.1: List of simulated and reported test cases for selected angles of attack.

Experimental condition	Simulated	C_p distribution reported
$q_\infty = 1.48$ kPa, $\alpha = -5.91°$	✓	✗
$q_\infty = 1.48$ kPa, $\alpha = -3.78°$	✗	✗
$q_\infty = 1.47$ kPa, $\alpha = -1.69°$	✓	App. D.1.1
$q_\infty = 1.47$ kPa, $\alpha = -0.62°$	✗	✗
$q_\infty = 1.47$ kPa, $\alpha = 0.42°$	✓	✗
$q_\infty = 1.47$ kPa, $\alpha = 2.53°$	✓	Sec. 7.2.4
$q_\infty = 1.48$ kPa, $\alpha = 4.64°$	✓	App. D.1.2
$q_\infty = 1.47$ kPa, $\alpha = 6.75°$	✓	Sec. 7.2.4
$q_\infty = 1.48$ kPa, $\alpha = 8.85°$	✓	✗
$q_\infty = 1.47$ kPa, $\alpha = 10.97°$	✓	App. D.1.3
$q_\infty = 1.48$ kPa, $\alpha = 13.06°$	✓	✗
$q_\infty = 1.46$ kPa, $\alpha = 15.16°$	✓	✗
$q_\infty = 1.48$ kPa, $\alpha = 17.26°$	✓	✗
$q_\infty = 1.48$ kPa, $\alpha = 19.35°$	✓	✗

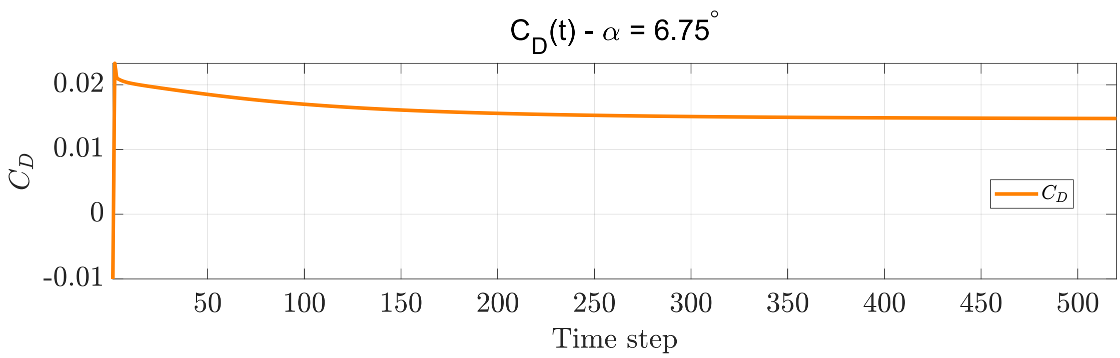
7.2.4 Results and Discussion

The solver is first assessed in terms of integral quantities and is then progressively refined by considering sectional loading and local pressure distributions.

All the reported results were obtained after convergence of the steady solution. As anticipated above, the solution was considered converged when the value of $\mathcal{R}(\mu)$ reached the prescribed threshold of $\varepsilon_\mu = 10^{-4}$. An example of the corresponding convergence of the global aerodynamic coefficients can be observed in Fig. 7.5, where the histories of C_L and C_D are reported for the case $\alpha = 6.75°$ and $q_\infty = 1.47$ kPa.



(a) $C_L(t)$



(b) $C_D(t)$

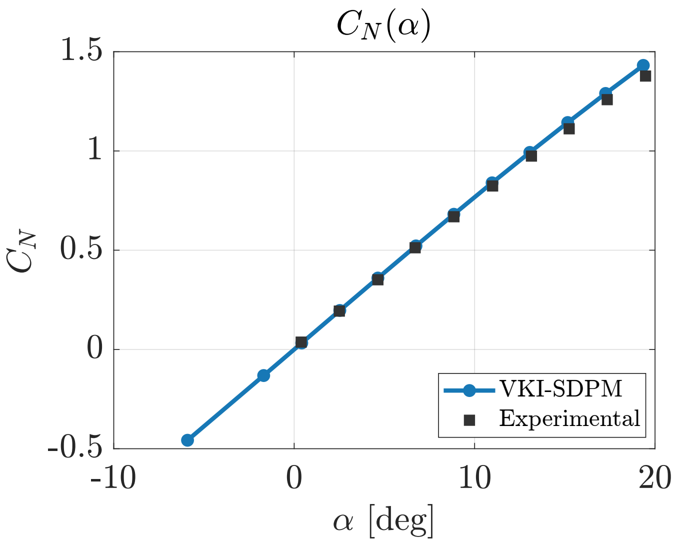
Figure 7.5: C_L and C_D histories for the Yip–Shubert test case at $\alpha = 6.75°$ and $q_\infty = 1.47$ kPa.

Global Aerodynamic Coefficients

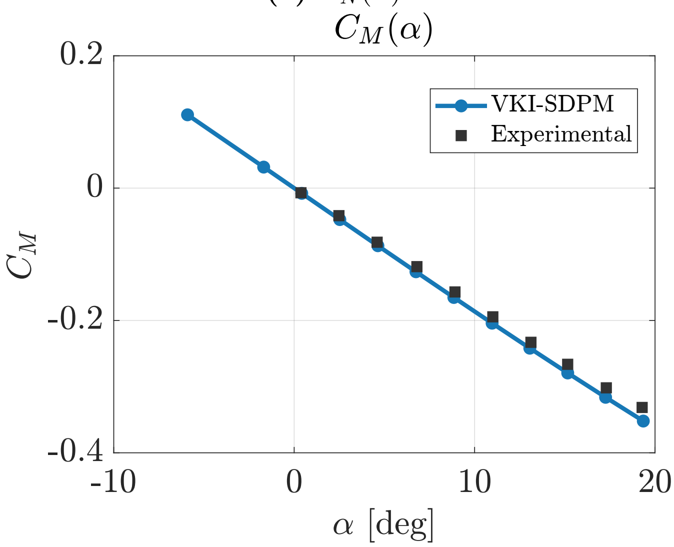
The first comparison concerns the total normal-force coefficient as a function of the angle of attack, providing a compact measure of the global lifting behaviour of the wing. As illustrated in Fig. 7.6a, the numerical prediction follows the experimental trend over the considered range of α. The first noticeable but limited deviations appear between $\alpha \approx 12°$ and $15°$.

The pitching-moment coefficient $C_M(\alpha)$ provides a complementary assessment because it is more sensitive to the chordwise pressure distribution. As shown in

Fig. 7.6b, the general trend is maintained and, although discrepancies emerge at lower incidences than in the $C_N(\alpha)$ comparison, they remain limited over a wide range of α. This may suggest that differences in the pressure distributions start to emerge at moderate incidences but remain localised and limited in magnitude.



(a) $C_N(\alpha)$.



(b) $C_M(\alpha)$.

Figure 7.6: Normal-force coefficient and pitching moment coefficient versus angle of attack. Comparison for the Yip–Shubert test case.

Spanwise Loading

A more detailed interpretation of the aerodynamic loading can be obtained by analysing the spanwise behaviour of the sectional pitching-moment coefficient and the normal force coefficient.

Figure 7.7 reports the curves $C_m(\alpha)$ at the spanwise stations $\eta = 0.05,\ 0.48,\ 0.82,\ 0.99$. At a given value of α, the agreement generally improves from the root towards the midspan region. However, as the tip is approached, discrepancies emerge at lower incidences and become more pronounced. This behaviour is consistent with the observations reported by Yip [9], where the onset of a significant tip-flow influence is indicated around $\alpha = 4°$. Therefore, larger discrepancies are expected near the tip. The remaining $C_m(\alpha)$ distributions are reported in App. D.1.4.

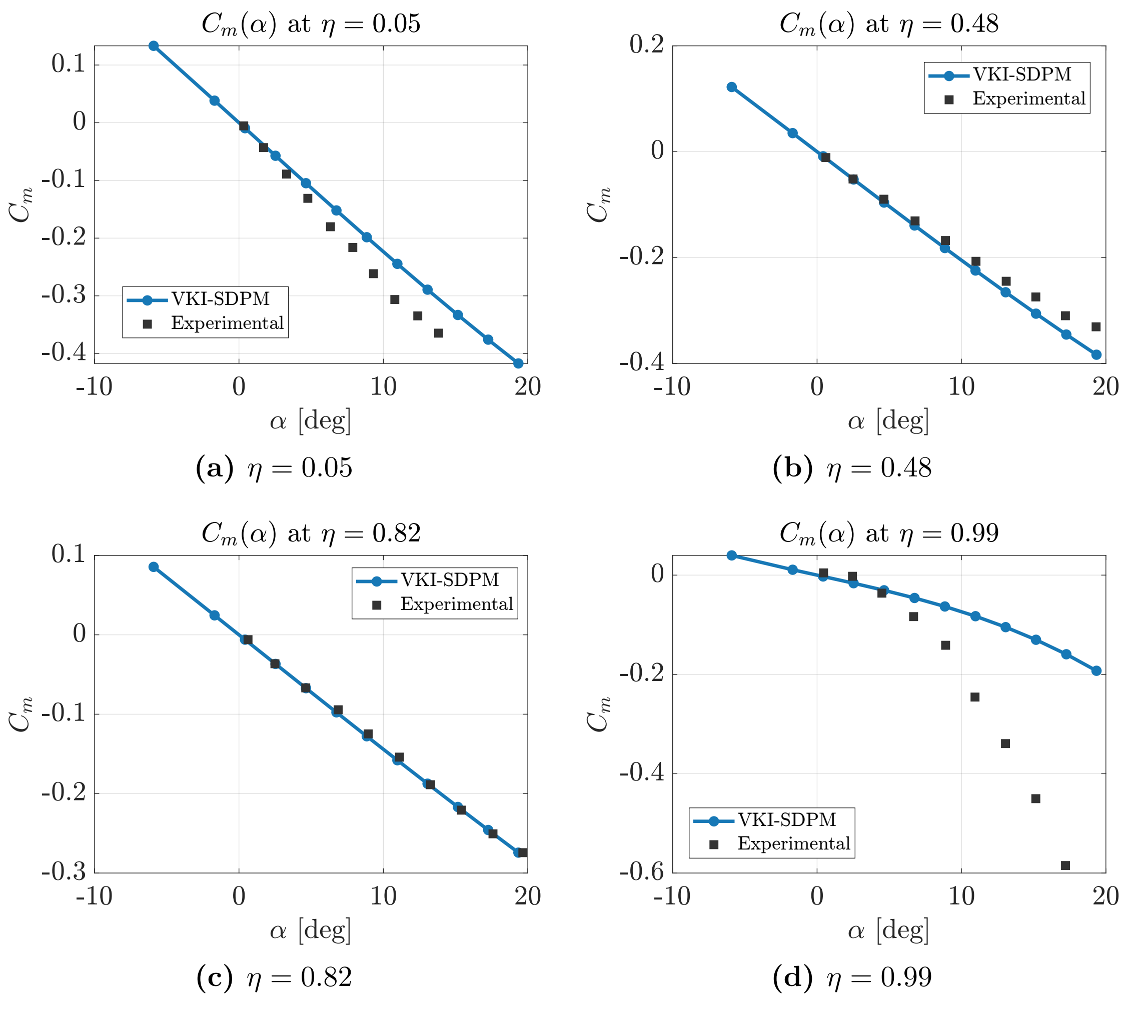


Figure 7.7: $C_m(\alpha)$ at different spanwise stations for the Yip–Shubert test case.

Figure 7.8 compares the normalised sectional loading, C_n/C_N, obtained from VKI-SDPM with the experimental data at $\alpha = 6.75°$. The considered operating condition already lies in a range where the tip-flow influence is relevant; therefore, the reduction in agreement at the tip is expected. The other relevant discrepancy is located near the root and is most likely associated with the partial effectiveness of the reflection plate, as discussed in [9].

From these results and observations, it can be said that both $C_m(\alpha)$ and $C_n(\eta)$ indicate that the midspan region is well reproduced. the largest discrepancies are concentrated near the root and tip where experimental uncertainties and modelling limitations are expected to be more relevant.

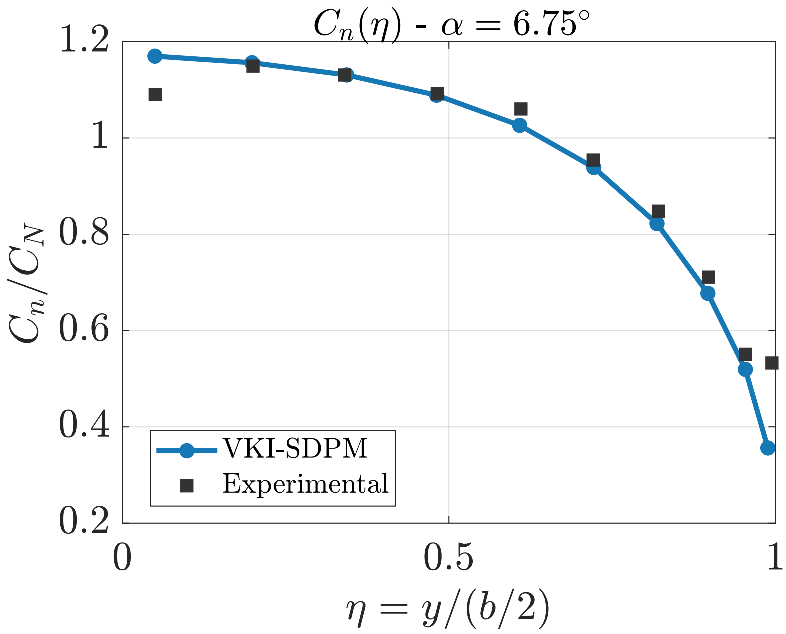
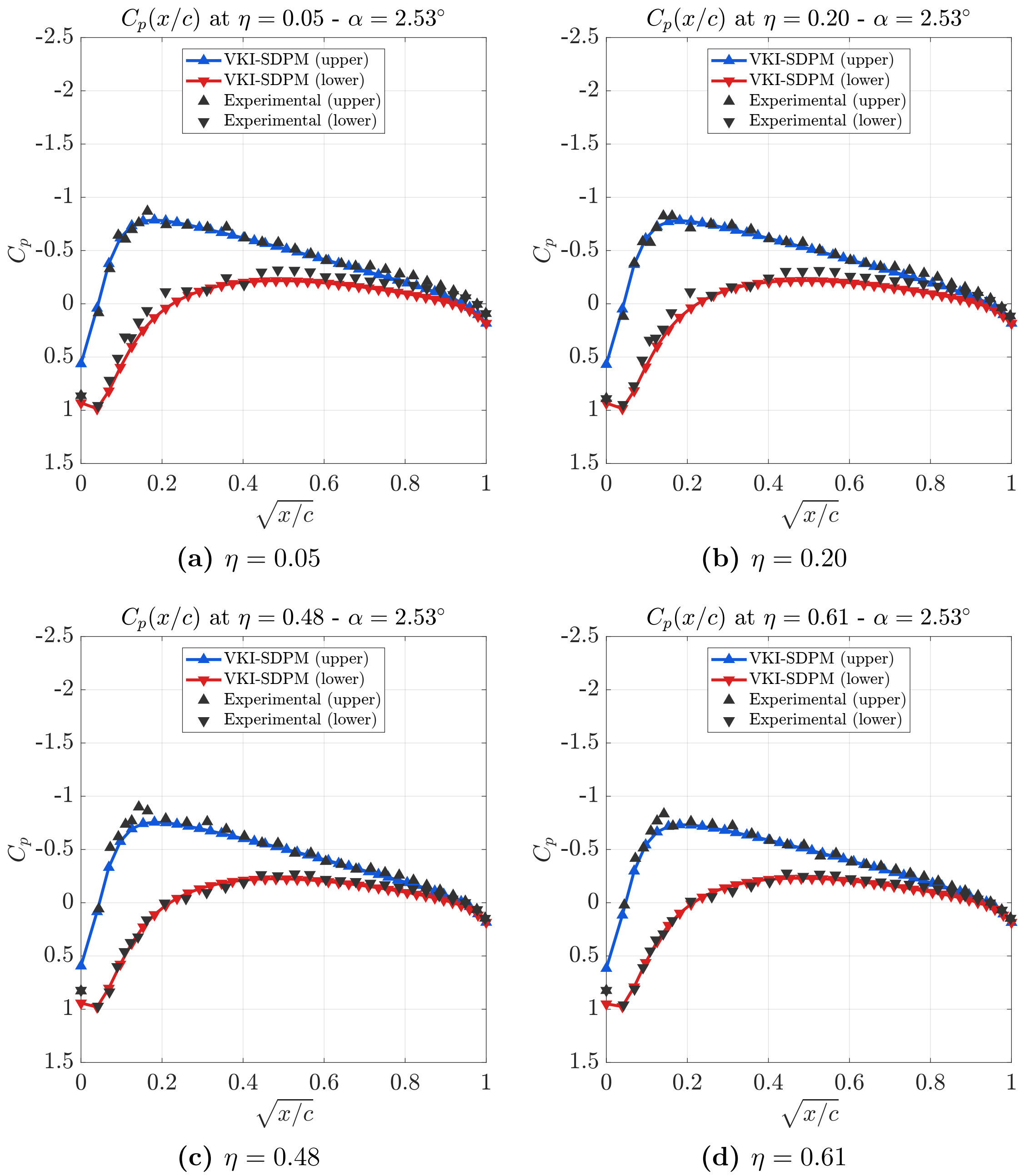


Figure 7.8: Normalised spanwise loading C_n/C_N for $\alpha = 6.75°$.

Pressure Coefficient Distribution

A closer assessment is obtained by examining the pressure distributions along the span. The results are reported for a low- and moderate-incidence case, which are considered meaningful for the discussion.

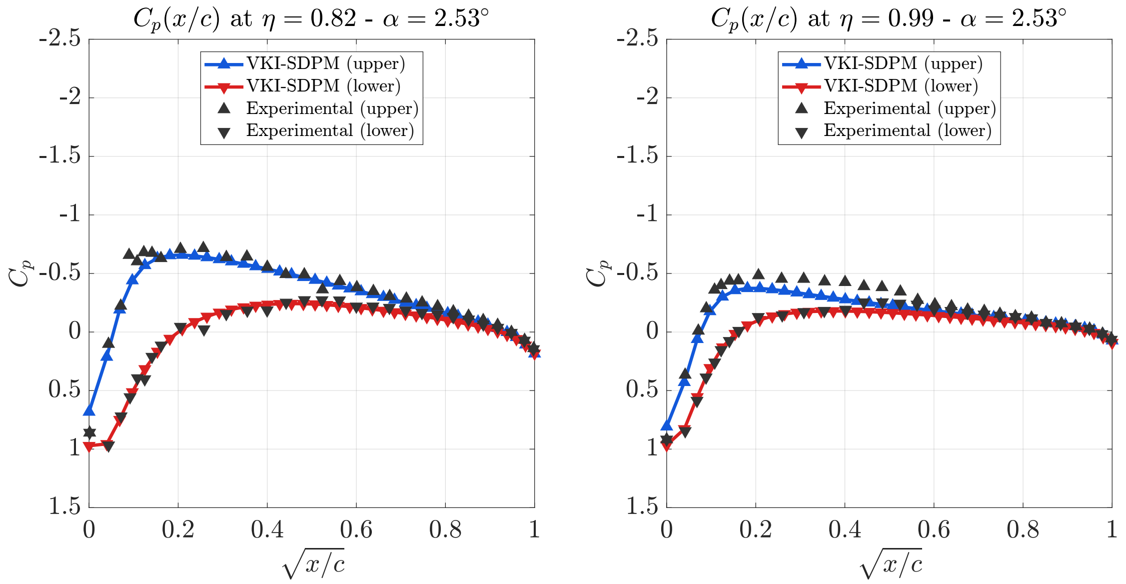
Fig. 7.9 reports the C_p distribution for $\alpha = 2.53°$ and $q_\infty = 1.47$ kPa. VKI-SDPM correctly reproduces the overall chordwise evolution of the suction and pressure sides of the wing. As expected, limited discrepancies can be detected near the root and tip regions, especially on the leading-edge region of the suction side. At this low incidence, the flow remains attached and is therefore closer to the conditions under which the assumptions underlying VKI-SDPM are valid.



(a) $\eta = 0.05$ **(b)** $\eta = 0.20$

(c) $\eta = 0.48$ **(d)** $\eta = 0.61$

Figure 7.9: Experimental and numerical comparison of C_p distributions for the Yip–Shubert test case at $\alpha = 2.53°$.



(e) $\eta = 0.82$ **(f)** $\eta = 0.99$

Figure 7.9 (continued)

For the test case at $\alpha = 6.75°$ and $q_\infty = 1.47$ kPa, additional numerical results from the original report are available. These were obtained by an inviscid finite-strength surface-singularity method, based on [58], which is conceptually close to VKI-SDPM. Therefore, the two numerical solutions and the experimental measurements are compared in Fig. 7.10.

The agreement among the three datasets is maintained over most of the spanwise stations, although some discrepancies are detectable. These are mainly located on the suction side near the leading edge; however, the two numerical solutions remain relatively consistent with each other. The most noticeable difference is in Fig. 7.10f for $\eta = 0.99$. Here, the tip-flow influence is much more evident than in the low-incidence case and it results in an increased suction peak and a secondary peak towards the trailing edge. The test case at $\alpha = 10.97°$, reported in App. D.1.3, shows a similar pressure distribution at $\eta = 0.99$, but with a greater intensity, consistently with the experimental observations.

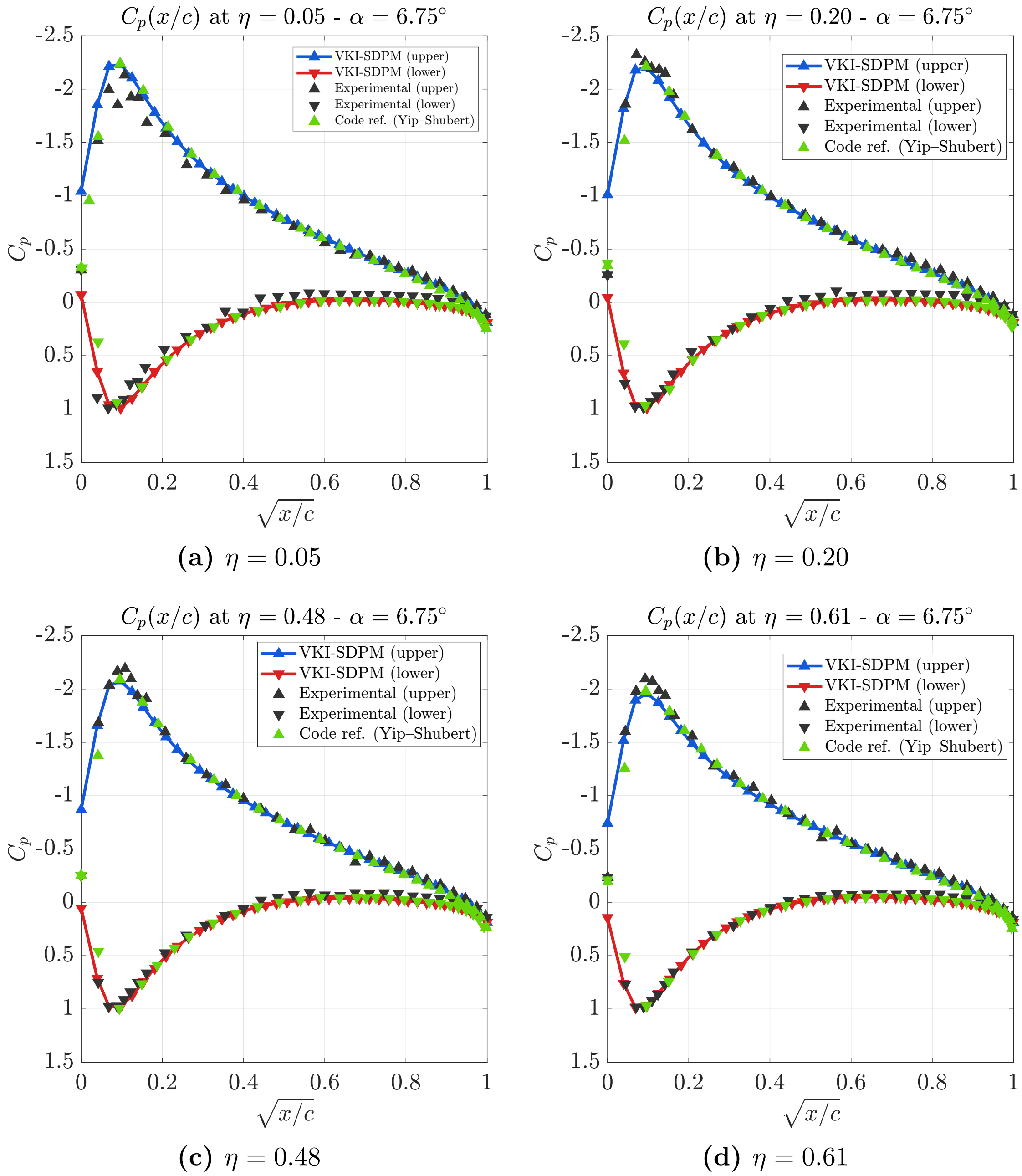
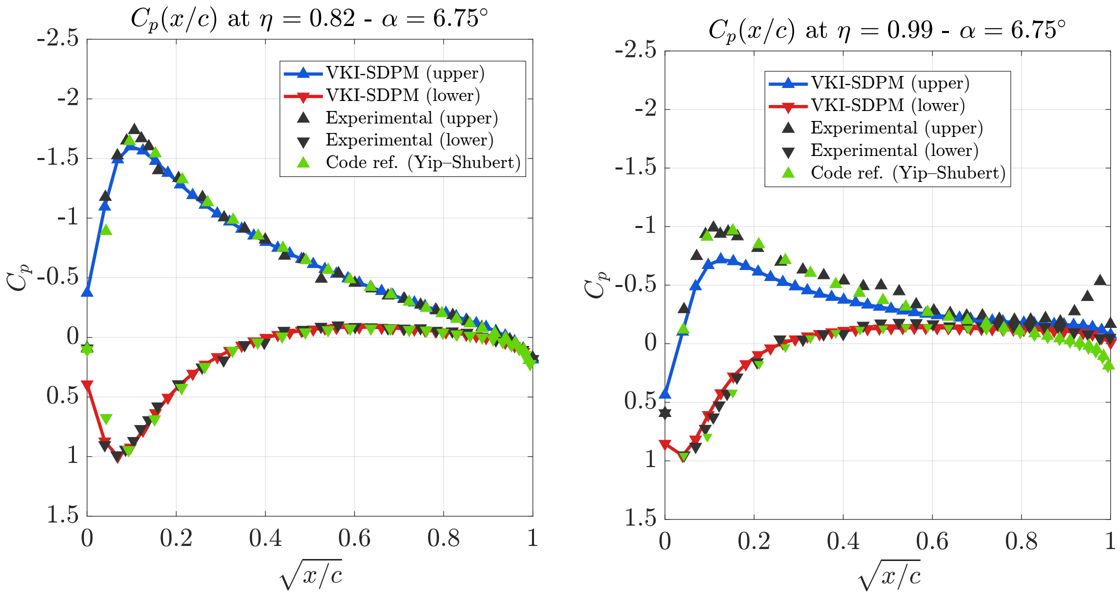


Figure 7.10: Experimental and numerical comparison of C_p distributions for the Yip–Shubert test case at $\alpha = 6.75°$. Numerical results from [9] are indicated by green markers.



(e) $\eta = 0.82$ **(f)** $\eta = 0.99$

Figure 7.10 (continued)

Yip–Shubert Test Case Conclusions

After analysing the results, conclusions about the current test case can be drawn. The validation has progressed from global aerodynamic coefficients to sectional loading and local pressure distributions.

The first quantities considered, $C_N(\alpha)$ and $C_M(\alpha)$, are correctly predicted over most of the investigated range of α. Discrepancies at higher incidences are expected and may be related to the increasing relevance of separation and tip-flow.

The spanwise distributions of $C_m(\alpha)$ and $C_n(\eta)$ provide further insight into where and when these discrepancies become more evident. The load distributions across the span are correctly predicted, especially for the test case at $\alpha = 6.75°$ and $q_\infty = 1.47$ kPa. The spanwise results also show discrepancies near the root and tip regions. The former are associated with the presence of the reflection plate, whereas the latter are related to tip-flow effects. The improved agreement in the midspan region supports the capability of the solver to reproduce the loading where root and tip effects are less influential.

The pressure coefficient distributions provide a final and more detailed assessment of the solver accuracy. At low incidences, the flow conditions more closely satisfy the assumptions underlying VKI-SDPM and therefore the solver can accurately predict the C_p distributions along the span. At higher incidences, the accuracy decreases but remains consistent with most of the measured data. Localised discrepancies can be detected near the leading edge on the suction side. The consistency with the additional numerical results reported in [9] suggests that the observed discrepancies are not specific to the present VKI-SDPM implementation. These may be attributed to the absence of viscous modelling, to experimental uncertainty, or to a combination of both. Finally, the C_p distribution near the tip, which is strongly influenced by the tip flow, highlights the magnitude of the discrepancies between the experimental measurements and the numerical predictions.

In conclusion, the overall comparison indicates that VKI-SDPM is able to reproduce the main aerodynamic behaviour of a finite unswept wing in the low-subsonic regime. The solver correctly predicts the aerodynamic loads from global to local scales, with higher accuracy when the assumptions of its formulation are more closely satisfied.

7.3 Kolbe–Boltz Test Case (NACA RM A51G31)

The second wing validation case is based on the experimental measurements of Kolbe and Boltz on a swept and tapered wing [10]. This benchmark extends the assessment performed with the Yip–Shubert case to a more demanding geometry. The case is therefore useful for verifying whether VKI-SDPM can reproduce the pressure distribution and aerodynamic loading trends associated with swept and tapered planforms. Sweep modifies the effective chordwise velocity component and redistributes the aerodynamic loading along the span, while taper changes the local chord and therefore the local sectional contribution to the global loads.

Part of the experimental dataset includes operating conditions for which compressibility and viscous effects may become non-negligible. For this reason, the comparison is focused primarily on the conditions that remain compatible with the incompressible-flow assumptions of VKI-SDPM.

7.3.1 Experimental Setup

The experimental model considered in this validation is the swept and tapered semispan wing tested in the Ames 12-foot pressure wind tunnel. The original investigation was aimed at assessing the effects of scale and compressibility on the forces, moments, and pressure distributions of a low-aspect-ratio swept wing. In the present work, only a restricted subset of the dataset is used as a primary validation target, namely the cases for which the flow conditions remain closest to the assumptions of incompressible potential flow.

The model was mounted vertically on a turntable, while the wind-tunnel test-section floor acted as a reflection plane. The forces and moments were transmitted directly to the balance system through the turntable. Additional tests were performed to evaluate the influence of several model–turntable juncture seals, but the results show no significant changes in the aerodynamic characteristics of the wing.

The wing had a leading-edge sweep angle of $\Lambda_{\mathrm{LE}} = 48.5°$, while the locus of the quarter-chord points was swept back by $\Lambda = 45°$. The model had no twist, an aspect ratio of $AR = 3$, and a taper ratio of $\lambda = 0.5$. The airfoil sections normal to the quarter-chord line were NACA 64A010 profiles. The tip was rounded by a half-body with a radius equal to the corresponding half-thickness of the tip section. The experimental setup and planform are schematically illustrated in Fig. 7.11.

Pressure orifices were installed in seven rows located in planes parallel to the plane of symmetry. The orifices were distributed along both the upper and lower surfaces from the leading-edge to approximately 91% of the local chord and were staggered on either side of each station plane. The seven spanwise stations considered in the report are

$$\eta = [0.086,\ 0.195,\ 0.382,\ 0.555,\ 0.707,\ 0.831,\ 0.924]\,, \tag{7.7}$$

where $\eta = y/(b/2)$ is the non-dimensional semispan coordinate, as shown in Fig. 7.12. Because of the staggered arrangement of the pressure orifices, the report notes that a slight saw-tooth variation may appear in the plotted chordwise pressure distributions, especially in regions with large spanwise pressure gradients. For this reason, the pressure distributions reported by Kolbe and Boltz were represented by smooth mean curves faired through the measured pressure-coefficient values.

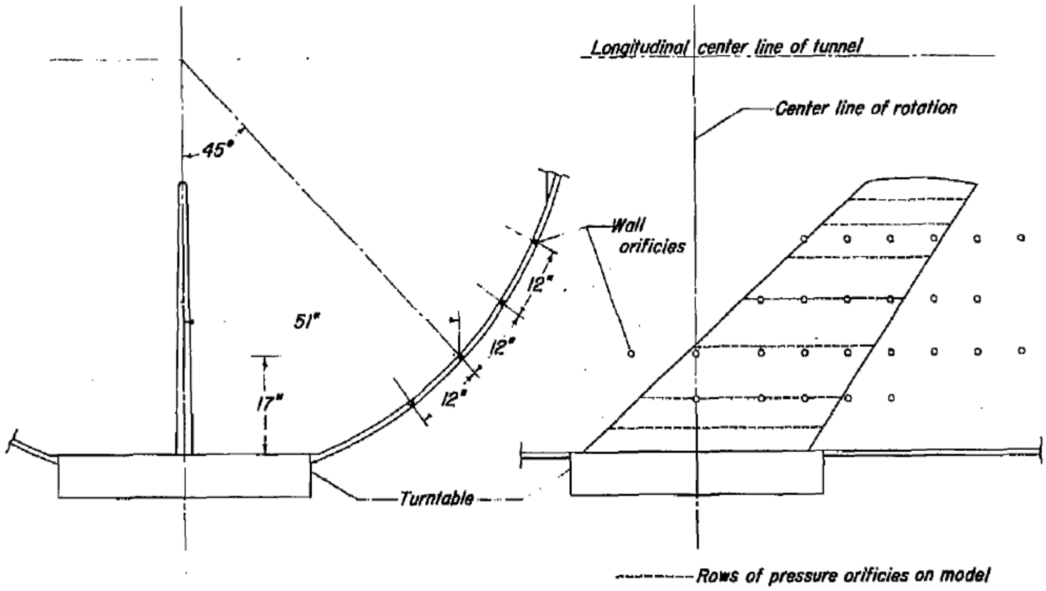


Figure 7.11: Schematic of the experimental setup used by Kolbe and Boltz [10].

The experimental campaign covered several combinations of Mach and Reynolds numbers. Chordwise pressure distributions at the seven spanwise stations were measured simultaneously with the total lift, drag, and pitching moment.

In the present validation, the low-Mach cases are used as the primary comparison. Higher-Mach cases above the compressibility threshold are retained as diagnostic comparisons to illustrate the possible loss of agreement when compressibility effects become increasingly relevant.

The adopted experimental conditions for the numerical simulations are summarised in Tab. 7.2.

7.3.2 Numerical Setup

The VKI-SDPM numerical model reproduces the full-span configuration with the geometry generated from the reported aspect ratio, taper ratio, and sweep. The

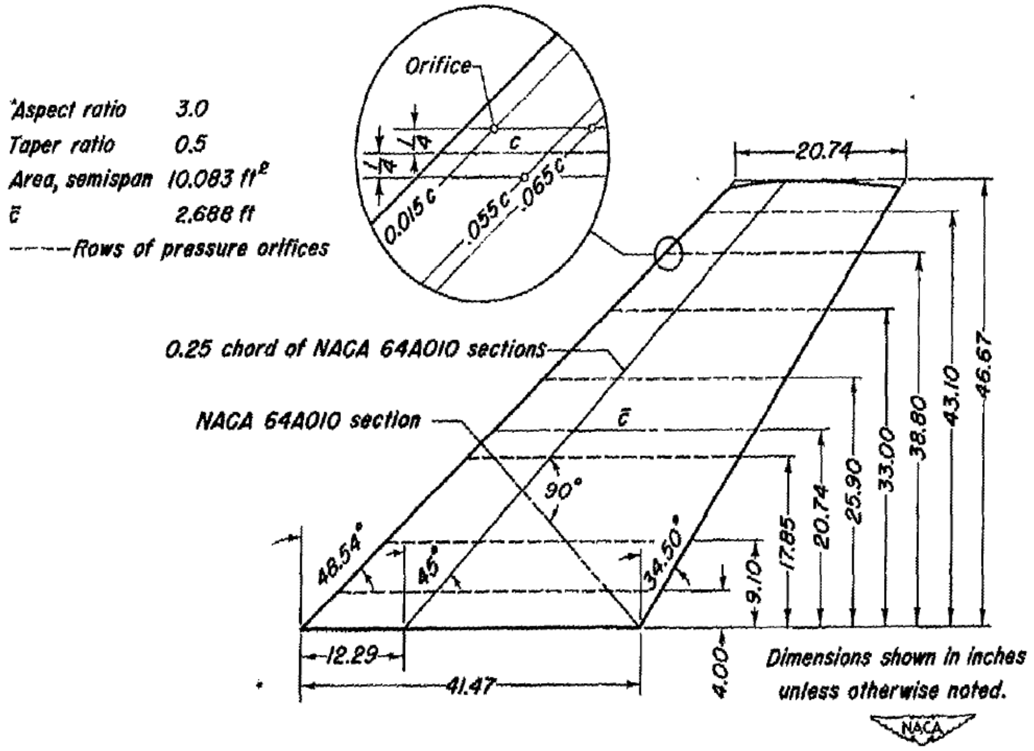


Figure 7.12: Planform schematic of the Kolbe–Boltz wing [10]. Dimensions are given in imperial units.

Table 7.2: Kolbe–Boltz experimental conditions and role in the VKI-SDPM validation.

$Re \cdot 10^{-6}$	M	Simulated	$C_p(x/c)$ Reported in
4	0.08	✓	/
4	0.25	✓	Sec. 7.3.4, App. D.2.2
4	0.40	✓	App. D.2.3
4	0.60	✓	/
4	≥ 0.80	✗	/
6	0.25	✗	/
8	0.08	✓	App. D.2.1
8	≥ 0.25	✗	/
12	0.25	✗	/
18	0.25	✗	/

mean aerodynamic chord is computed analytically from the taper ratio and the root chord as in [59],

$$\bar{c} = \frac{2}{3} c_r \frac{1 + \lambda + \lambda^2}{1 + \lambda}. \tag{7.8}$$

This value is used as the reference chord for the wake length and for the moment coefficient normalisation.

The surface mesh is generated using $N_c = 80$ chordwise panels and $N_r = 20$ spanwise panels, as shown in Fig. 7.13. The chordwise discretisation uses sine spacing, while the spanwise discretisation uses cosine spacing. The same mesh is used for all selected operating conditions. This panel density is selected consistently with the baseline mesh identified for the straight wing case in Sec. 6.2; although, because the present geometry is swept and tapered, a dedicated sensitivity analysis would be required to fully exclude discretisation effects.

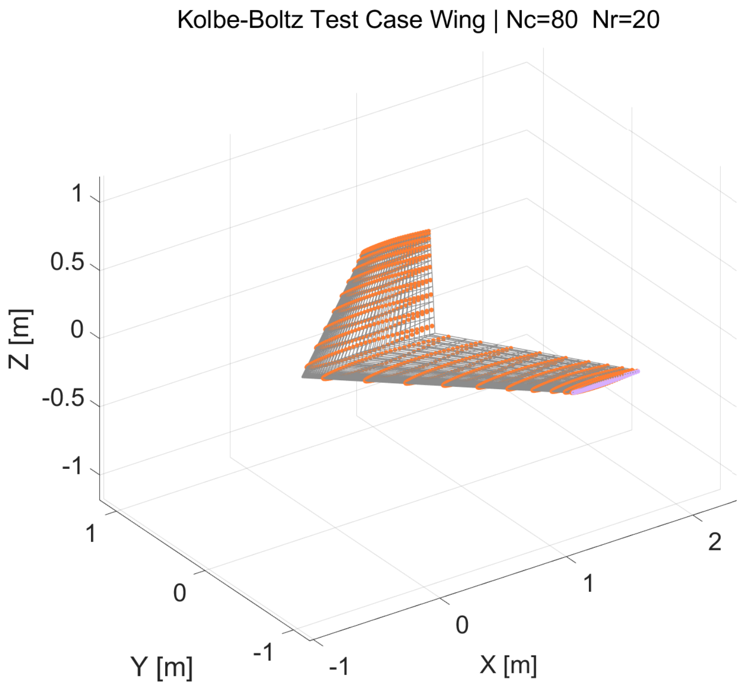
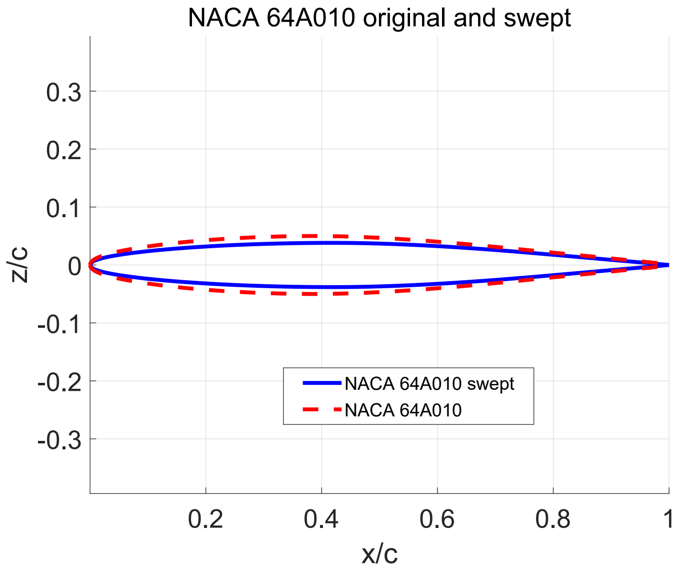
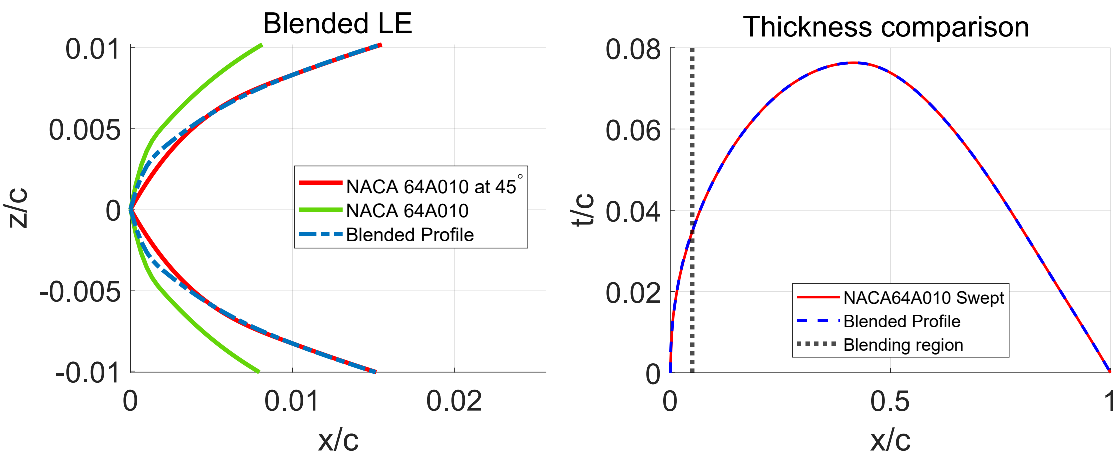


Figure 7.13: Surface mesh for the Kolbe–Boltz validation campaign.

The original report provides both the NACA 64A010 section normal to the quarter-chord line and the corresponding profile defined in planes parallel to the plane of symmetry. These two descriptions lead to different chordwise profiles, as shown in Fig. 7.14a. However, the tabulated geometry for the parallel version contains a limited number of points in the leading-edge region. For this reason, a local leading-edge blending procedure was applied to obtain a smoother profile while preserving the original thickness distribution as closely as possible. The resulting profile and the corresponding thickness comparison are shown in Fig. 7.14.



(a) Comparison between the NACA 64A010 and the corresponding section parallel to the plane of symmetry.



(b) Leading-edge blending.

(c) Thickness comparison.

Figure 7.14: NACA 64A010 airfoil blending and thickness comparison.

The wake is modelled as a prescribed wake, with a downstream extension set to

$$l_{\mathrm{w}} = K_{\mathrm{w}}\, c_{\mathrm{ref}} = 13\, \bar{c}, \tag{7.9}$$

where $c_{\text{ref}} = \bar{c}$ is the mean aerodynamic chord. This value was sufficient to reduce the residual $\mathcal{R}(\mu)$, defined in Eq. (7.3), below the numerical threshold of $\varepsilon_\mu = 10^{-4}$.

The prescribed-wake assumption is considered appropriate for the present benchmark based on the results obtained in Sec. 6.3. Because the wake geometry is prescribed, neither a vortex-core model nor a viscous core-growth model is required for the present prescribed-wake calculation. An illustration of the prescribed wake geometry is reported in Fig. 7.15.

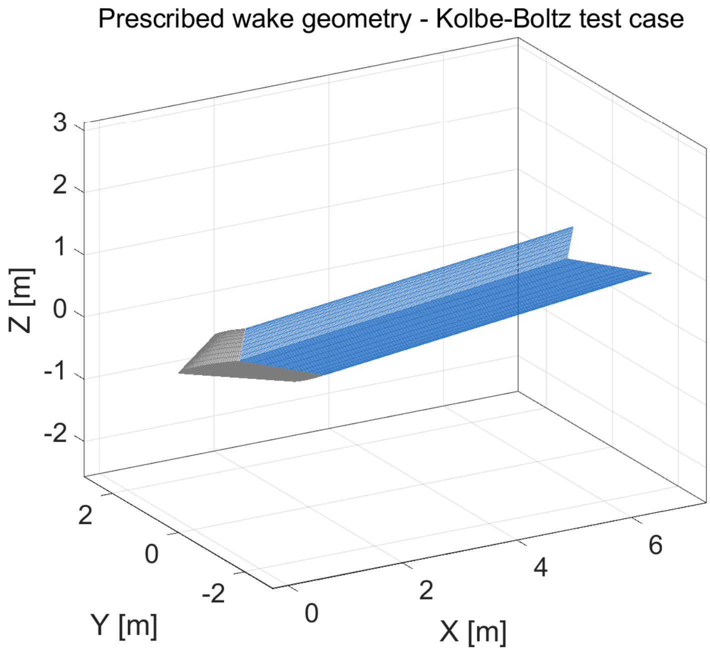


Figure 7.15: Prescribed wake geometry for the Kolbe–Boltz test case at $M = 0.25$ and $\alpha_u = 6°$.

7.3.3 Comparison Quantities and Corrections

The validation is based on the comparison of pressure coefficient distributions, global aerodynamic coefficients, and, where available, sectional or spanwise loading distributions. The main global trends are examined in terms of $C_L(\alpha)$, $C_L(C_D)$, and $C_L(C_M)$. Here, C_L and C_D are the total lift and drag coefficients, respectively, while C_M is the total pitching moment coefficient evaluated about the quarter-chord point of the wing mean aerodynamic chord, consistently with the convention adopted in [10].

The experimental pressure coefficient distributions are compared at the spanwise stations reported in Eq. (7.7) by linearly interpolating the numerical solution between neighbouring spanwise stations.

The original report distinguishes between corrected and uncorrected angles of attack, denoted by α and α_u, respectively. Corrections for tunnel-wall interference due to lift are applied to the angle of attack and drag coefficient following [60], with the theoretical span-loading charts from [61]. The following corrections are then applied:

$$\Delta\alpha = 0.769\ C_L, \tag{7.10}$$

$$\alpha = 0.99\ \alpha_u + \Delta\alpha, \tag{7.11}$$

$$\Delta C_D = 0.0109\ C_L^2, \tag{7.12}$$

Additional tabulated tare corrections were subtracted from the drag coefficient to compensate for the forces acting on the exposed surface of the turntable. No correction is applied to the pitching moment data [10].

Constriction corrections were also applied in [10] to the freestream Mach number and dynamic pressure following [57]. For the low-subsonic cases considered in the present validation, these corrections remain small and are not explicitly considered.

The pressure data and the coefficients derived from pressure integration are reported in the original tables as functions of the uncorrected angle of attack α_u. Therefore, in the present work, the same angle-of-attack convention is adopted. However, the VKI-SDPM simulations are performed by imposing the corresponding corrected angle of attack α, computed from Eq. (7.11) using the experimental value of C_L associated with each condition.

In [10], aeroelastic effects were assumed to be negligible on the basis of the structural rigidity of the model and previous static-loading tests.

For the selected cases in Tab. 7.2, the simulated angles of attack are those listed in Eq. (7.13). In the main text, pressure distributions are reported only for a subset of these angles, while additional pressure-distribution comparisons are provided in App. D.2.

$$\alpha_u = [1°,\ 2°,\ 3°,\ 4°,\ 6°,\ 8°,\ 10°,\ 12°,\ 14°,\ 16°,\ 18°,\ 20°,\ 22°,\ 24°]. \tag{7.13}$$

7.3.4 Results and Discussion

The Kolbe–Boltz validation results are discussed from global quantities to local distributions. Since VKI-SDPM is based on an inviscid formulation, Reynolds-number effects cannot be captured. The comparison is therefore mainly performed at the fixed Reynolds number of $4 \cdot 10^6$, while the Mach sweep is used to assess the range over which the incompressible assumption remains valid.

All the reported results were obtained after convergence of the steady solution. Convergence was considered achieved when the residual $\mathcal{R}(\mu)$, defined in Eq. (7.3), dropped below the numerical threshold of $\varepsilon_\mu = 10^{-4}$. This is reflected in the time histories of C_L and C_D, as illustrated for the case of $M = 0.25$ and $\alpha_u = 6°$ in Figs. 7.16 and 7.17.

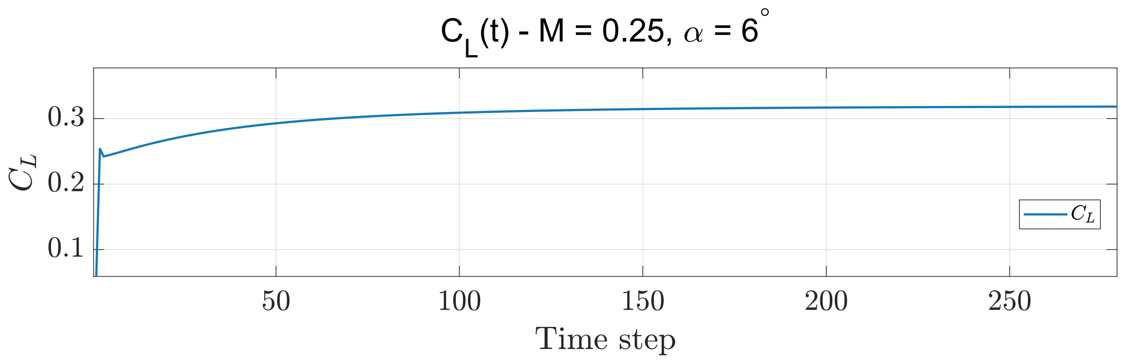


Figure 7.16: C_L history for $M = 0.25, \alpha_u = 6°$.

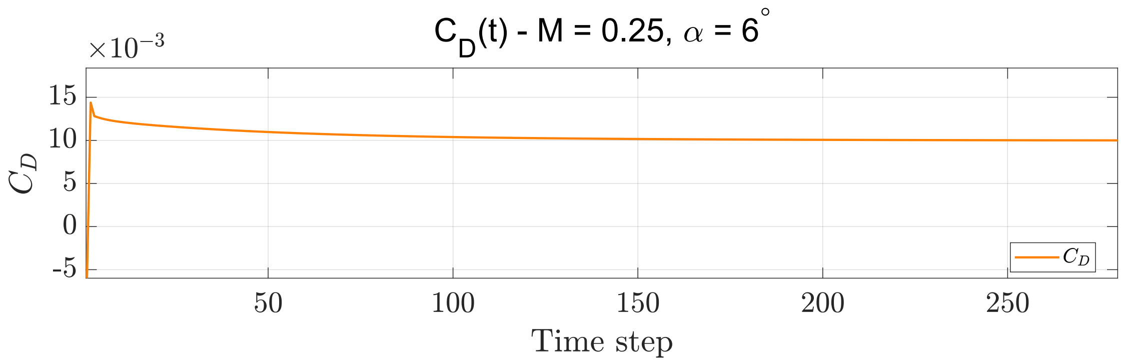
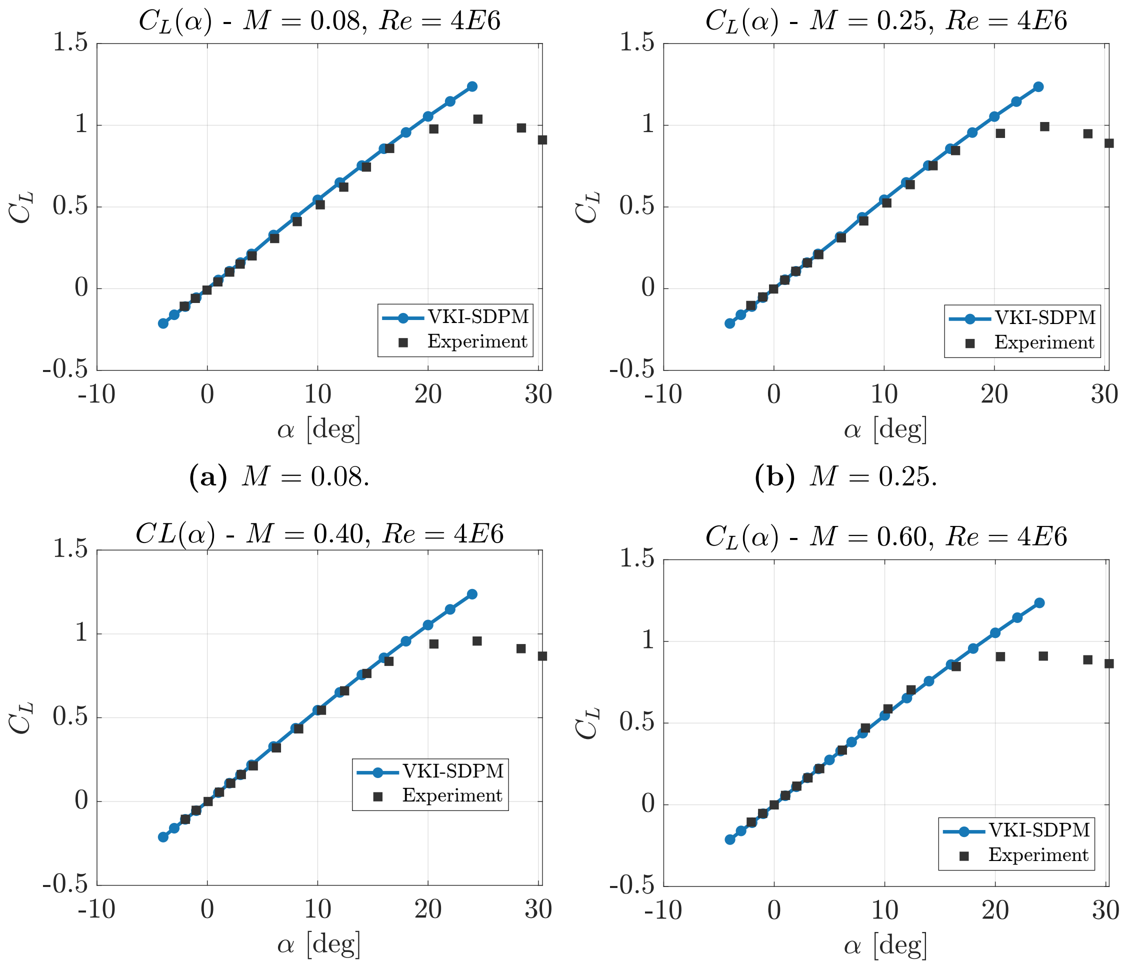


Figure 7.17: C_D history for $M = 0.25, \alpha_u = 6°$.

Global Aerodynamic Coefficients

Figure 7.18 shows $C_L(\alpha)$ for four Mach numbers. The low-Mach cases, $M = 0.08$ and $M = 0.25$, represent the most relevant validation range for the present incompressible formulation.

At low incidence, the numerical and experimental lift curves show the same general linear trend. This result indicates that VKI-SDPM captures the dominant inviscid loading mechanism associated with the swept and tapered planform. As the angle of attack increases, viscous and separated-flow effects become progressively more relevant, eventually leading to stall. In [10], Kolbe and Boltz report that no apparent leading-edge separation was observed for $C_L < 0.3$ at any tested M and Re, while at higher lift coefficients the onset of leading-edge separation and reattachment over the outer wing sections became relevant. The increase in Mach number has a relatively small effect, which may partly be attributed to the wing sweep and the resulting reduction in the effective flow component normal to the leading-edge.



(a) $M = 0.08$.

(b) $M = 0.25$.

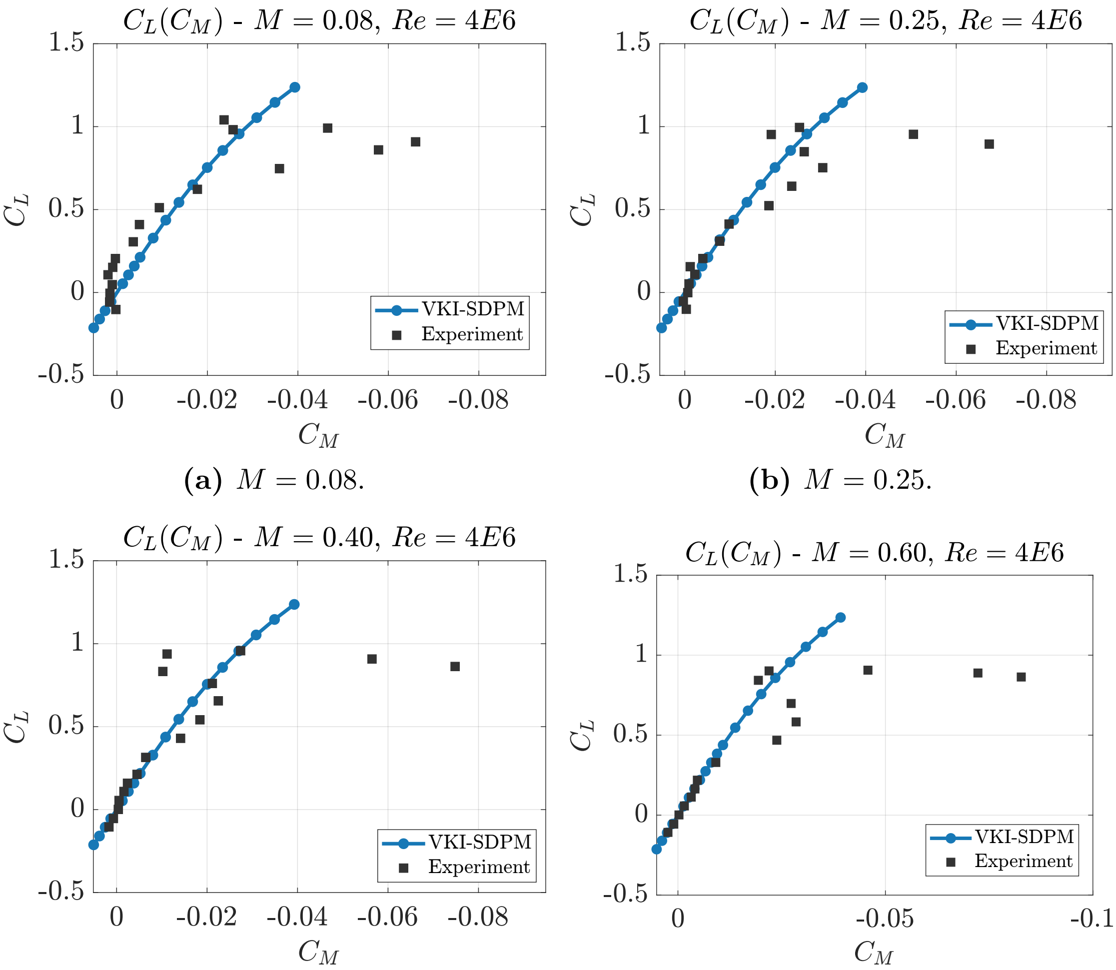
(c) $M = 0.40$.

(d) $M = 0.60$.

Figure 7.18: $C_L(\alpha)$ for $M = 0.08,\ 0.25,\ 0.40,\ 0.60$ at $Re = 4 \cdot 10^6$.

The pitching moment behaviour is reported in Fig. 7.19. Improvements in the comparison for low values of C_L can be observed at higher Mach numbers. This should not be interpreted as a direct prediction of compressibility effects by

VKI-SDPM. Rather, it is likely the result of a partial compensation between the incompressible inviscid prediction and the compressible effects that redistribute pressure and move the aerodynamic centre. However, as C_L and Mach number increase, the experimental trend increasingly departs from the numerical prediction. This behaviour may be related to the increase in relevance of separation effects.



(a) $M = 0.08$.

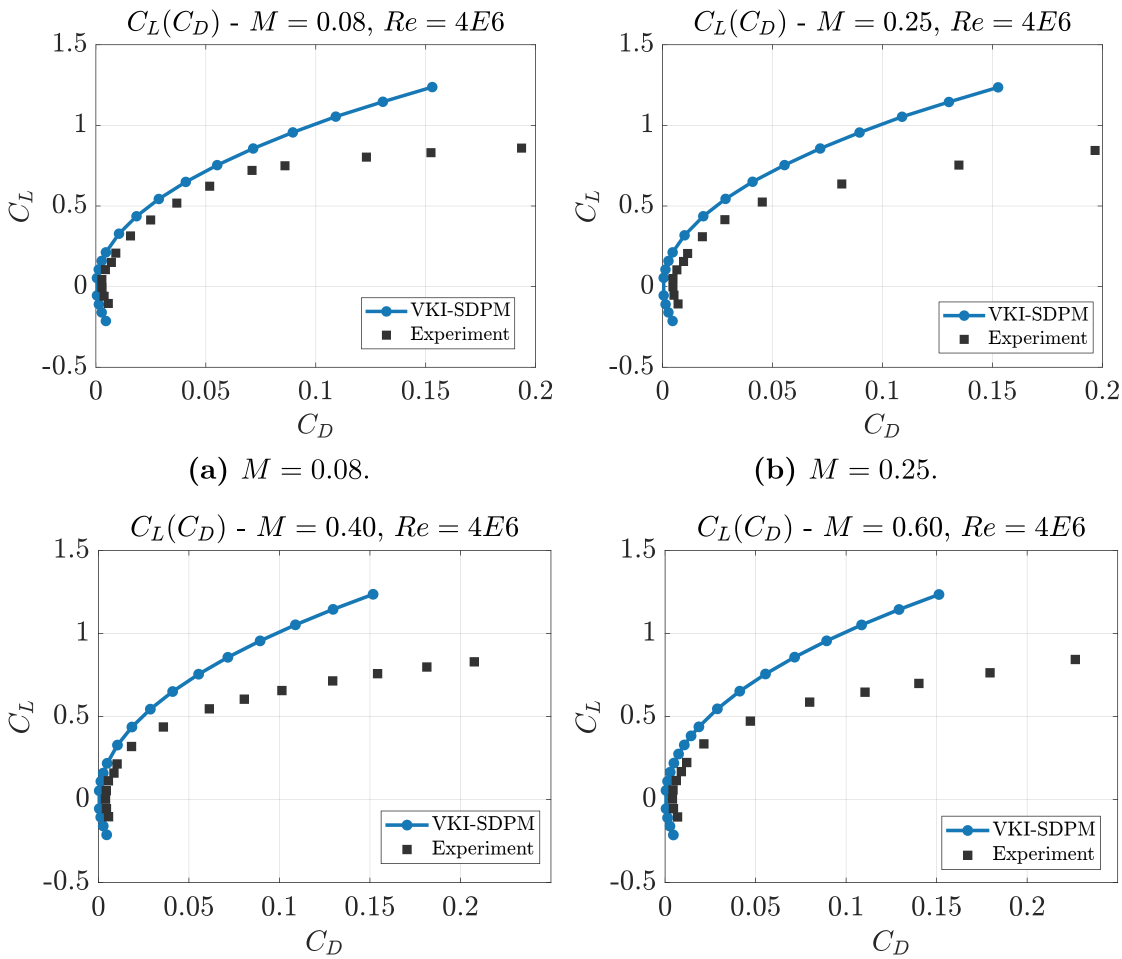
(b) $M = 0.25$.

(c) $M = 0.40$.

(d) $M = 0.60$.

Figure 7.19: $C_L(C_M)$ for $M = 0.08,\ 0.25,\ 0.40,\ 0.60$ at $Re = 4 \cdot 10^6$.

Finally, Fig. 7.20 reports the aerodynamic polar of the wing for different Mach numbers. The experimental drag includes viscous, separation, and possible compressibility contributions, whereas VKI-SDPM does not include these contributions. As shown in Fig. 7.20, the inviscid curve follows the viscous one for low values of C_L before separation effects become significant. The ever-present offset between the two curves is associated with the viscous and profile-drag contributions that are absent from the inviscid formulation. However, the $C_L(C_D)$ comparisons show similar trends in the low Mach and low C_L cases. Therefore, when compressibility and viscous effects remain limited, VKI-SDPM captures the main aerodynamic trends.



(a) $M = 0.08$.

(b) $M = 0.25$.

(c) $M = 0.40$.

(d) $M = 0.60$.

Figure 7.20: $C_L(C_D)$ for $M = 0.08,\ 0.25,\ 0.40,\ 0.60$ at $Re = 4 \cdot 10^6$.

Spanwise Loading

The spanwise loading distribution is then examined. This comparison is particularly relevant for the Kolbe–Boltz case because sweep and taper both affect the redistribution of aerodynamic loading along the span.

Figure 7.21 reports the spanwise distribution of the sectional normal force coefficient C_n at the fixed Mach number $M = 0.25$ and for $\alpha_u = 2°,\ 4°,\ 6°$. The comparison is performed at the same spanwise locations used in the experimental pressure measurements.

The numerical results reproduce the main spanwise trend of the experimental loading distribution for all three incidences. The best agreement is obtained around $\eta \approx 0.7$ for all incidences, while the agreement tends to deteriorate towards the tip. Better overall agreement is obtained at lower incidences. The overprediction by VKI-SDPM is expected, since viscous effects are not modelled and the flow over the experimental wing sections may be affected by separation. In addition, the modifications introduced in the numerical representation of the leading-edge region may also play a role in local discrepancies. Nevertheless, the discrepancies observed in Fig. 7.21 are consistent with the expected limitations of the present formulation.

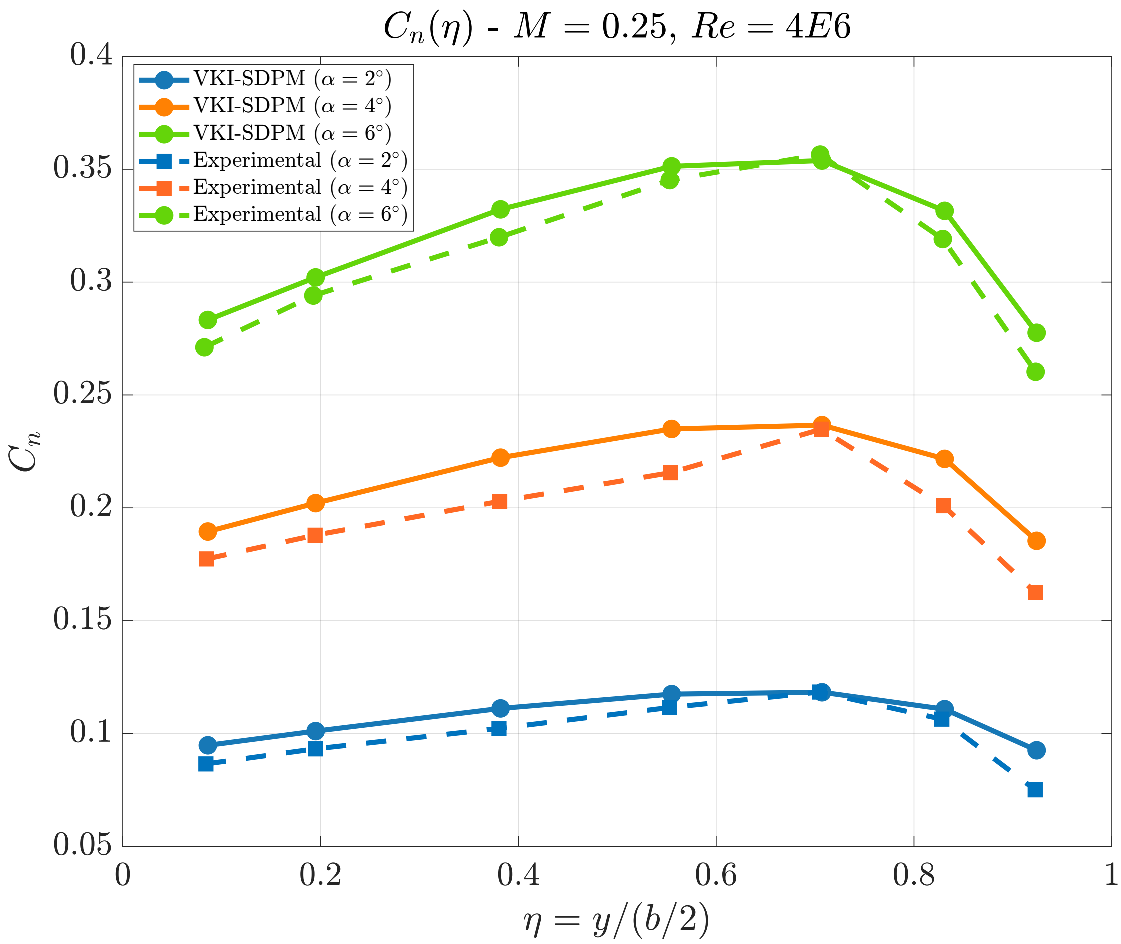


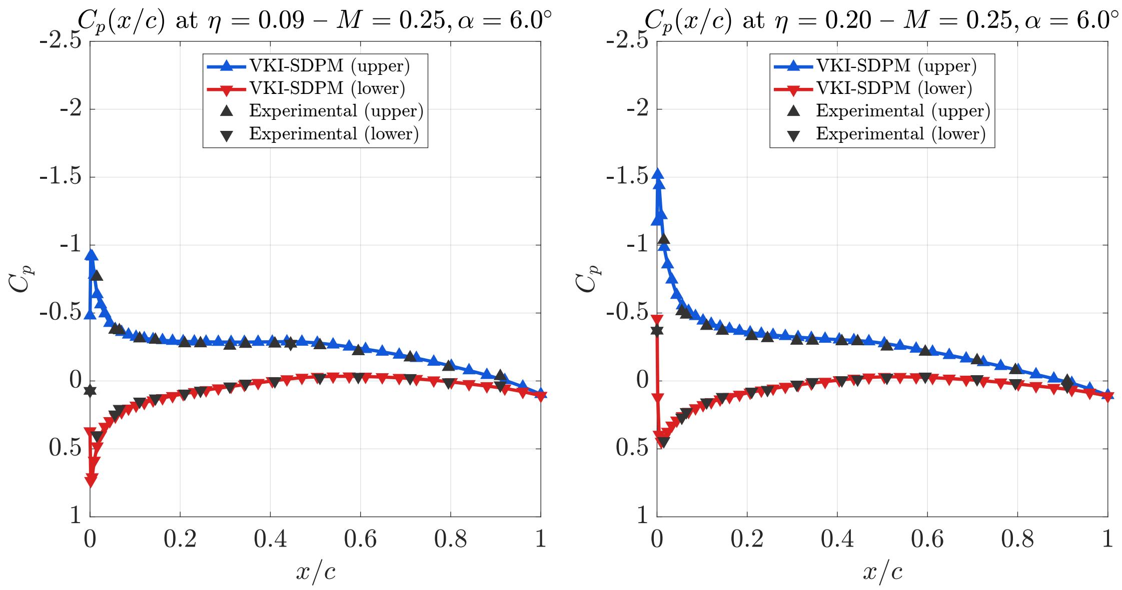
Figure 7.21: C_n spanwise distribution comparisons at $M = 0.25$ for $\alpha_u = 2°$ (blue), $\alpha_u = 4°$ (orange), and $\alpha_u = 6°$ (green).

Pressure Coefficient Distributions

For the pressure-coefficient distributions, two incidences are selected at a fixed Mach number. The first corresponds to a moderate-incidence case with $\alpha_u = 6°$, representative of the end of the nearly linear aerodynamic regime. The second corresponds to a higher-incidence case with $\alpha_u = 10°$, where the pressure field is increasingly affected by leading-edge separation. Additional C_p distributions are reported in App. D.2.

Figure 7.22 reports the pressure coefficient distributions for the case at $M = 0.25$ and $\alpha_u = 6°$. The comparison shows that VKI-SDPM captures the dominant chordwise pressure behaviour over most of the span, although the suction peak is generally overpredicted, with the discrepancy becoming more pronounced towards the tip.

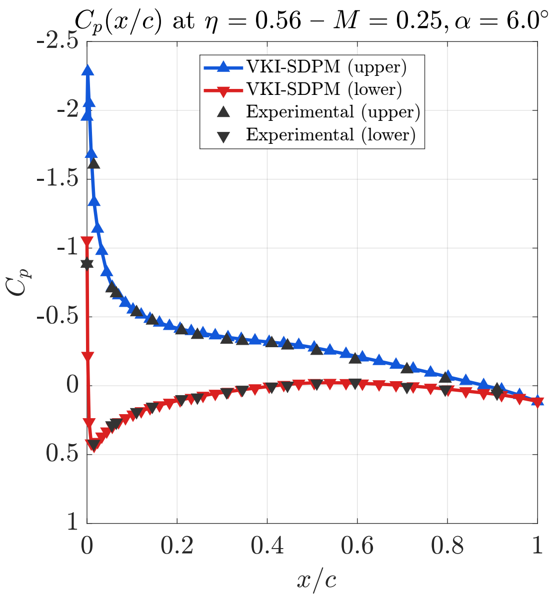
Figure 7.18b shows that for $\alpha_u = 6°$ the corresponding C_L is above 0.3. Therefore, the experimental condition is already within the range where leading-edge separation and reattachment may influence the pressure distribution [10]. The resulting viscous effects tend to reduce the suction peak relative to the inviscid prediction [5]. Lower incidences reduce the discrepancy in this region, as reported in App. D.2.2 for $M = 0.25$ and $\alpha_u = 4°$.



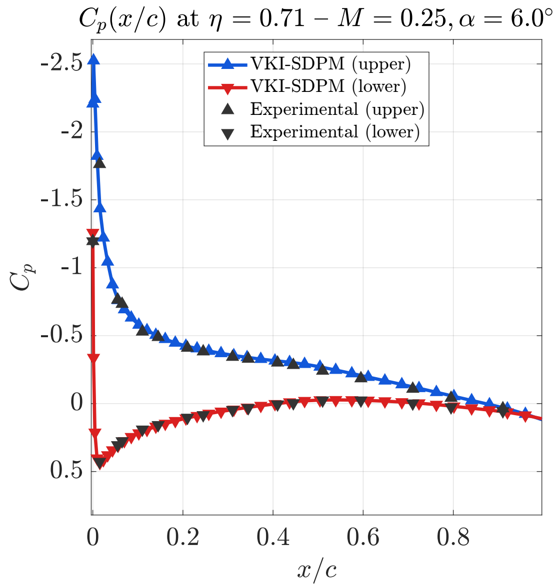
(a) $\eta = 0.09$

(b) $\eta = 0.20$

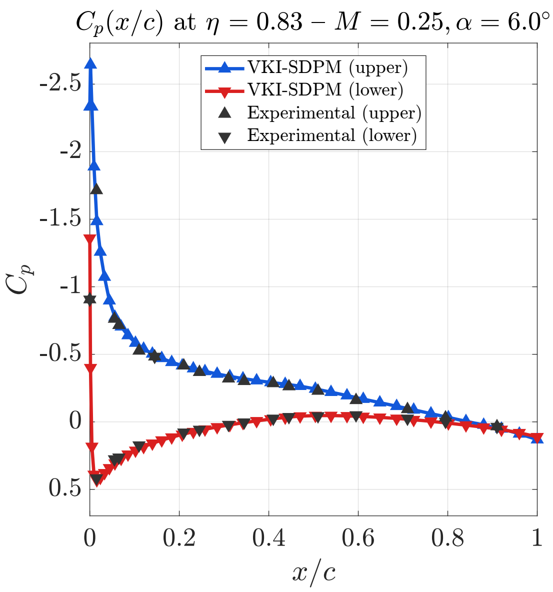
Figure 7.22: Experimental and numerical comparison of C_p distributions for the Kolbe–Boltz test case at $M = 0.25$, $\alpha_u = 6°$.



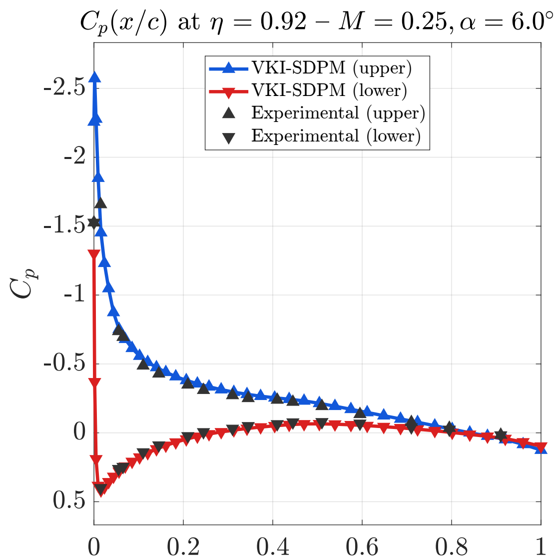
(c) $\eta = 0.56$



(d) $\eta = 0.71$



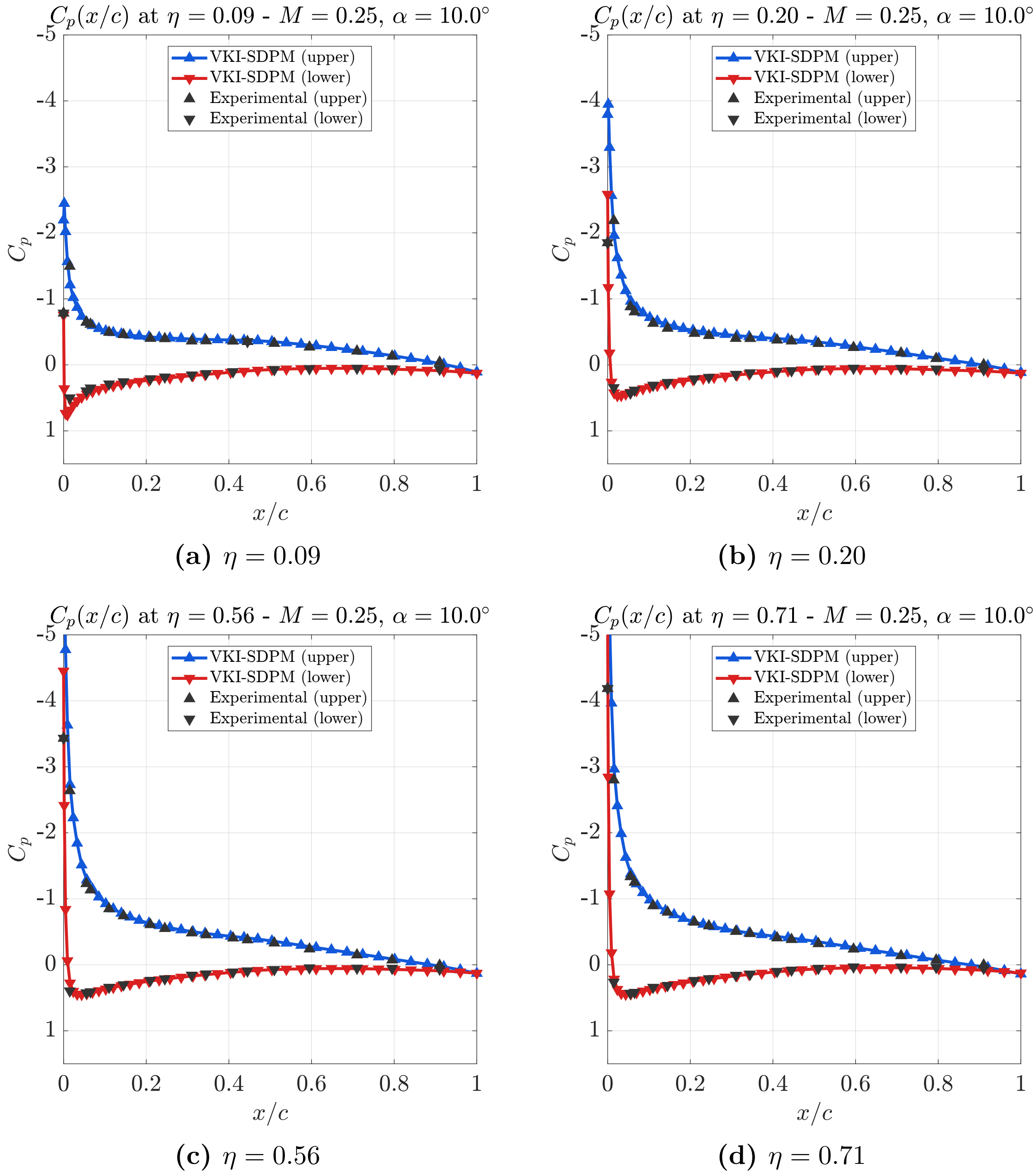
(e) $\eta = 0.83$



(f) $\eta = 0.92$

Figure 7.22 (continued)

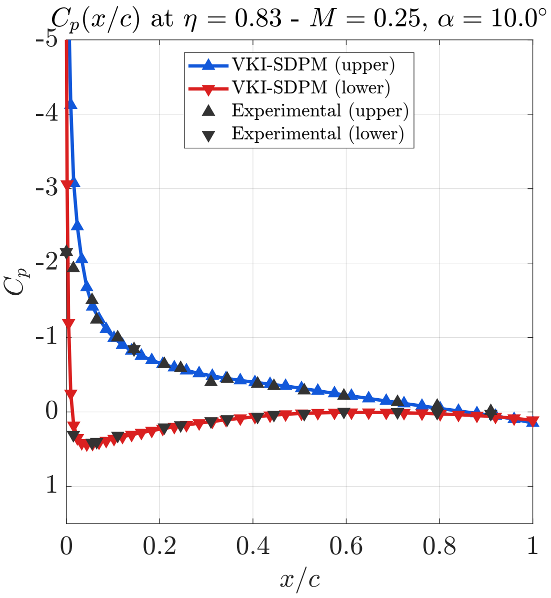
The higher-incidence case is reported in Fig. 7.23. Here, the overprediction at the LE becomes more evident. In addition, Fig. 7.23f shows a slightly delayed pressure recovery near the trailing edge, while VKI-SDPM predicts a more regular recovery. These discrepancies are expected, since separation effects increase with α.



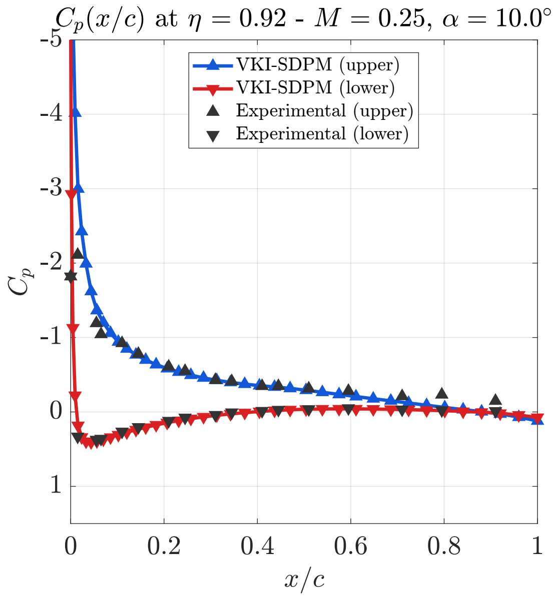
(a) $\eta = 0.09$ **(b)** $\eta = 0.20$

(c) $\eta = 0.56$ **(d)** $\eta = 0.71$

Figure 7.23: C_p distributions for the Kolbe–Boltz test case at $M = 0.25,\ \alpha_u = 10°$.



(e) $\eta = 0.83$



(f) $\eta = 0.92$

Figure 7.23 (continued)

Kolbe–Boltz Test Case Conclusions

The Kolbe–Boltz validation extends the assessment of VKI-SDPM to swept and tapered wings. This benchmark is more demanding than the unswept-wing case because the aerodynamic loading is influenced not only by finite-wing effects but also by the three-dimensional loading redistribution associated with sweep and taper

In this context, the lift coefficient and the sectional loading distribution in the low-Mach-number range represent meaningful validation quantities. The agreement obtained for these quantities indicates that the solver is able to capture both the integral loads and the main distributed-loading trends associated with the wing geometry.

The discrepancies increase as the angle of attack and Mach number increase, consistently with the growing influence of compressibility and leading-edge separation. In particular, the overprediction observed in the spanwise loading may be related to the excessive suction predicted near the leading-edge, as seen in the C_p distributions.

Viscous effects are particularly evident in the drag-polar comparison, where C_D is affected by the absence of friction and profile-drag modelling. Nevertheless, the low-lift trends remain qualitatively consistent with the experimental behaviour.

The pressure-coefficient distributions confirm that the solver captures the main chordwise pressure behaviour. However, as anticipated above, visible discrepancies appear in the suction-peak region. These differences are associated with the limitations of the inviscid, incompressible formulation adopted in VKI-SDPM. A possible improvement would be the coupling of the panel solver with a viscous model capable of resolving, or at least accounting for, boundary-layer development. Such an extension would allow the external inviscid solution to be corrected in regions where viscous effects modify the leading-edge suction level.

In conclusion, the Kolbe–Boltz test case confirms that VKI-SDPM can reproduce the dominant inviscid features of a swept and tapered wing, while also highlighting the physical limits of the present formulation when the experimental flow departs from the assumptions of incompressible, inviscid potential flow.

7.4 NREL UAE Phase VI Test Case

The third validation case is based on the NREL Unsteady Aerodynamics Experiment (UAE) Phase VI [11]. This benchmark involves a wind-turbine rotor and is therefore used to assess the capability of VKI-SDPM to handle rotating lifting surfaces, spanwise-varying blade geometry, and wake-induced three-dimensional effects.

The original purpose of the NREL UAE campaign was to provide accurate quantitative aerodynamic and structural measurements for a full-scale horizontal-axis wind turbine in a controlled wind-tunnel environment [11]. The wind-tunnel test was designed to separate rotor aerodynamic effects from the inflow anomalies typically encountered during field operation. For this reason, the dataset provides a valuable reference for numerical validation.

7.4.1 Experimental Setup

The tests were conducted in the NASA Ames 24.4 m by 36.6 m open-loop test section where the wind turbine model was mounted with the hub located at the tunnel centreline, as illustrated in Fig. 7.24.

The Phase VI rotor is a two-bladed, stall-regulated wind turbine with a nominal diameter of 10.058 m when equipped with the standard tip. The blades were tapered and twisted, with the aerodynamic portion of the span based on the NREL S809 airfoil from [62], illustrated in Fig. 7.25.

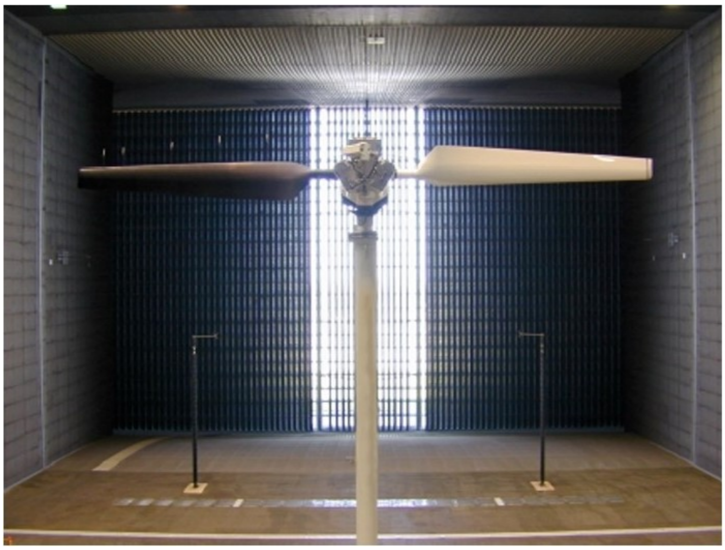


Figure 7.24: View of the full-scale wind-turbine model tested in the NASA Ames wind tunnel [11].

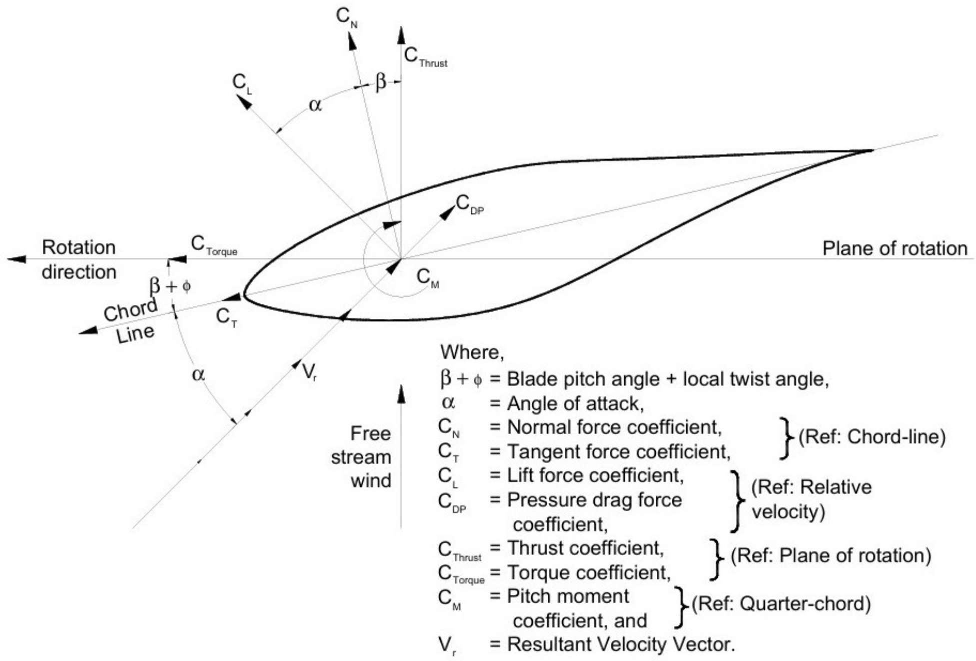


Figure 7.25: S809 airfoil section and aerodynamic-coefficient convention adopted in the NREL report [11].

Blade surface pressures were measured at five primary radial stations, located at 30%, 46.6%, 63.3%, 80%, and 95% of the span of the instrumented blade. At each of these stations, full-chord pressure-tap distributions were available. The taps were concentrated near the leading edge in order to better resolve the suction peak and the most active region of the pressure distribution.

The experimental database also includes measurements obtained with probe extensions, providing local flow angle, spanwise flow angle, and dynamic pressure at several radial stations. However, these data may be affected by probe interference, range limitations, and boom-wake effects, especially in yawed operation. Therefore, the present validation is based on the dataset named Sequence S, where the turbine was tested in the upwind, rigid, zero-cone configuration with the probes removed.

7.4.2 Numerical Setup

The numerical model reproduces only the aerodynamic portion of the rotor. In this region, the blade is modelled by interpolating the airfoil profile coordinates and the tabulated radial distributions of chord and twist from [11]. The cylindrical and transition regions are not modelled. Moreover, the tower and nacelle are not included in the numerical model. This modelling choice isolates the rotor-blade aerodynamic response but also implies that any effects associated with the excluded experimental components cannot be reproduced. In this context, results obtained under yawed inflow are the most affected, since yaw introduces azimuthally varying local flow conditions and may amplify instrumentation-related uncertainties.

The computational mesh is generated by discretising each blade in the chordwise and spanwise directions. Spanwise clustering is applied near the blade root and tip by a cosine spacing law, while chordwise clustering is applied near the leading and trailing edges by a sine spacing law. The discretisation selected after the mesh sensitivity analysis in Sec. 6.2 is $N_c = 80$ and $N_r = 20$, where N_c is the number of chordwise panels and N_r is the number of spanwise panels per blade. The blade mesh is reported in Fig. 7.26.

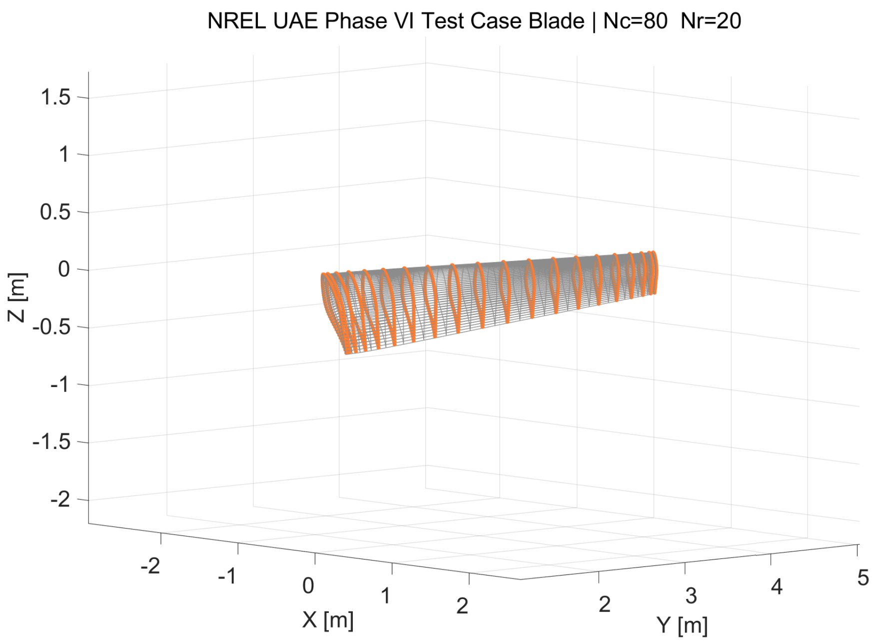


Figure 7.26: Computational mesh for the NREL UAE Phase VI blade.

For the validation cases, quantities such as rotational speed, blade pitch angle, air density, airspeed and reference sectional dynamic pressure are set to the mean measured values reported in each results dataset file. The blade pitch angle is imposed following the convention adopted in the report, where pitch is defined at the blade tip and is positive toward feather. The pitch and twist axes coincide and are located at 30% of the chord, as reported in [11].

The rotor simulation is performed using a free-wake time-marching solution. Based on the sensitivity analysis in Sec. 6.3, $N_w = 8$ revolutions are used, which are sufficient to reach a periodic or statistically steady behaviour. The azimuthal step is set to $\Delta\psi = 15°$. A representation of the rotor wake in the yawed configuration after seven revolutions is reported in Fig. 7.27.

The induced velocity is regularised using a Lamb–Oseen core model, combined with a Squire core-growth law. An initial core radius is set to $r_{c,0} = 0.01\ c_{80\%}$, where $c_{80\%}$ is the chord at 80% of the blade span. The value of the constant a_1 for the Squire law is set to $2 \cdot 10^{-4}$, consistent with the formulation reported in [19] and [27].

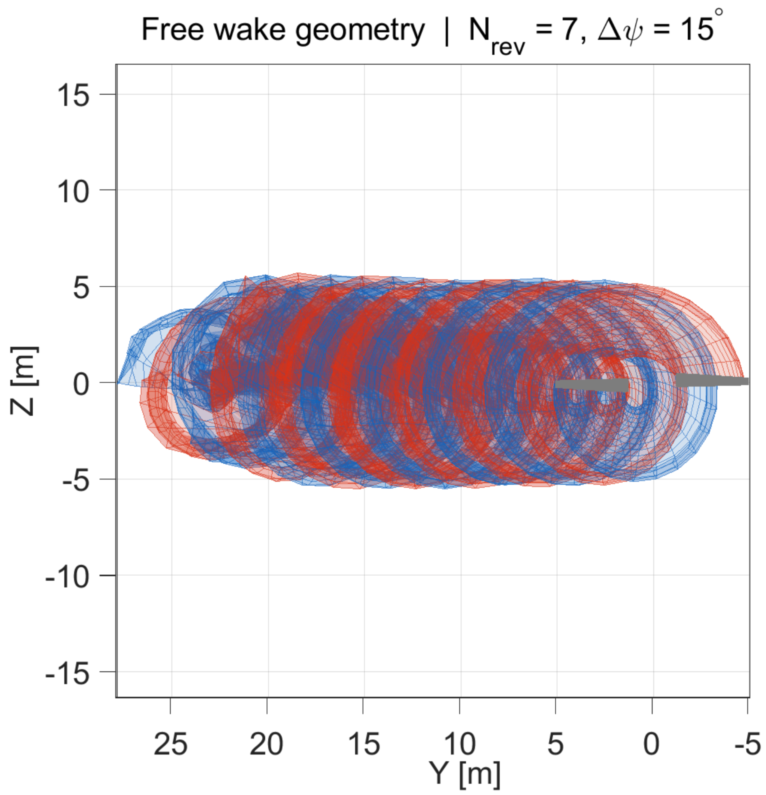


Figure 7.27: Front view of the NREL UAE Phase VI rotor free wake structure after seven revolutions with a yaw angle of 30° and $V_\infty = 8$ m/s.

7.4.3 Comparison Quantities and Corrections

The validation is based primarily on pressure coefficient distributions and pressure-derived sectional aerodynamic coefficients. In this validation campaign, the following notation is adopted for the sectional aerodynamic coefficients: C_N denotes the normal-force coefficient, C_T the tangential-force coefficient, C_M the pitching-moment coefficient, C_{TH} the thrust coefficient, and C_{TQ} the torque coefficient. This coefficient nomenclature is kept in the current section unless otherwise stated.

The experimental pressure coefficient is defined as follows:

$$C_p = \frac{P_{\text{meas}}}{Q_{\text{norm}}}. \tag{7.14}$$

Here, P_{meas} is the measured pressure difference after corrections for hydrostatic and centrifugal effects. Q_{norm} is the local dynamic pressure obtained from the stagnation-point pressure at the corresponding radial station [11].

The experimental C_N and C_T coefficients are obtained by integrating the chordwise pressure distributions around each instrumented section. C_N acts perpendicular to the local chord line, and is computed using the convention in [11] as

$$C_N = \sum_{i=1}^{N_p} \frac{C_{p,i} + C_{p,i+1}}{2} \left(x_i - x_{i+1}\right), \tag{7.15}$$

C_T acts parallel to the chord line and is computed in a similar way. According to [11],

$$C_T = \sum_{i=1}^{N_p} \frac{C_{p,i} + C_{p,i+1}}{2} \left(y_i - y_{i+1}\right), \tag{7.16}$$

In both Eqs. (7.15) and (7.16), N_p is the number of pressure-tap intervals, $C_{p,i}$ is the pressure coefficient at the i-th tap, and x_i and y_i are the normalised chordwise and normal coordinates of the tap location.

The pitching-moment coefficient about the quarter-chord point is computed as

$$C_M = -\sum_{i=1}^{N_p} \frac{C_{p,i} + C_{p,i+1}}{2} \left[(x_i - x_{i+1}) \left(\frac{x_i + x_{i+1}}{2} - 0.25\right) + (y_i - y_{i+1}) \left(\frac{y_i + y_{i+1}}{2}\right)\right]. \tag{7.17}$$

The report further defines sectional thrust and torque coefficients by projecting C_N and C_T using the blade pitch angle θ_c and twist angle θ_{tw}, and setting the cone angle to zero:

$$C_{TH} = \left[C_N \cos(\theta_{\text{tw}} + \theta_c) - C_T \sin(\theta_{\text{tw}} + \theta_c)\right]. \tag{7.18}$$

$$C_{TQ} = \left[C_N \sin(\theta_{\text{tw}} + \theta_c) + C_T \cos(\theta_{\text{tw}} + \theta_c)\right], \tag{7.19}$$

The global experimental thrust and torque estimates are obtained by integrating the sectional thrust and torque coefficients along the span and multiplying by the number of blades. The NREL report explicitly notes that they are rough estimates based on a limited number of full-chord pressure stations, do not include a tip-loss model, and are not intended to be valid under strongly yawed conditions, since the blades are not symmetrically loaded [11]. For this reason, the present validation emphasises sectional coefficients and pressure distributions rather than integrated rotor loads.

The tunnel blockage was estimated in the original report using a method adapted from Glauert for propeller wind-tunnel testing [63]. The reported blockage was below 2% for all tested conditions and below 1% for most conditions.

In the present validation, only a subset of the experimental database is used. The selected test cases, reported in Tab. 7.3, belong to the dataset named Sequence S, which also includes yawed conditions. The selected configuration is the upwind, rigid, zero-cone configuration with standard blade tips, a nominal rotational speed of 72 rpm, and a pitch angle of 3°. In the comparison plots, the experimental data are reported as time-averaged values with error bars equal to $\pm 3\sigma_{\text{std}}$, where σ_{std} is the standard deviation. Under a Gaussian assumption, this interval would contain approximately 99.7% of the samples; here it is used as an indicator of the temporal variability within each acquisition window.

The selected wind-speed range is chosen so as to include attached or mildly separated operating conditions as well as higher-loading cases where stall and rotational three-dimensional effects become more relevant. The latter are not interpreted as strict validation points for the present inviscid formulation, but as diagnostic cases used to identify the physical limits of the model.

Table 7.3: NREL UAE Phase VI operating conditions selected for the VKI-SDPM validation.

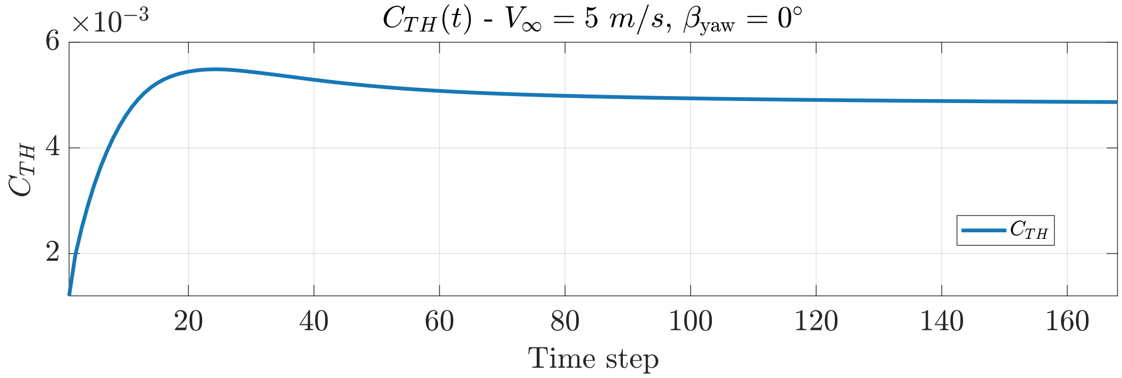
Sequence	V_∞	Pitch angle	Yaw angle	Results reported in
S0500000	5 m/s	3°	0°	Sec. 7.4.4
S0500300	5 m/s	3°	30°	App. D.3.1
S0800000	8 m/s	3°	0°	App. D.3.2
S0800300	8 m/s	3°	30°	Sec. 7.4.4
S1000000	10 m/s	3°	0°	Sec. 7.4.4
S1000300	10 m/s	3°	30°	App. D.3.3

7.4.4 Results and Discussion

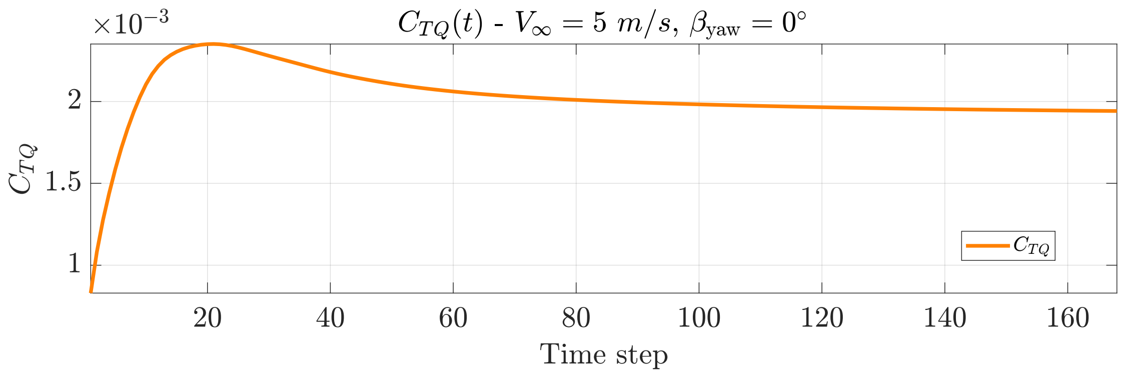
The analysis of the results proceeds from the convergence of the time-marching solution to the global numerical consistency of the spanwise loading, and then to the local pressure coefficient distributions at the instrumented radial stations.

Before analysing the aerodynamic comparison, the convergence behaviour of the time-marching solution is examined. Unlike the fixed-wing test cases, the rotor problem is intrinsically unsteady because the blades rotate while the wake is continuously shed and convected downstream. As a result, the relative position between the blades and the previously shed wake changes periodically during the simulation. Therefore, convergence is not interpreted with a strict residual threshold for $\mathcal{R}(\mu)$, as in Eq. (7.3), but rather as the achievement of a periodic or statistically steady behaviour.

Figures 7.28 and 7.29 report the thrust and torque coefficient histories for S0500000 and S0800300, respectively. In the yawed configuration, a stronger periodic oscillation is expected since the blades experience azimuthally varying local inflow conditions during each revolution. However, both cases present a transient response that tends to disappear when the wake has reached a sufficiently developed state.

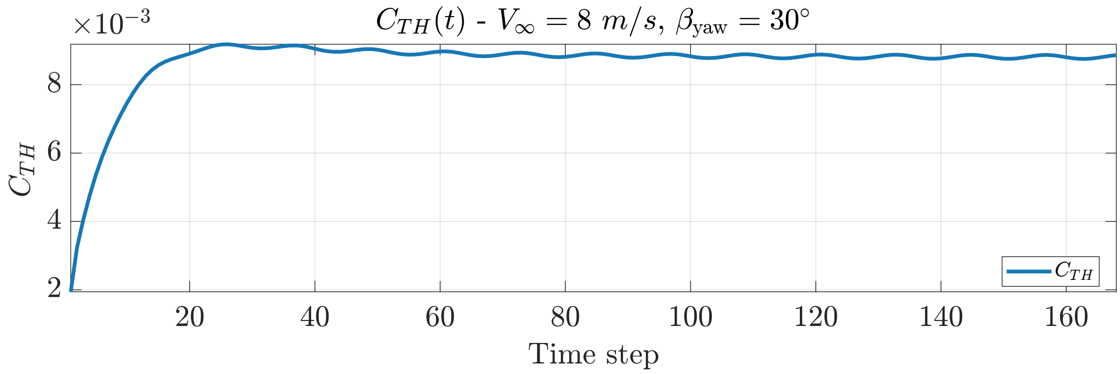


(a) S0500000 thrust coefficient history.

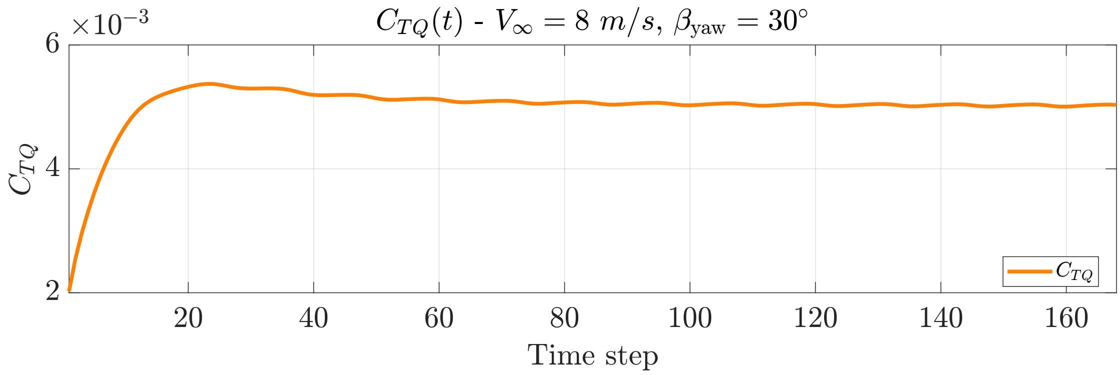


(b) S0500000 torque coefficient history.

Figure 7.28: Thrust and torque histories for the S0500000 test case.



(a) S0800300 thrust coefficient history.



(b) S0800300 torque coefficient history.

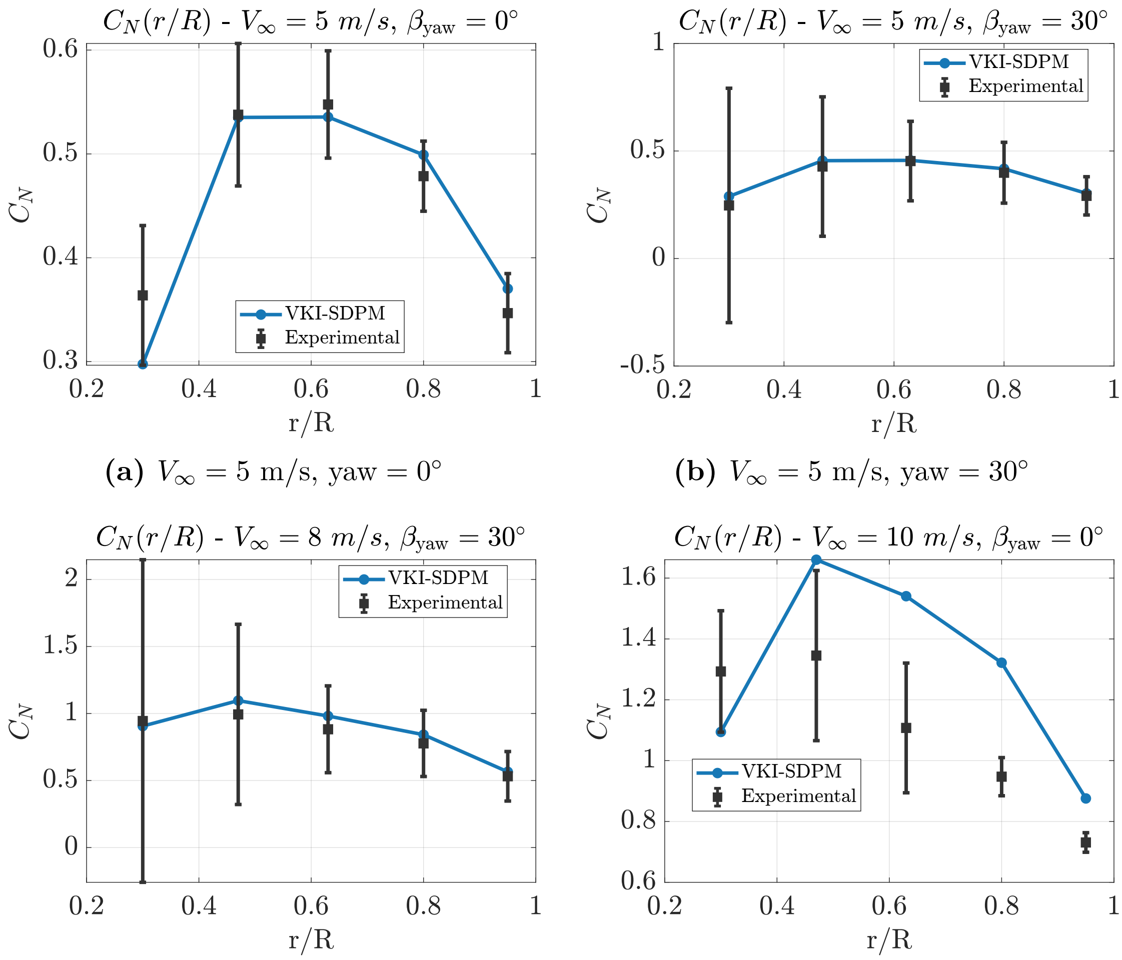
Figure 7.29: Thrust and torque histories for the S0800300 test case.

Spanwise Loading

The comparison in terms of spanwise loading between the numerical and experimental coefficients at the instrumented radial stations provides a meaningful validation check on the physical behaviour of the rotor aerodynamics.

Fig. 7.30 reports the radial distribution for the normal-force coefficient C_N. It can be seen that at low and moderate wind speeds, generally good agreement between the experimental and numerical data is maintained. An apparent improvement at the root station is noticeable in the yawed cases compared with the corresponding axial cases. However, the numerical model does not include the hub, the cylindrical root region, or the root transition geometry; consequently, the numerical model cannot fully match the effects at the innermost station, and the near-root improvement cannot be confidently related to a yaw-stabilising effect. The quantitative agreement deteriorates at higher inflow velocities, as in Figure 7.30d.

As the inflow velocity increases, the local incidence tends to increase and therefore separation effects may be present.



(a) $V_\infty = 5$ m/s, yaw = 0°

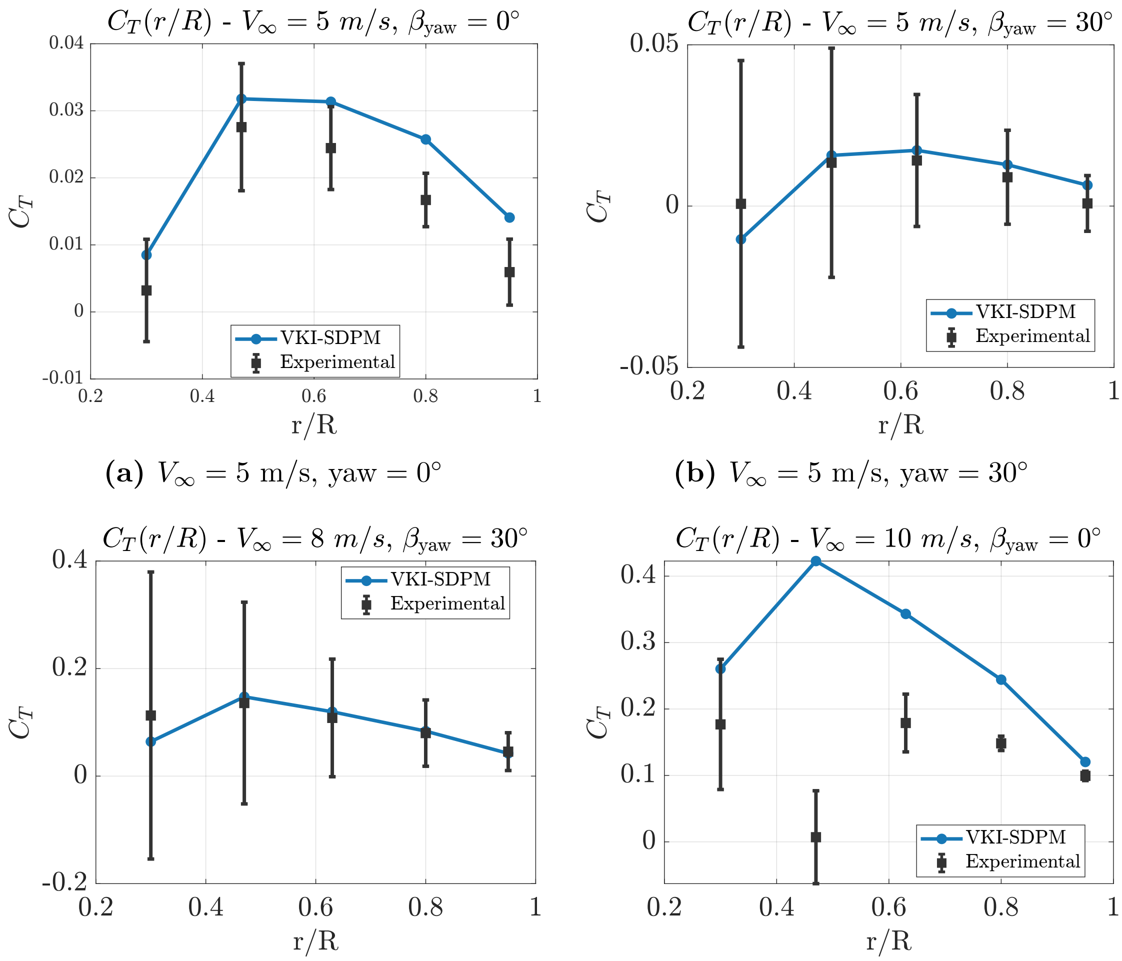
(b) $V_\infty = 5$ m/s, yaw = 30°

(c) $V_\infty = 8$ m/s, yaw = 30°

(d) $V_\infty = 10$ m/s, yaw = 0°

Figure 7.30: Radial distribution of the normal-force coefficient C_N for the selected NREL UAE Phase VI operating conditions.

Figure 7.31 reports the spanwise distribution for the tangential coefficient C_T. The agreement for C_T is generally poorer than that obtained for C_N, even at the lower inflow velocities, but it still tends to improve in the presence of yaw. Figure 7.31d shows a more pronounced discrepancy at high inflow speeds than in the previous case. This may be explained by the higher sensitivity of the tangential force to viscous effects associated with flow separation.



(a) $V_\infty = 5$ m/s, yaw $= 0°$

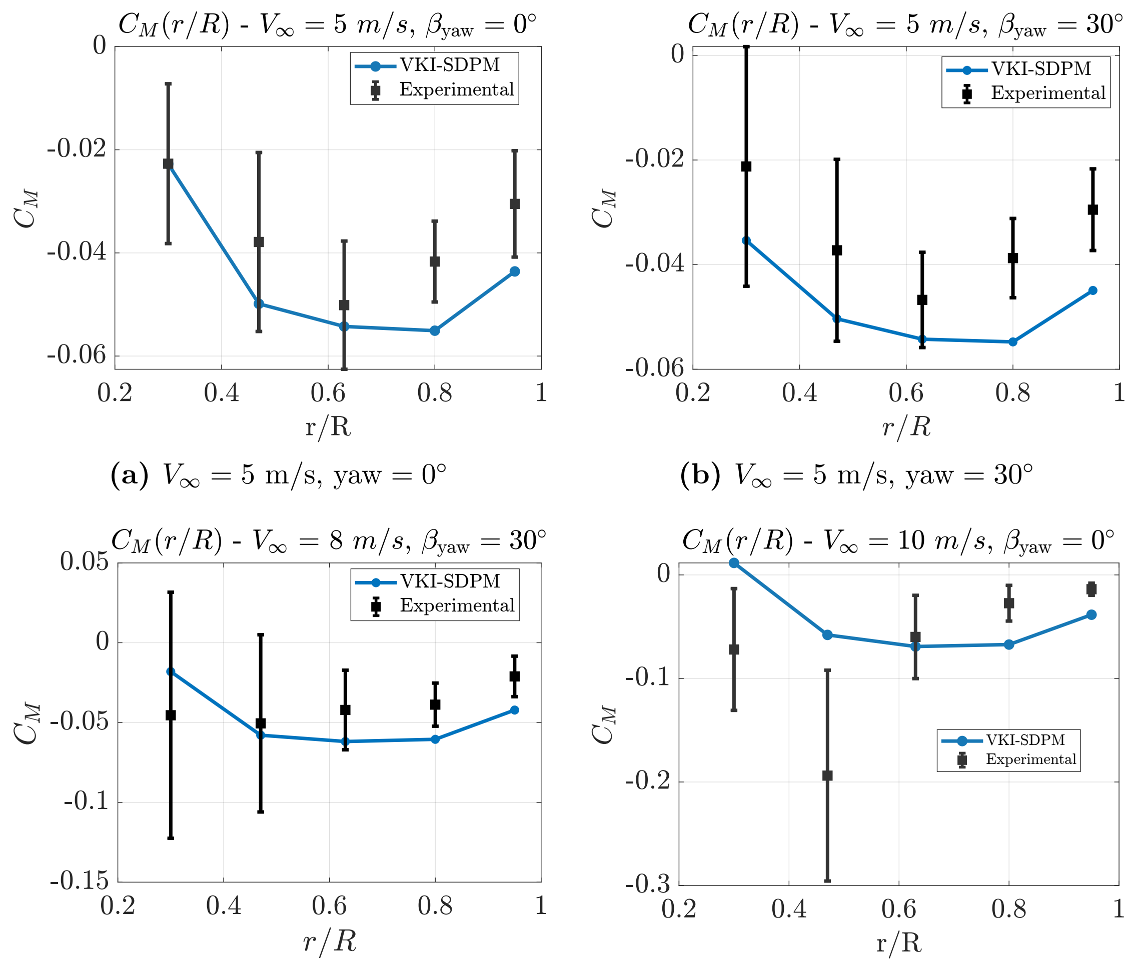
(b) $V_\infty = 5$ m/s, yaw $= 30°$

(c) $V_\infty = 8$ m/s, yaw $= 30°$

(d) $V_\infty = 10$ m/s, yaw $= 0°$

Figure 7.31: Radial distribution of the tangential-force coefficient C_T for the selected NREL UAE Phase VI operating conditions.

The radial distribution of C_M shows general agreement for both axial and yawed inflow, although with lower accuracy. This may be related to the greater sensitivity of the pitching moment to local pressure distribution errors compared with the force coefficients. In this case, the accuracy deteriorates with increasing wind speed.



(a) $V_\infty = 5$ m/s, yaw $= 0°$

(b) $V_\infty = 5$ m/s, yaw $= 30°$

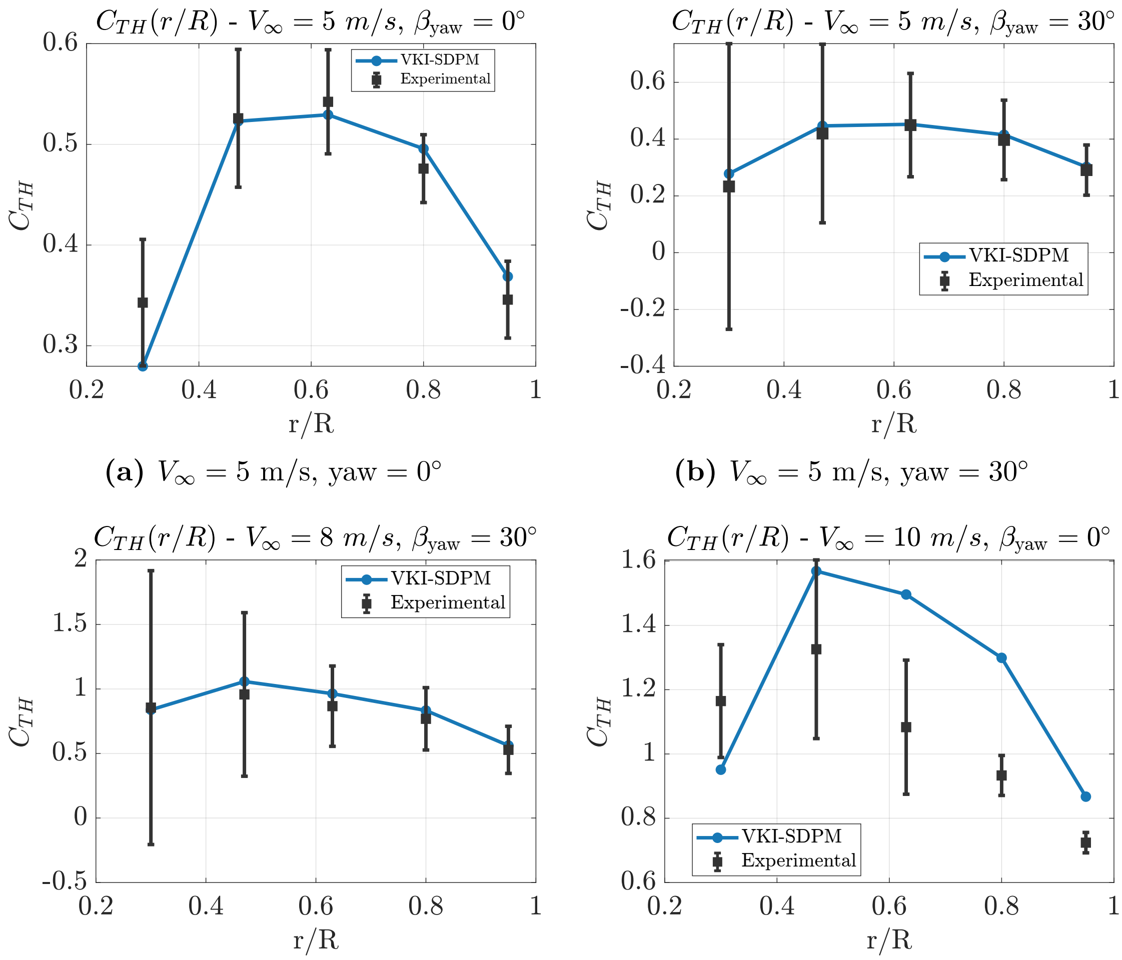
(c) $V_\infty = 8$ m/s, yaw $= 30°$

(d) $V_\infty = 10$ m/s, yaw $= 0°$

Figure 7.32: Radial distribution of the pitching moment coefficient C_M for the selected NREL UAE Phase VI operating conditions.

In conclusion, Figures 7.33 and 7.34 report the radial distributions for C_{TH} and C_{TQ}. A generally satisfactory agreement with the experimental data is observed, especially at low wind speed. This is expected since these two coefficients are obtained by projecting C_N and C_T according to the local pitch and twist, as indicated in Eqs. (7.18) and (7.19). Therefore, their behaviour combines the trends observed for the normal- and tangential-force coefficients.

Overall, the numerical spanwise distributions reproduce the main radial trends in both axial and yawed conditions, although the quantitative agreement is not uniform across all coefficients and operating points. These discrepancies may be related both to experimental measurement uncertainties and to modelling simplifications



(a) $V_\infty = 5$ m/s, yaw $= 0°$

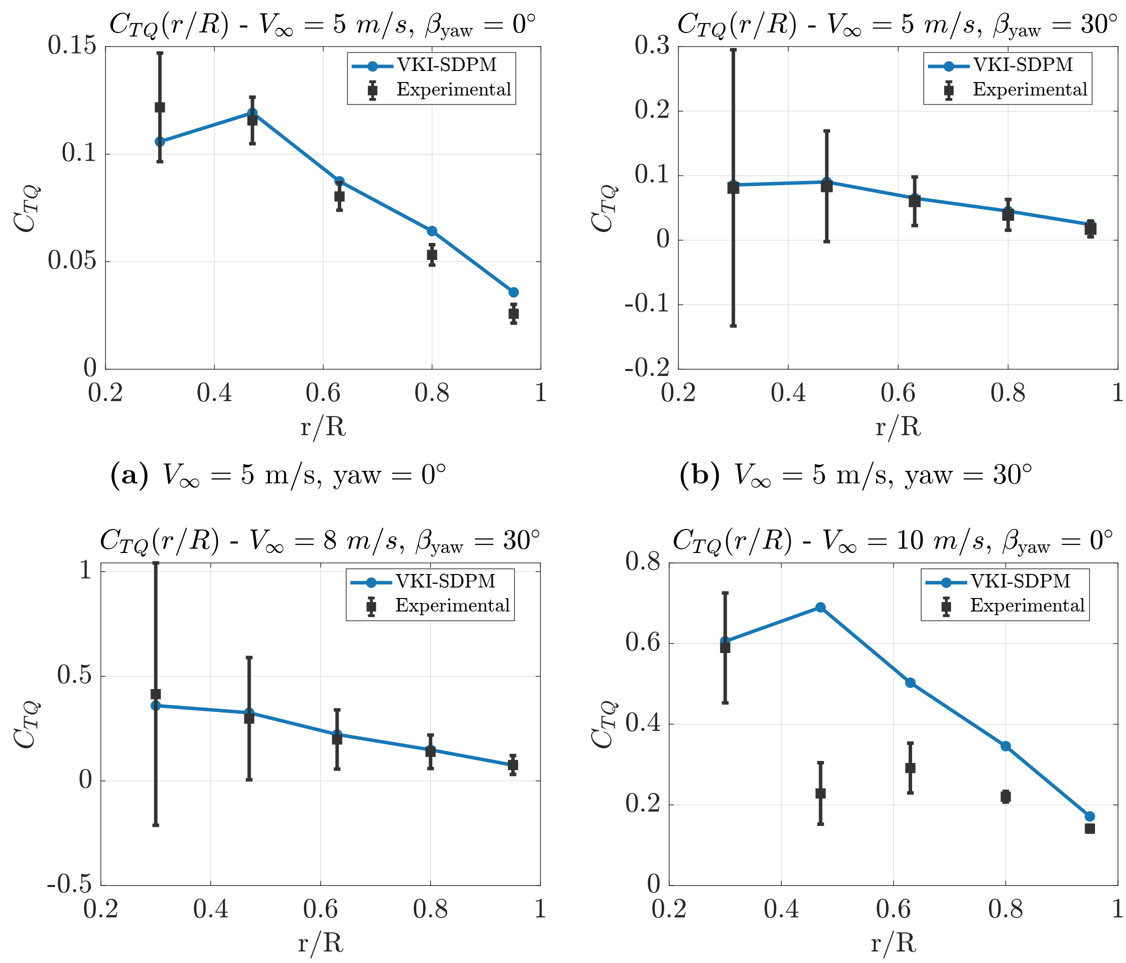
(b) $V_\infty = 5$ m/s, yaw $= 30°$

(c) $V_\infty = 8$ m/s, yaw $= 30°$

(d) $V_\infty = 10$ m/s, yaw $= 0°$

Figure 7.33: Radial distribution of the sectional thrust coefficient C_{TH} for the selected NREL UAE Phase VI operating conditions.

and limitations of the numerical representation. However, the spanwise comparisons support the capability of the solver to reproduce the main radial trends in the rotor loading.



(a) $V_\infty = 5$ m/s, yaw $= 0°$

(b) $V_\infty = 5$ m/s, yaw $= 30°$

(c) $V_\infty = 8$ m/s, yaw $= 30°$

(d) $V_\infty = 10$ m/s, yaw $= 0°$

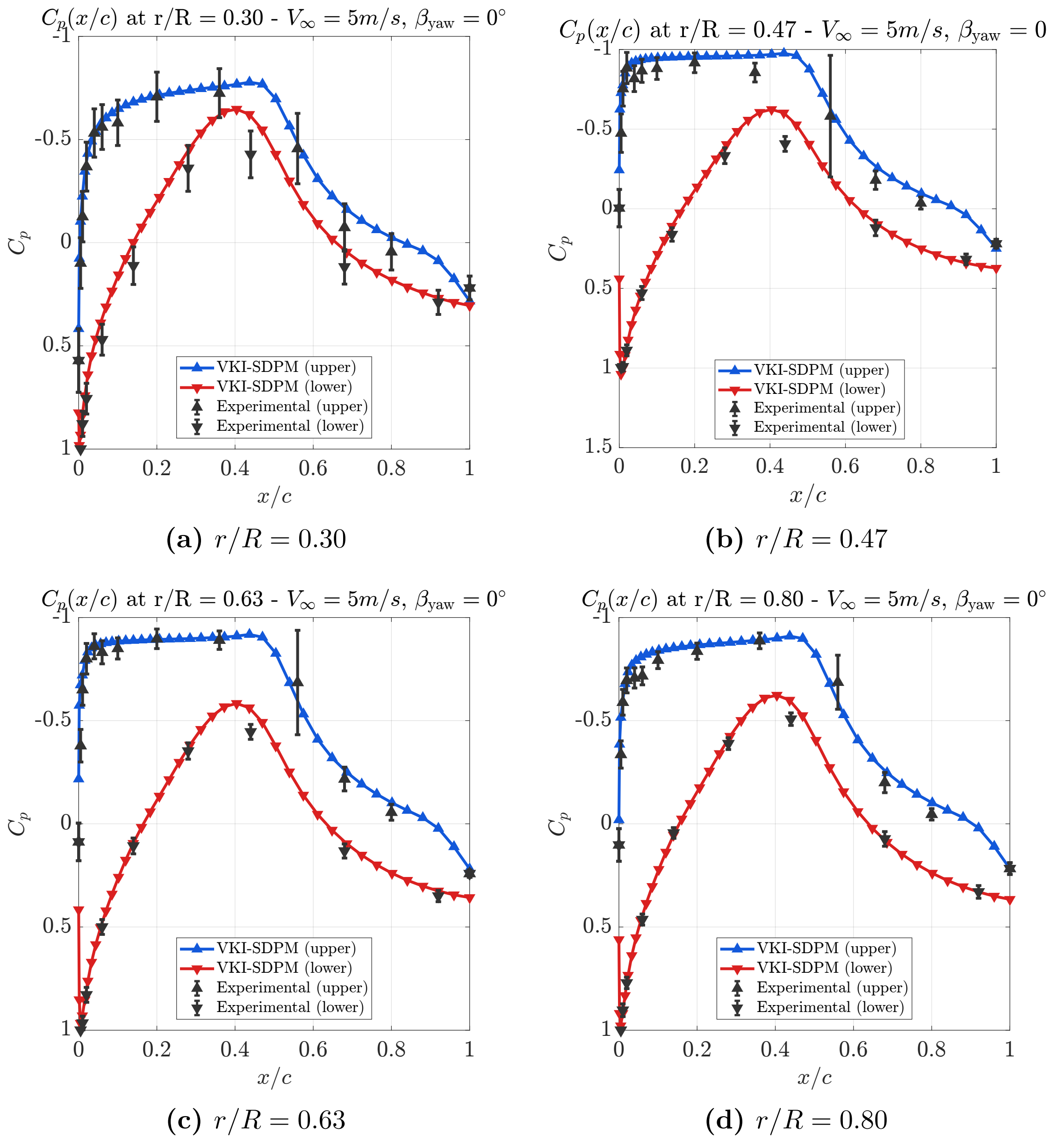
Figure 7.34: Radial distribution of the sectional torque coefficient C_{TQ} for the selected NREL UAE Phase VI operating conditions.

Pressure Coefficient Distributions

The pressure coefficient distributions provide a local assessment of the numerical solution. The comparison is performed at the instrumented radial stations and follows the pressure normalisation convention adopted in the NREL post-processing [11], [64].

Three representative operating conditions are reported. The first corresponds to axial flow at $V_\infty = 5$ m/s (S0500000), the second to yawed inflow at $V_\infty = 8$ m/s (S0800300), and the third to axial flow at $V_\infty = 10$ m/s (S1000000).

The C_p distributions for S0500000 are reported in Fig. 7.35. In this operating condition, the inflow is aligned with the rotor axis and the blade is moderately loaded. Under these conditions, the assumptions underlying an inviscid lifting-surface formulation are more applicable. Local discrepancies are mainly located on the pressure side of the profile at about $x/c \approx 0.5$ and tend to decrease towards the tip.

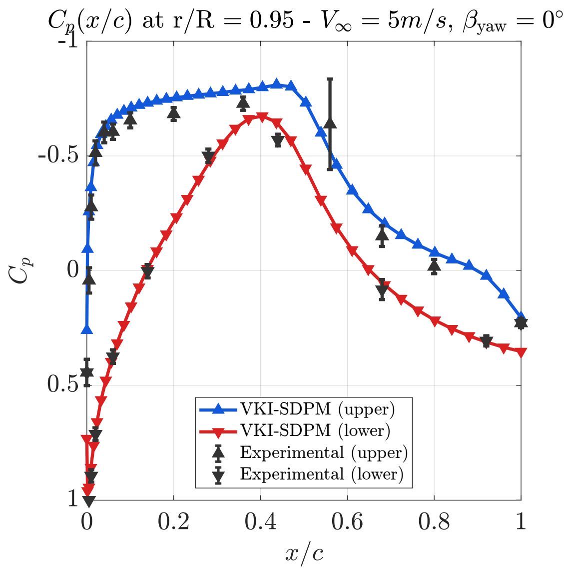


(a) $r/R = 0.30$ **(b)** $r/R = 0.47$

(c) $r/R = 0.63$ **(d)** $r/R = 0.80$

Figure 7.35: Experimental and numerical comparison of pressure coefficient distributions for the NREL UAE Phase VI axial-flow case at $V_\infty = 5$ m/s.

The C_p distributions for the S0800300 case are reported in Fig. 7.36. The



(e) $r/R = 0.95$

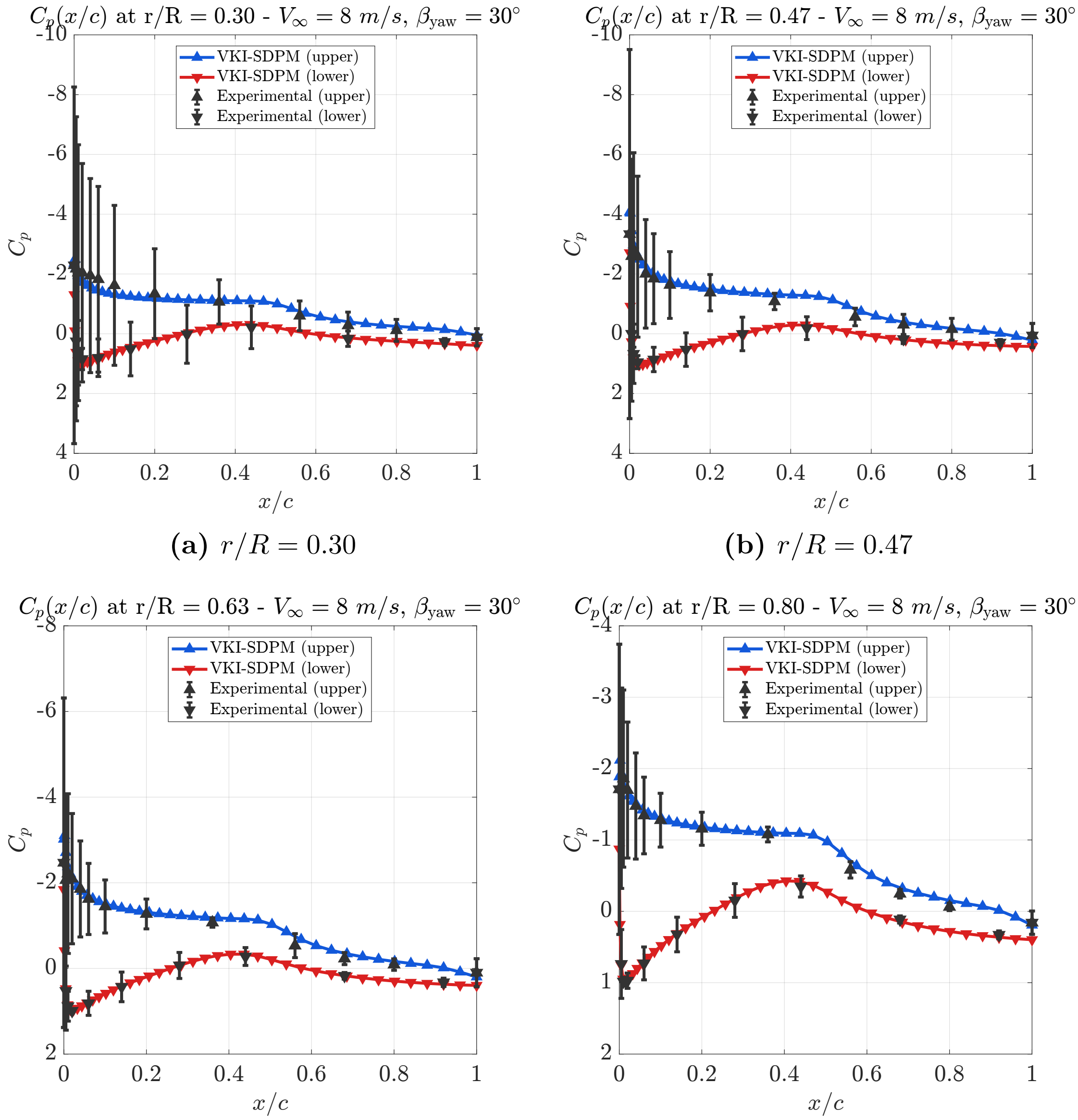
Figure 7.35 (continued)

apparent improvement observed in the radial loads is also reflected in the pressure distributions. However, it should also be noted that the experimental measurements under yawed operation are more affected by unsteady effects and interferences, resulting in larger error bars, thereby making it difficult to link the yawed inflow to a stabilising effect.

The pressure coefficient distributions for $V_\infty = 10$ m/s are reported in Fig. 7.37. At fixed collective pitch and rotational speed, as the freestream velocity increases, the local inflow angle increases as well. As a result, the blade operates closer to stall over portions of the span [6]. In this condition, the pressure distributions are more sensitive to viscous separation.

This effect is particularly evident in Fig. 7.37b, where the section is operating closer to stall. The same behaviour is reflected in the spanwise distributions in the spanwise distributions where the station at $r/R = 0.47$ is the most critical for the S1000000 case. VKI-SDPM cannot reproduce the associated viscous separation effects; therefore, a pronounced discrepancy is observed.

For the remaining stations, the discrepancies are mainly located on the suction side where the solver overpredicts the suction magnitude C_p. However, the overall trends over the remaining portions of the distributions are still captured, with residual discrepancies located near the trailing edge. The presumed stabilising effect of the yawed condition is also observed at this inflow level for the test case S1000300, reported in App. D.3.3.



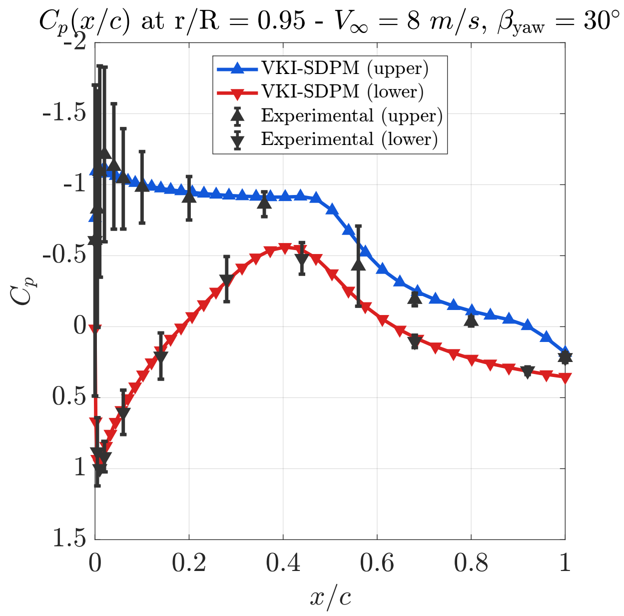
(a) $r/R = 0.30$

(b) $r/R = 0.47$

(c) $r/R = 0.63$

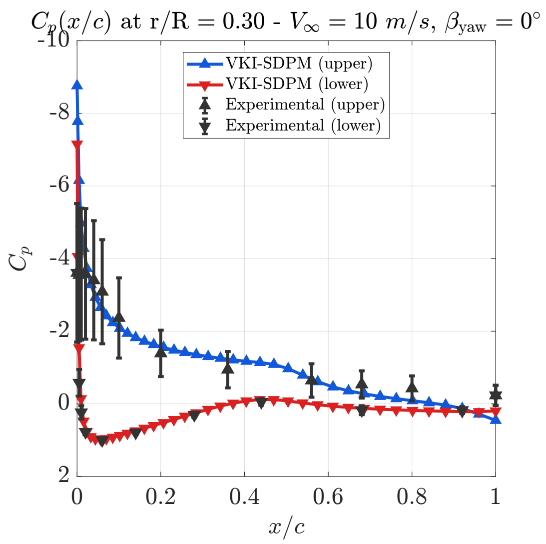
(d) $r/R = 0.80$

Figure 7.36: Experimental and numerical comparison of pressure coefficient distributions for the NREL UAE Phase VI yawed-flow case at $V_\infty = 8$ m/s and yaw = 30°.

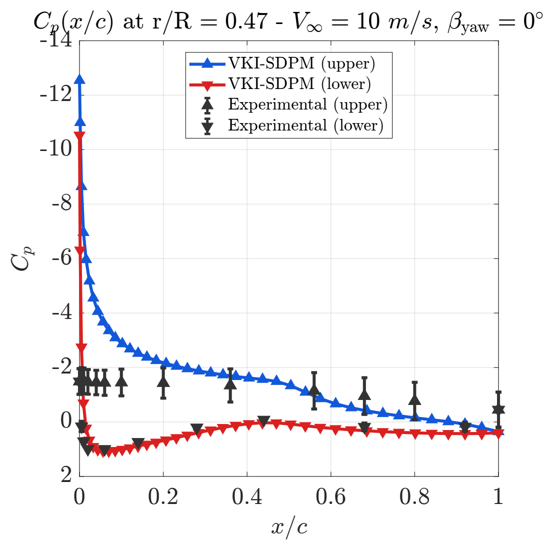


(e) $r/R = 0.95$

Figure 7.36 (continued)

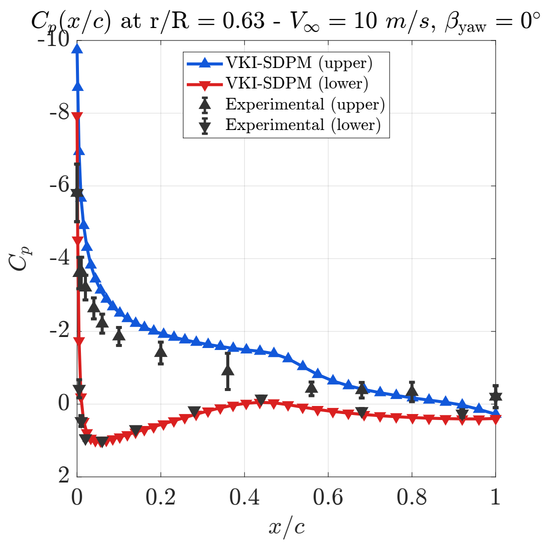


(a) $r/R = 0.30$

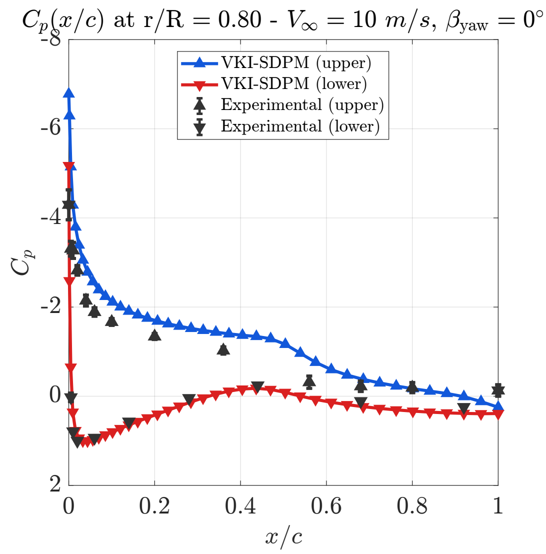


(b) $r/R = 0.47$

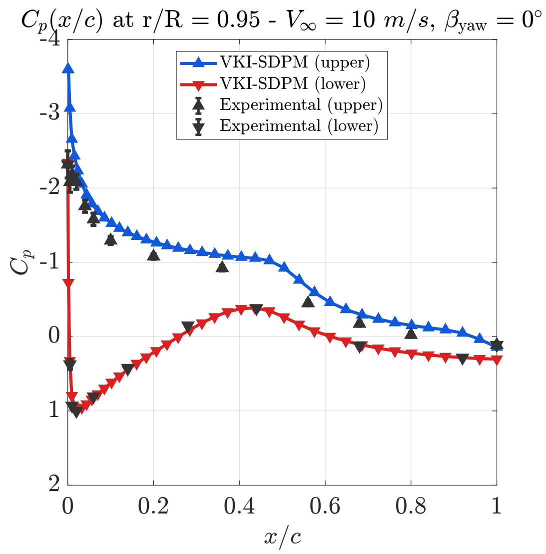
Figure 7.37: Experimental and numerical comparison of pressure coefficient distributions for the NREL UAE Phase VI in axial flow at $V_\infty = 10$ m/s.



(c) $r/R = 0.63$



(d) $r/R = 0.80$



(e) $r/R = 0.95$

Figure 7.37 (continued)

NREL UAE Phase VI Test Case Conclusions

The NREL UAE Phase VI validation extends the assessment of VKI-SDPM to rotor configurations. This represents a significant increase in complexity because the aerodynamic response depends on radial variations of chord and twist, rotational kinematics, and the induced velocity field from the free wake structure.

The convergence analysis confirms that the time-marching solution reaches the expected asymptotic behaviour. In axial flow, the solution tends toward a nearly steady response in the rotating frame. In yawed flow, the solution reaches a periodic oscillatory state, which is consistent with the azimuthal variation of the local inflow conditions.

The quantitative results obtained from the spanwise loading and chordwise pressure distributions support the reliability of the solver within the range of validity of its underlying assumptions. Both sets of results show physically coherent trends and satisfactory agreement with the experimental data.

Focusing on the pressure coefficient distributions, VKI-SDPM can reproduce the dominant chordwise pressure behaviour in low-loading axial and yawed inflow conditions. The agreement deteriorates at higher wind speeds, where the increased local angle of attack and separation effects become more relevant.

Overall, the NREL UAE Phase VI test case confirms the applicability of VKI-SDPM to rotors in axial and yawed inflow configurations, within the range of operating conditions for which the inviscid and incompressible assumptions remain adequate.

7.5 Caradonna–Tung Test Case (NASA-TM-81232)

The last validation case considered for VKI-SDPM is based on the Caradonna–Tung two-bladed rotor in hover [12]. Unlike the validation presented in Ch. 4, this case is used to assess the capability of the solver to simulate a hovering rotor with a freely evolving wake. This makes the test case particularly relevant since the absence of a freestream makes the aerodynamic loading strongly coupled with the wake-induced velocity field.

The experimental setup has already been introduced in Ch. 4; however, the most relevant information for the current validation is briefly recalled below and summarised in Tab. 7.4.

The test case consists of a two-bladed model helicopter rotor operating in hover. The blades are rectangular, untwisted, and untapered, with NACA 0012 airfoil sections. The original experiment included chordwise pressure measurements at several radial stations and hot-wire measurements of the tip-vortex trajectory below the rotor disc.

The selected subset of simulated cases is dictated by the assumptions of VKI-SDPM. Higher rotor speeds would cause larger portions of the blade to experience compressibility effects, while higher collective pitch angles would increase the likelihood of leading-edge separation. Therefore, the cases at rotor speeds of 650 rpm and 1250 rpm were selected, both for collective pitch angles of 5° and 8°.

Table 7.4: Parameters used to configure VKI-SDPM for the Caradonna–Tung test case.

Parameter	**Symbol**	**Value**
Rotor radius	R	1.143 m
Number of blades	N_b	2
Blade chord (constant)	c	0.1905 m
Aspect ratio	AR	6
Blade twist	θ_{tw}	0°
Airfoil section	—	NACA 0012
Collective pitch	θ_c	5°, 8°
Rotor speed	Ω	650, 1250 rpm

7.5.1 Numerical Setup

The numerical model reproduces the aerodynamic rotor geometry of the Caradonna–Tung experiment. The blade airfoil is modelled as a NACA 0012 section with the same analytical closure used in Sec. 7.2. The support column and the hub geometry are not included in the numerical model.

The rotor is simulated in hover by imposing the angular velocity Ω and setting the freestream velocity to zero. The shed wake is freely convected using the local induced-velocity.

The time-marching simulation is advanced for $N_\mathrm{w} = 20$ full revolutions with an azimuthal step of $\Delta\psi = 15°$. This number of revolutions is sufficient for the starting-vortex structure to convect sufficiently far downstream and for the solution to reach a stable periodic behaviour.

For the hover simulations, the initial interaction between the body and the starting vortex may produce numerical instabilities when wake panels pass very close to body panels. In the literature, this sensitivity is a well-known difficulty of panel and vortex methods, as reported in [48]. An example of such initial instability is illustrated in Fig. 7.38 where, right below the rotor disc, wake panels develop spiky geometries due to the high induced velocities.

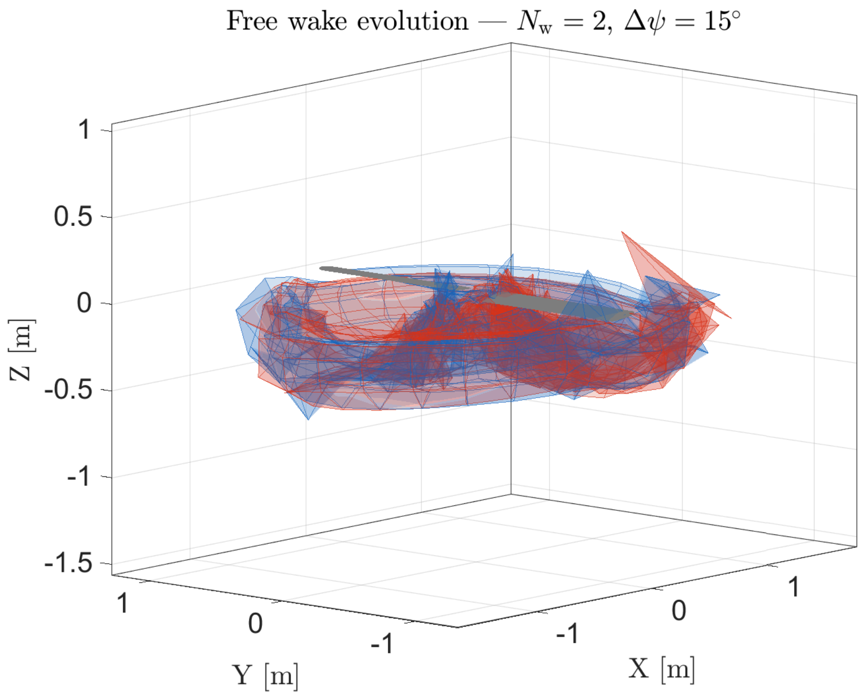


Figure 7.38: Snapshot of the initial instability in the VKI-SDPM free-wake simulation for the Caradonna–Tung rotor.

Therefore, the goal is to keep the initial unstable structures confined, prevent them from spreading to the rest of the wake, and convect them sufficiently far downstream not to influence the steady solution. Examples of the fully developed wake geometries are reported in Fig. 7.39 for the test case at 650 rpm and $\theta_c = 5°$.

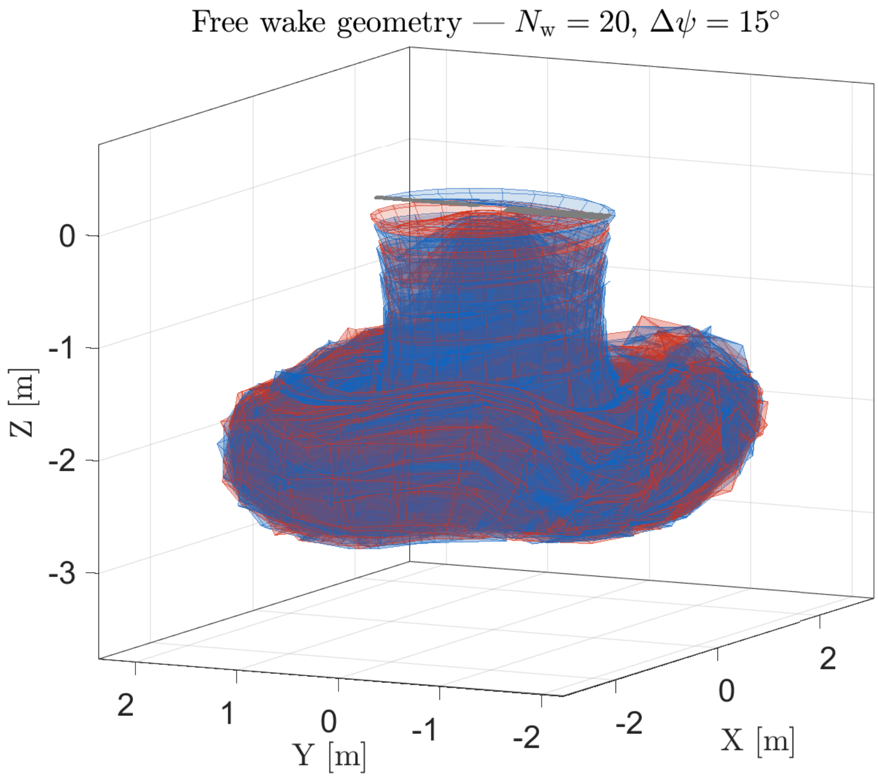


Figure 7.39: Free-wake structure obtained from VKI-SDPM for the Caradonna–Tung rotor at 650 rpm and $\theta_c = 5°$.

To reach a stable steady state, the strategy adopted in this validation campaign consists of using a stepped variation of the diffusion parameter δ within the selected core-growth model. The nature and the dependencies of this parameter were discussed in Subsec. 2.2.3. Here, the choice made for the regularisation is not intended to represent any physical effect, but is rather used as a numerical strategy to mitigate the above-mentioned instabilities. Therefore, for the first four rotor revolutions the value of δ is kept sufficiently high to smooth most of the starting vortex and then is reduced for the newly shed wake panels.

The regularisation model selected for this simulation combines the Lamb–Oseen vortex-core model with the wake-age core-growth law. For all simulations, the initial value of δ is set to $8 \cdot 10^3$ which is above the literature reference values reported in [18] and [19]. However, the target final value was set to $\delta = 10^2$, which is within the indicated range for helicopter rotors in hover.

The initial core radius is set to 1% of c_{ref}, which in this case corresponds to the blade chord. This choice is made to avoid excessively increasing the viscous damping in the initial iterations, already introduced with δ.

The numerical mesh was modified with respect to the baseline from Sec. 6.2. $N_c = 80$ chordwise panels and $N_r = 10$ radial panels per blade are used, with cosine and sine spacing laws, respectively. Using the results of the sensitivity study as a guideline, this choice should not introduce a significant discretisation error.

7.5.2 Comparison Quantities

The thrust coefficient used in the validation comparison is defined as

$$C_T = \frac{T}{\rho V_{\mathrm{tip}}^2 S_{\mathrm{ref}}}, \tag{7.20}$$

where T is the rotor thrust, $V_{\mathrm{tip}} = \Omega R$ is the blade-tip speed, and $S_{\mathrm{ref}} = \pi R^2$ is the rotor swept area.

The pressure coefficient distributions are compared at the experimental radial stations for $r/R = 0.50,\ 0.68,\ 0.80,\ 0.96$. The experimental pressure data are obtained from measurements on the blade surface and are corrected for the effects associated with the rotating pressure-measurement system [12]. The pressure coefficient is computed using the local tangential speed in the definition of the dynamic pressure

$$q(r) = \frac{1}{2}\rho(\Omega r)^2, \tag{7.21}$$

where r is the radial position of the considered blade section.

The numerical results for both thrust and pressure coefficients are averaged over the last two revolutions.

The sectional lift coefficient is obtained by integrating the chordwise pressure distribution as

$$C_l\left(\frac{r}{R}\right) = \int_0^1 \left[C_{p,\mathrm{lower}}\left(\frac{x}{c}, \frac{r}{R}\right) - C_{p,\mathrm{upper}}\left(\frac{x}{c}, \frac{r}{R}\right)\right] d\left(\frac{x}{c}\right). \tag{7.22}$$

The wake trajectory is compared using the radial and axial positions of the tip vortex as functions of the vortex age. The vortex age is expressed in terms of azimuthal angle, so that the wake position can be represented as

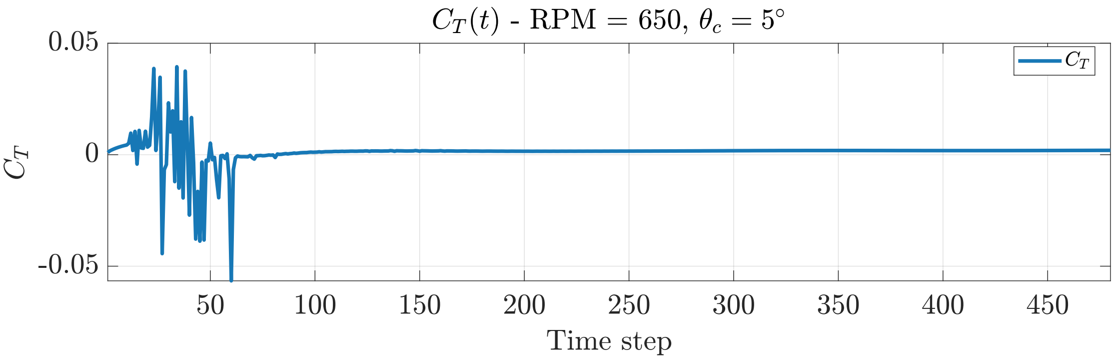
$$\frac{r_{\mathrm{w}}}{R} = f_r(\psi_{\mathrm{w}}), \qquad \frac{z_{\mathrm{w}}}{R} = f_z(\psi_{\mathrm{w}}), \tag{7.23}$$

where r_{w} and z_{w} are the radial and axial coordinates of the tip vortex, and ψ_{w} is the vortex age.

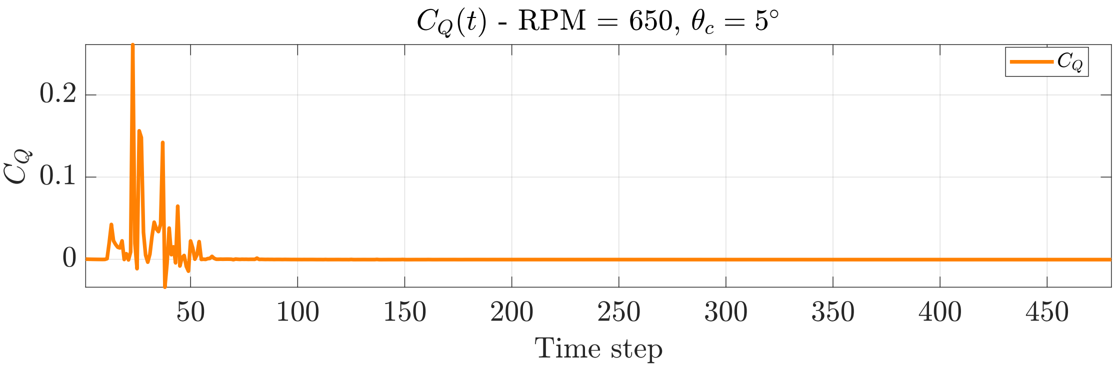
In the experiment, the tip-vortex position is inferred from hot-wire measurements obtained from below the rotor. It is reported that the vortex trajectory is not steady and that the probe location was selected by identifying the positions where vortex-core hits are most frequently observed [12]. Therefore, the wake-trajectory comparison should be interpreted as a comparison of the mean vortex path.

7.5.3 Results and Discussion

The convergence of the solution is first assessed through the time history of the thrust and torque coefficients. Figure 7.40 reports representative thrust- and torque-coefficient histories for the validation case at $\Omega = 650$ rpm, $\theta_c = 5°$. As mentioned before, the initial transient, associated with the wake start-up, produces numerical oscillations that tend to decrease with time as the starting vortices are successfully convected downstream. After the transient, the values of C_T and C_Q increase and finally approach a small-amplitude periodic behaviour [48]. The reported behaviour therefore indicates that a stable periodic solution is reached and supports the numerical strategy implemented for the initial wake stabilisation.



(a) Thrust coefficient history.



(b) Torque coefficient history.

Figure 7.40: Thrust and torque coefficient histories for the VKI-SDPM simulation of the Caradonna-Tung rotor at 650 rpm and $\theta_c = 5°$.

A more quantitative comparison can be performed by comparing the mean value of C_T. Following [50], besides the incompressible mean value, the compressible value obtained by the Prandtl–Glauert transformation is reported in Tab. 7.5. This is obtained by applying the correction based on the local Mach number before integrating the C_p distribution. This correction can be considered reliable since the operating Mach-number range remains within the linear range of the correction [15].

The available numerical results are close to the experimental values, indicating that VKI-SDPM correctly captures the global loading level of the rotor. However, this level of agreement should not be assumed to hold universally, since the tuning of the wake-regularisation parameters remains sensitive for this type of simulation.

Table 7.5 also reports the wall-clock time required for each simulation. The results indicate that for $N_\mathrm{w} = 20$ with an azimuthal step of $\Delta\psi = 15°$ the average wall-clock time is 1105.3 s $\approx$ 18.4 min, which is still within the estimated range for this class of solvers reported in [1].

Table 7.5: Thrust coefficients and simulation times for the Caradonna–Tung hover cases.

	650 rpm		1250 rpm	
θ_c	5°	8°	5°	8°
M_tip	0.23	0.23	0.44	0.44
$C_{T,\mathrm{exp}}$	—	—	0.00213	0.00459
$\overline{C}_{T,M=0}$	0.00191	0.00404	0.00222	0.00459
σ_std (last 2 revs)	$3.26 \cdot 10^{-5}$	$2.94 \cdot 10^{-5}$	$8.42 \cdot 10^{-6}$	$1.16 \cdot 10^{-5}$
$\Delta_{M=0}$ [%]	—	—	+4.23	0.0
$\overline{C}_{T,\mathrm{PG}}$	0.00194	0.00411	0.00238	0.00490
Δ_PG [%]	—	—	+11.74	+6.75
Wall-clock time [s]	1142.5	1155.7	1070.0	1053.1

Sectional Loading Distribution

Figure 7.41 reports the comparison between experimental and numerical sectional lift distributions for $\Omega = 1250$ rpm at $\theta_c = 8°$. In order to better understand the effects of compressibility on the results, the distribution obtained with the Prandtl–Glauert correction is also reported. The numerical trend is physically reasonable and the quantitative agreement is satisfactory for most of the radial stations. The effect of the compressibility correction on the sectional loads becomes more evident towards the tip as the radial Mach number increases.

The noticeable mismatches between numerical and experimental data have also been reported for other potential solvers in [12]. In the report such differences in load predictions are linked to differences between the numerical and experimental wake geometries.

Moreover, the experimental pressure data reported in [12] are not uniformly available for all the radial sections. Therefore, an additional mismatch may be caused by the difference in chordwise resolution between the numerical and experimental pressure integrations.

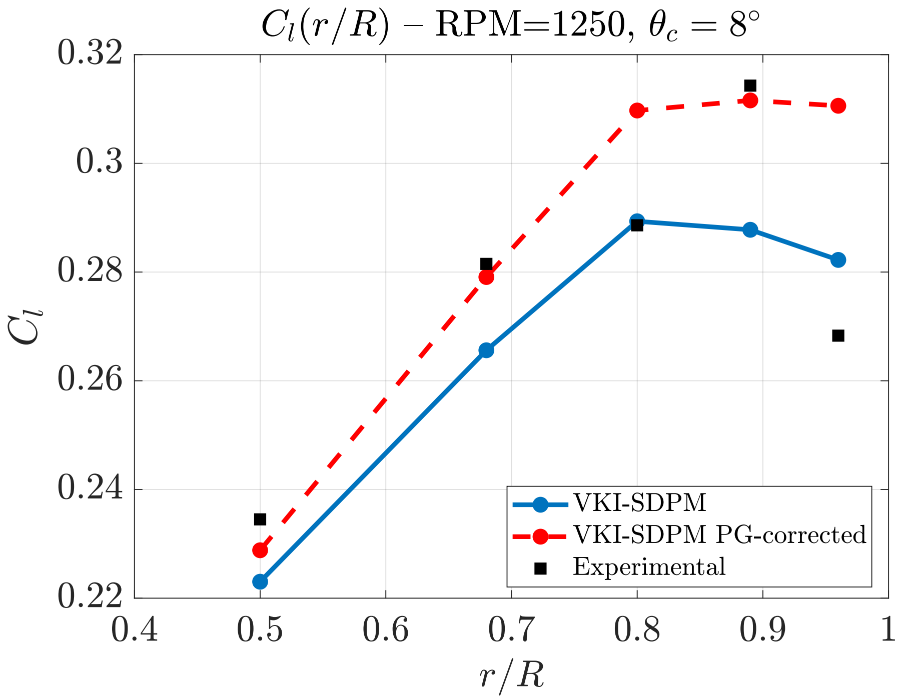


Figure 7.41: Experimental and numerical comparison of sectional lift distributions for the Caradonna–Tung rotor at $\Omega = 1250$ rpm and $\theta_c = 8°$.

Pressure Coefficient Distributions

pressure coefficient distributions are compared for two operating conditions: $\Omega =$ 650 rpm, $\theta_c = 5°$, and $\Omega = 1250$ rpm, $\theta_c = 8°$, reported in Figs. 7.42 and 7.43, respectively. Additional pressure-distribution comparisons for the cases at $\Omega =$ 650 rpm, $\theta_c = 8°$, and $\Omega = 1250$ rpm, $\theta_c = 5°$ are provided in App. D.4.

For both operating conditions, VKI-SDPM reproduces the chordwise pressure behaviour at the considered radial stations. However, small discrepancies localised near the leading edge are still detectable, especially at higher rotational speed. These discrepancies may be caused by mild compressibility effects, since since no clear evidence of flow separation is observed in the experimental pressure distributions.

Overall, chordwise pressure trends are well reproduced for both cases and no significant quantitative differences between numerical and experimental results are observed. This supports the consistency between the computed chordwise pressure distributions and the spanwise lift distribution discussed previously.

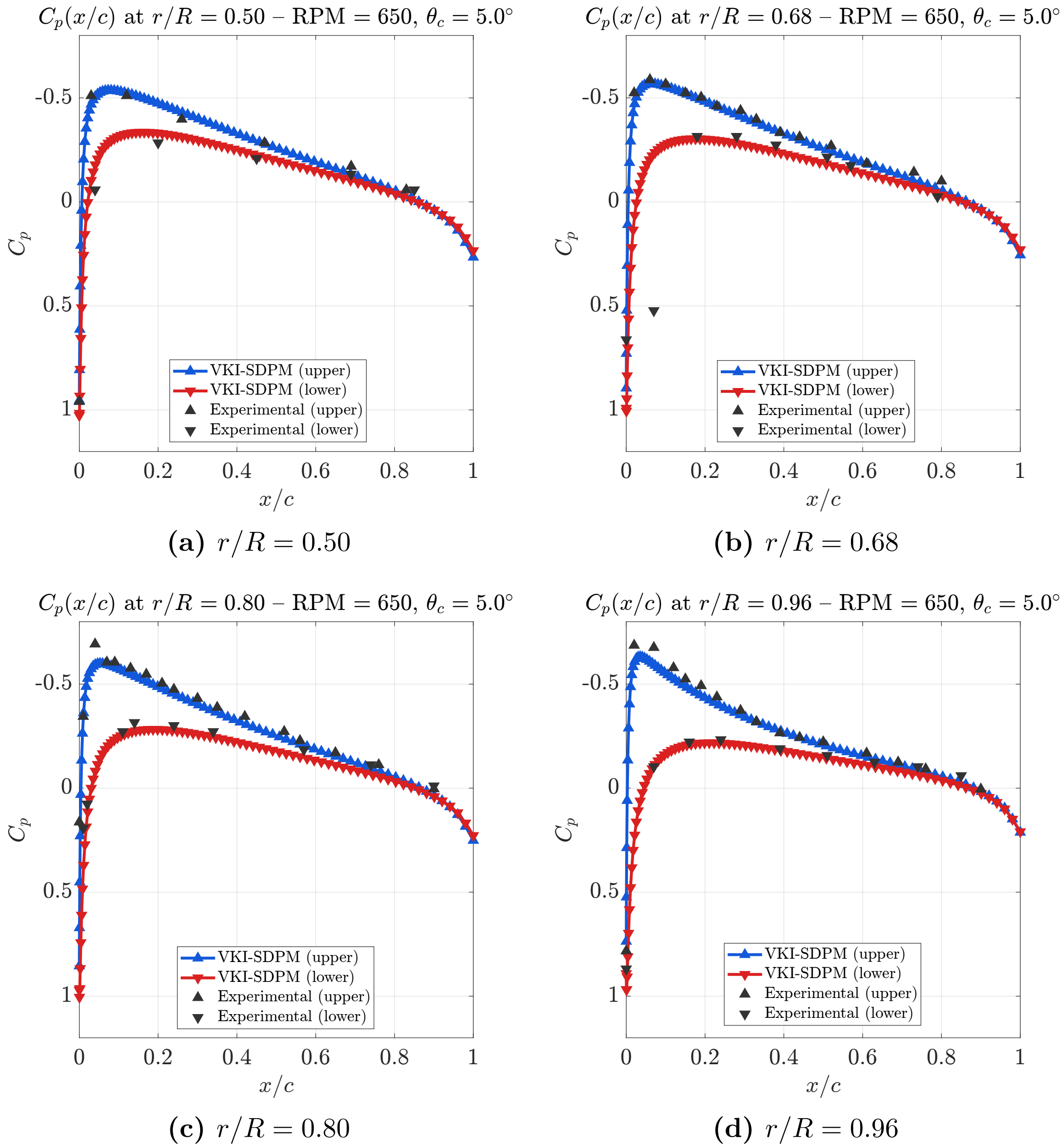
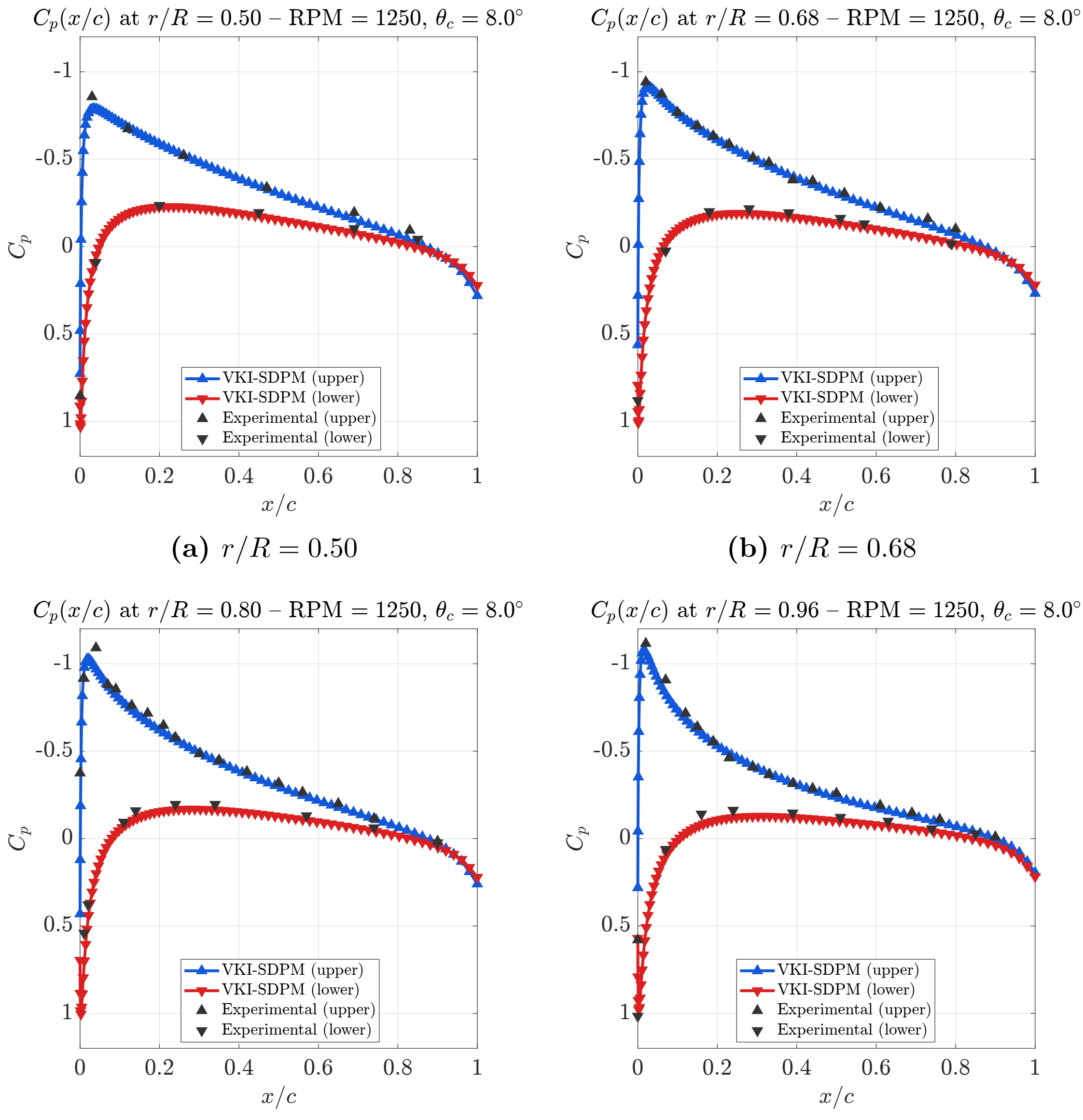


Figure 7.42: Experimental and numerical comparison of pressure coefficient distributions for the Caradonna–Tung rotor at $\Omega = 650$ rpm and $\theta_c = 5°$.



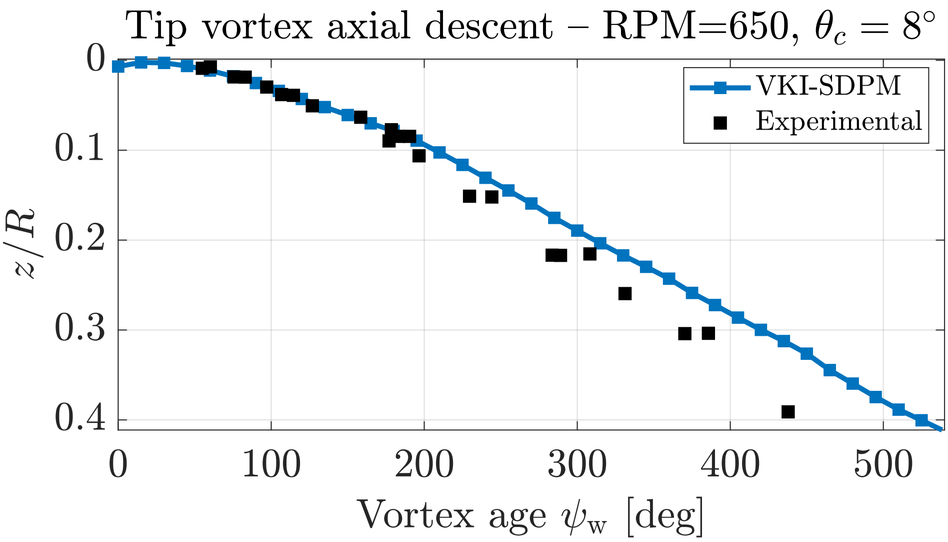
(a) $r/R = 0.50$ (b) $r/R = 0.68$ (c) $r/R = 0.80$ (d) $r/R = 0.96$

Figure 7.43: Experimental and numerical comparison of pressure coefficient distributions for the Caradonna–Tung rotor at $\Omega = 1250$ rpm and $\theta_c = 8°$.

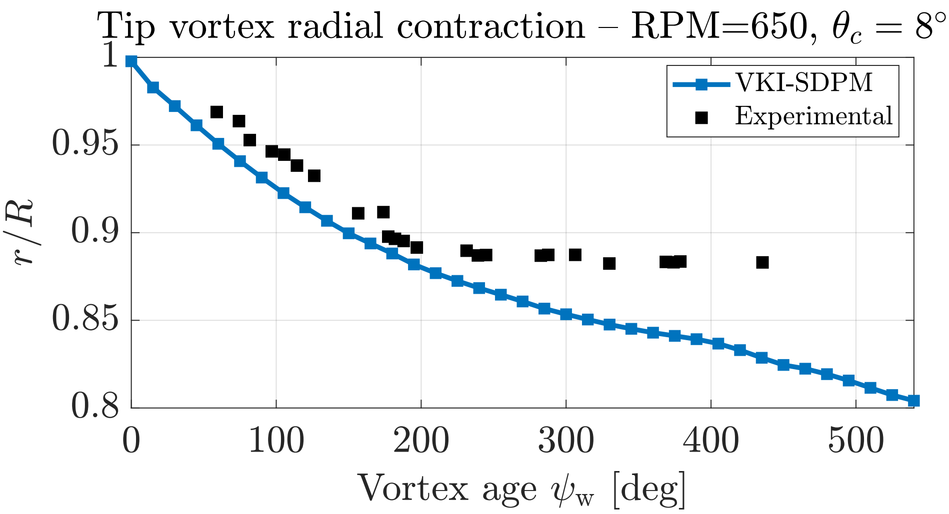
Tip-Vortex Trajectory

The final comparison concerns the tip-vortex trajectory. This quantity directly assesses the ability of VKI-SDPM to generate and convect a physically consistent free wake in hover.

Figures 7.44 and 7.45 compare the numerically computed axial trajectories and radial contraction with the available experimental data.

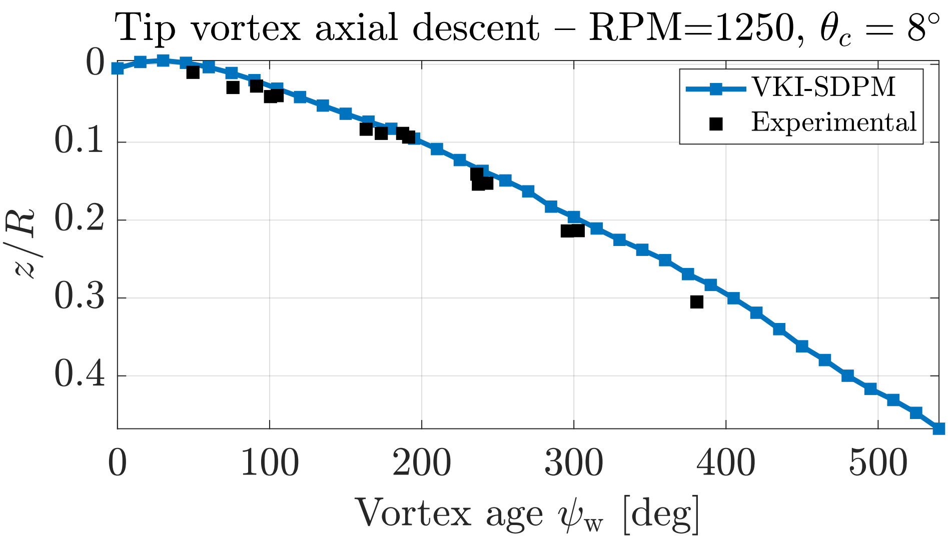


(a) Wake axial trajectory comparison.

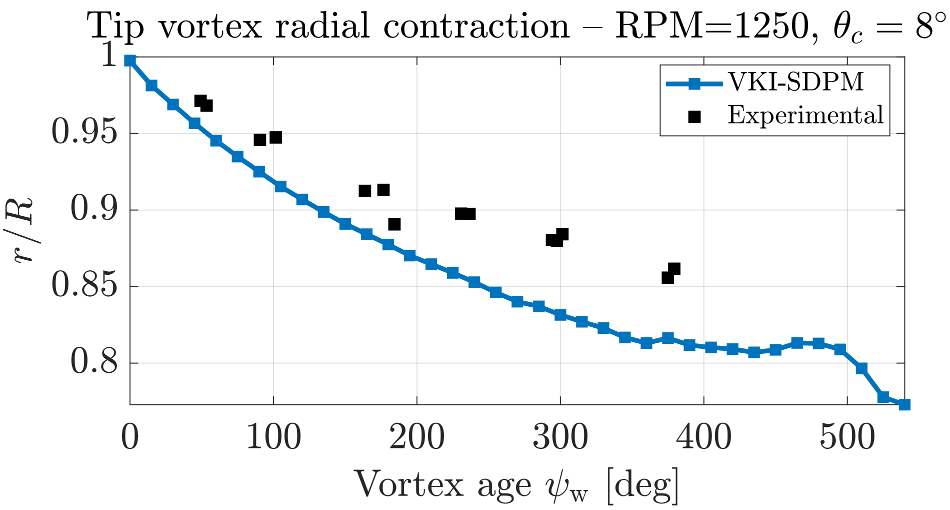


(b) Wake radial trajectory comparison.

Figure 7.44: Experimental and numerical comparison of the tip-vortex trajectory for the Caradonna–Tung rotor at $\Omega = 650$ rpm and $\theta_c = 8°$.



(a) Wake axial trajectory comparison.



(b) Wake radial trajectory comparison.

Figure 7.45: Experimental and numerical comparison of the tip-vortex trajectory for the Caradonna–Tung rotor at $\Omega = 1250$ rpm and $\theta_c = 8°$.

The comparison shows that VKI-SDPM accurately reproduces the mean axial descent of the tip vortex in the near-wake region. On the other hand, the radial contraction is captured only qualitatively. The discrepancy increases with downstream distance from the rotor disc. In hover, the wake contraction is generally influenced by the rotor load and therefore by the induced-velocity field and wake regularisation [45], [46]. In the regularisation analysis for axial flow in Subsec. 6.3.3, it was observed that the tip-vortex radial contraction is more affected by the regularisation parameters than the axial trajectory. This sensitivity may be further amplified by the absence of a freestream, thereby highlighting the challenging nature of hover simulations.

A more detailed sensitivity study of δ and $r_{c,0}$ could help identify parameter values that improve the prediction of the wake contraction. However, as discussed in Subsec. 2.2.3, at present no analytical relation exists to determine the above-mentioned regularisation parameters from the geometrical and operating conditions. Therefore, a parameter-selection process is therefore required to achieve a compromise between numerical stability and physical accuracy.

In conclusion, the obtained results can be regarded as satisfactory for the purposes of this validation.

7.5.4 Caradonna–Tung Test Case Conclusions

The Caradonna–Tung test case extends the validation of VKI-SDPM to hover-rotor aerodynamics. This is a challenging condition because of the complexity of the wake convection.

Therefore, the convergence of the numerical solution towards a stable periodic state was first assessed. The thrust- and torque-coefficient histories support the attainment of a stable periodic solution. The resulting mean loading is then assessed through comparison with the available experimental data.

The pressure coefficient comparison shows good agreement with the experimental data and supports the capability of VKI-SDPM to reproduce the chordwise pressure distributions and the main radial loading trends.

The mean tip-vortex trajectory is well reproduced with higher accuracy for the axial path than for the radial contraction. This is likely a consequence of the greater sensitivity of the radial contraction to the regularisation parameters. These differences may also partially explain the discrepancies observed in the spanwise lift-coefficient distribution.

Nevertheless, the overall numerical results are in good agreement with the experimental data. This provides a particularly important validation since the wake geometry is obtained from the free-wake calculation in a challenging operating condition, thereby providing a stronger physical validation of the solver.

Finally, the Caradonna–Tung test case demonstrates the capability of VKI-SDPM to reproduce the dominant aerodynamic mechanisms of a hover rotor within the range where the inviscid, incompressible potential-flow assumptions remain acceptable.

7.6 Validation Campaign Conclusions and Remarks

The validation campaign presented in this chapter has assessed VKI-SDPM over a sequence of test cases of increasing aerodynamic and geometrical complexity.

The Yip–Shubert test case provided the baseline validation of VKI-SDPM for a finite unswept wing in low-subsonic conditions. This case was particularly useful because it involved a simple geometry and detailed pressure measurements, allowing the surface pressure field, integral coefficients, and spanwise loading to be assessed. The results showed that VKI-SDPM reproduces the dominant inviscid behaviour of a finite wing, with good agreement for the global and sectional loads.

The Kolbe–Boltz swept-wing case extended the validation to a more demanding wing configuration. The presence of sweep and taper introduced a non-uniform spanwise loading distribution and a stronger coupling between local chordwise pressure behaviour and global aerodynamic coefficients. The comparison confirmed that VKI-SDPM is able to reproduce the main loading trends of swept and tapered planforms at low Mach numbers and moderate angles of attack.

The NREL UAE Phase VI test case extended the assessment from wings to rotors. This represented a significant increase in complexity because the aerodynamic response depends on spanwise-varying blade geometry, rotational kinematics, and azimuthal variations in yawed inflow. The results showed that VKI-SDPM reproduces the correct behaviour in both axial and yawed operating conditions within the experimental uncertainty. In particular, the spanwise loading and pressure coefficient distributions were in satisfactory agreement with the experimental data in the operating range where the assumptions underlying VKI-SDPM remain valid.

Finally, the Caradonna–Tung hover-rotor case represented the most demanding validation for the free-wake capability of VKI-SDPM. Unlike the NREL case, no freestream velocity convects the wake away from the rotor disc. Therefore, the prediction of the induced velocity field becomes a central part of the validation. The results showed that VKI-SDPM can reach a steady hover solution and reproduce the experimental thrust coefficient with good accuracy. The pressure coefficient distributions confirmed that the solver captures the local loading mechanisms of the hovering rotor. The comparison of the tip-vortex trajectory further supported the physical consistency of the free-wake evolution.

The validation campaign confirms that VKI-SDPM works as a robust and physically consistent aerodynamic solver, for the class of problems targeted in this thesis. The solver provides reliable pressure distributions, integrated loads, and induced-velocity-field information when the flow is predominantly attached and compressibility and viscous effects remain sufficiently limited for the potential-flow assumptions to provide an adequate representation.

Chapter 8

Thesis Conclusions and Future Work

The previous chapters presented the formulation, implementation, and validation of the two aerodynamic solvers developed in this thesis: VKI-BELL and VKI-SDPM. The objective of this chapter is to compare the two formulations in terms of modelling capabilities, common validation results, computational requirements, and range of applicability.

The comparison is not intended to identify a single best solver in absolute terms. Rather, it aims to clarify the type of aerodynamic problem for which each formulation is more appropriate. The Caradonna–Tung hover-rotor test case is used as the common reference case since it was considered in the validation of both solvers.

The chapter closes the thesis by first comparing VKI-BELL and VKI-SDPM in terms of modelling capabilities, accuracy, and computational cost, and then by drawing the main conclusions of the work. The limitations identified throughout the validation campaigns are finally used to outline possible future developments for both solvers.

8.1 Overview of the Developed Solvers

VKI-BELL is based on a lifting-line representation of the rotor blades. The bound circulation is defined along the blade span and is solved through a fixed-point iterative procedure. The aerodynamic closure is provided by sectional airfoil data, so that lift and drag coefficients are obtained from prescribed tables as functions of the local angle of attack. The induced velocity field is evaluated through a vortex-segment wake representation, whose geometry can be prescribed or updated through a semi-prescribed outer-loop strategy.

The main advantage of VKI-BELL is its compactness. Since the blade is represented by a lifting line rather than by a surface mesh, the number of aerodynamic unknowns remains limited and the computational cost is relatively low. This makes the solver suitable for rapid rotor-load estimations, sensitivity analyses, and preliminary-design studies. At the same time, the model relies on sectional aerodynamic data; it does not compute chordwise pressure distributions, represent the actual surface geometry, or determine the wake trajectory independently of the selected semi-empirical or iterative wake model

VKI-SDPM is based on a source–doublet panel formulation. The body is represented by a finite-thickness surface mesh, and the unknown doublet distribution is solved on the body panels. Its main advantage is the higher geometrical and aerodynamic fidelity. Since the surface of the blade or wing is explicitly discretised, the solver can provide detailed chordwise pressure distributions and local pressure-based loads, for which no external aerodynamic data are required. The free-wake model also allows the wake geometry to emerge from the computed induced-velocity field, rather than being imposed a priori. These capabilities come at a higher computational cost and require a careful treatment of wake regularisation in hover simulations.

The main modelling differences between the two solvers are summarised in Table 8.1.

Table 8.1: Summary of the main modelling capabilities of VKI-BELL and VKI-SDPM.

Capability	VKI-BELL	VKI-SDPM
Rotor configurations	Yes	Yes
Wing configurations	No	Yes
Thickness modelled	No	Yes
Twist modelled	Yes	Yes
C_p distributions	No	Yes
C_l	From tables	Computed from pressure
C_d	From tables, viscous contribution	Computed, inviscid only
Wake modelling	Fully prescribed / semi-prescribed	Free / prescribed
Computational time	Tens of seconds	Minutes to tens of minutes

8.2 Solvers Comparison for the Caradonna–Tung Test Case

The Caradonna–Tung hover-rotor test case provides the common reference for comparing VKI-BELL and VKI-SDPM. The comparison is performed with the awareness that the two solvers do not provide the same level of output detail.

For completeness, the results of VKI-BEMT are also reported in order to include the lower-fidelity rotor tool developed within the VKI numerical library. The selected operating condition is $\Omega = 1250$ rpm and $\theta_c = 8°$. The compared quantities are the mean thrust coefficient, the wake trajectory, and the sectional lift distribution.

8.2.1 Thrust Prediction and Computational Costs

The thrust coefficient provides the most compact metric for comparing the two solvers. In VKI-BELL, the thrust is obtained by integrating the sectional aerodynamic loads along the blade span. These loads depend on the bound circulation, the sectional polar closure, and the induced velocity generated by the selected wake model. In VKI-SDPM, the thrust is obtained by integrating the pressure distribution over the blade surface.

For the selected Caradonna–Tung hover condition, the comparison is reported in Tab. 8.2. VKI-SDPM value corresponds to the average over the last two rotor revolutions, with a standard deviation of $\sigma_{\text{std}} = 1.16 \cdot 10^{-5}$. The result is obtained without compressibility corrections and may be affected by the selected wake-regularisation parameters.

The same table also reports the simulation time required by each solver and the speed-up with respect to VKI-SDPM. As anticipated, the higher level of detail provided by VKI-SDPM comes with a larger computational cost; nevertheless, the execution time remains within the range reported for comparable medium-fidelity free-wake solvers in [1].

This comparison already shows the different practical roles of the two solvers: VKI-BELL is more appropriate for rapid rotor studies, whereas VKI-SDPM is more appropriate when the actual geometry and the time evolution of the wake must be resolved.

8.2.2 Tip-Vortex Trajectory

The most important distinction between the two solvers is the wake modelling. In VKI-BELL, the wake geometry is prescribed or semi-prescribed, depending on the selected wake-update strategy. Therefore, the wake trajectory reflects the modelling assumptions introduced in the wake construction. For the hover case

Table 8.2: Mean thrust-coefficient comparison for the Caradonna–Tung hover-rotor case.

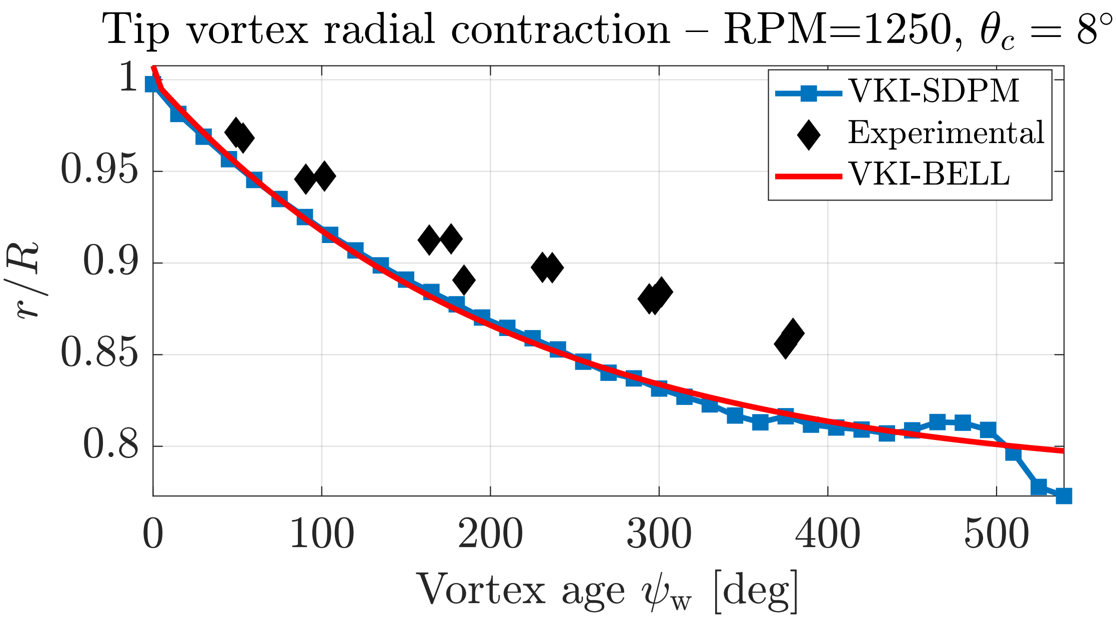
Source	C_T	ΔC_T [%]	Simulation time [s]	Speed-up
Experiment	0.00459	–	–	–
VKI-SDPM	0.00459	0.00	1053.1	1.0
VKI-BELL	0.00519	+13.07	10.71	98.3
VKI-BEMT	0.00574	+25.05	0.42	2507

used in VKI-BELL validation, the wake contraction and axial descent are defined through the corresponding Landgrebe empirical relations.

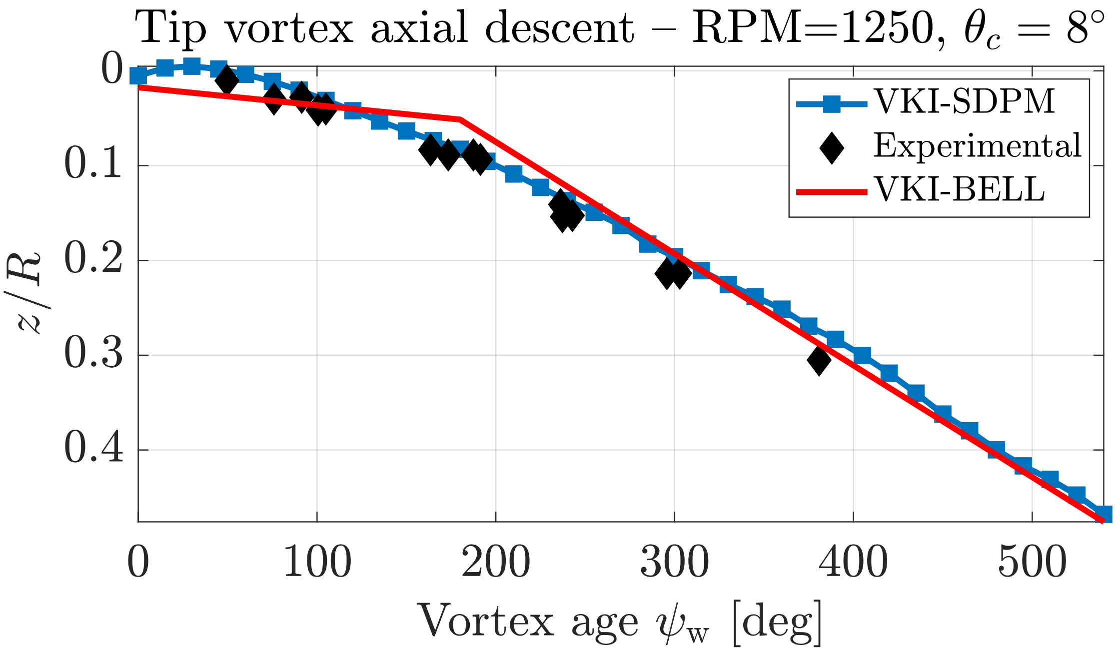
In contrast, VKI-SDPM solves the wake convection through a time-marching free-wake formulation. The tip-vortex trajectory is therefore an output of the simulation and provides an independent validation of the wake model.

Figure 8.1 compares the radial contraction and axial descent predicted by the two solvers. The radial contraction curves are almost superimposed, as shown in Fig. 8.1a. This result may be interpreted by considering the Landgrebe contraction law from Eq. (3.16), used in VKI-BELL: since this relation only depends on the thrust coefficient for a fixed azimuthal wake age ψ_{w}, and since VKI-SDPM predicts a value of C_T very close to the experimental one, used as input in VKI-BELL, the free-wake solution recovers a similar radial-contraction trend of the empirical relation.

A more significant difference appears in the axial descent. As shown in Fig. 8.1b, VKI-SDPM predicts a larger downstream displacement of the tip-vortex, especially around $\psi_{\mathrm{w}} \approx 180°$, which corresponds to the passage of the tip vortex under the opposite blade. In the VKI-BELL validation, issues related to the tip-vortex proximity to the rotor disc were discussed with reference to the original report. The improved agreement of the VKI-SDPM wake trajectory with the experimental data is therefore expected to contribute to a more accurate spanwise load prediction than that obtained with VKI-BELL, particularly in the region directly above the tip vortex.



(a) Tip-vortex radial contraction.



(b) Tip-vortex axial descent.

Figure 8.1: Comparison of the tip-vortex radial contraction and axial descent obtained with VKI-BELL and VKI-SDPM for the Caradonna–Tung test case at $\Omega = 1250$ rpm and $\theta_c = 8°$.

8.2.3 Sectional Lift Distribution

Figure 8.2 compares the sectional lift distributions obtained with VKI-SDPM, with and without the Prandtl–Glauert correction, VKI-BELL, VKI-BEMT, and the experimental data. As expected, VKI-SDPM provides the best agreement with the experimental results. However, VKI-BELL shows good agreement with both the experimental data and VKI-SDPM away from the region directly influenced by the tip vortex. VKI-BEMT provides the least accurate distribution among the considered solvers.

The hypothesis concerning the effect of tip-vortex positioning on the VKI-BELL loading prediction was discussed in Ch. 4. Now the comparison with VKI-SDPM supports that interpretation: a more accurate axial placement of the tip-vortex obtained by the free-wake solution is consistent with the improved sectional-loading prediction. This comparison helps define the limits of the VKI-BELL modelling approach and suggests that the observed local discrepancies are primarily associated with wake modelling rather than with numerical implementation errors.

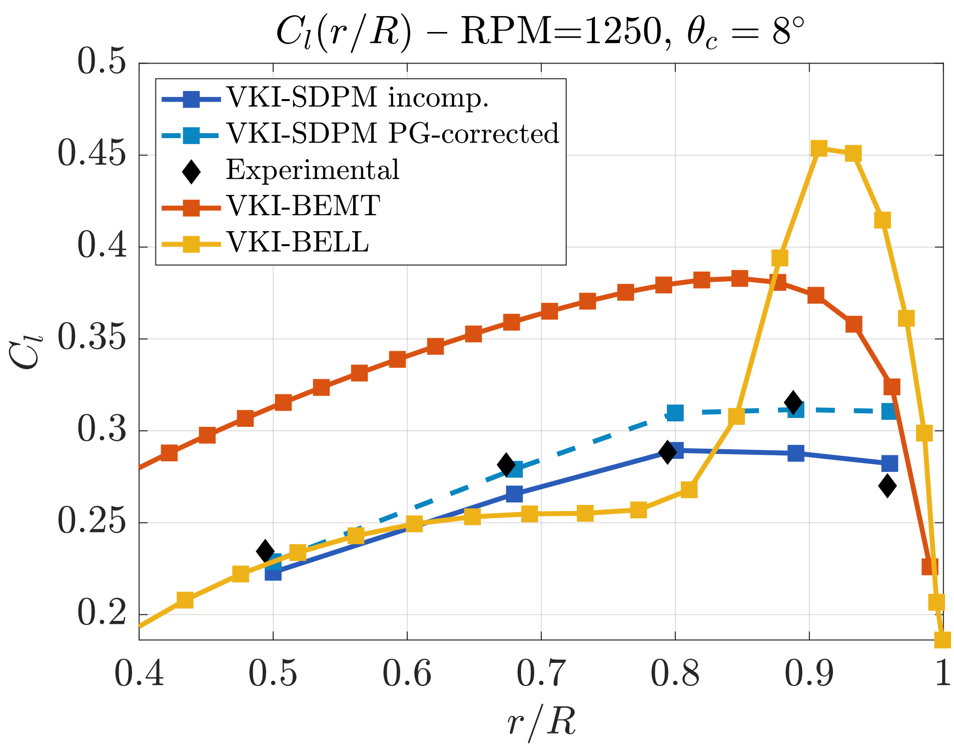


Figure 8.2: Comparison of sectional lift distributions obtained with VKI-BELL and VKI-SDPM for the Caradonna–Tung hover-rotor case.

8.3 General Conclusions

The work presented in this thesis has led to the development, implementation, and validation of two original in-house aerodynamic solvers: VKI-BELL and VKI-SDPM. The two codes address different levels of aerodynamic modelling and capability, as shown by their direct comparison.

VKI-BELL has been shown to be an efficient rotor-load prediction tool with a low computational cost. The Caradonna–Tung validation confirmed that the solver can reproduce the global thrust and most of the sectional loading distribution, although its accuracy remains strongly dependent on the selected wake model and on the quality of the external sectional data.

VKI-SDPM has been validated as a higher-fidelity potential-flow solver. The validation campaign covered unswept wings, swept and tapered wings, wind-turbine rotors, and hover rotors. The results showed that the solver can reproduce global loads, sectional loads, pressure distributions, and free-wake trajectories with satisfactory accuracy when the flow remains compatible with its assumptions. Its main limitations are associated with the absence of viscous-separation and boundary-layer modelling, the lack of profile-drag and compressibility corrections, and the sensitivity of free-wake simulations to vortex-core regularisation.

The comparison between the two solvers shows that their roles are complementary rather than interchangeable. VKI-BELL is the more efficient tool for rotor-load prediction when the geometry can be reduced to a lifting line and reliable sectional polars are available. VKI-SDPM is the more general and physically detailed tool when geometrical accuracy, pressure distributions, and free-wake evolutions are required.

Overall, the thesis demonstrates that VKI-BELL and VKI-SDPM provide two complementary aerodynamic modelling tools operating at different fidelity levels. Both offer substantially lower computational requirements than conventional high-fidelity CFD approaches, while providing different levels of geometrical and aerodynamic detail. The objectives stated in Chapter 1 have therefore been achieved, and the results provide a solid basis for future extensions, discussed in the last section.

8.4 Possible Future Work

The two solvers have reached different levels of maturity and therefore require different types of extension. VKI-BELL should be further validated and expanded as a fast rotor-oriented tool, while VKI-SDPM should be developed toward a more general medium-fidelity aerodynamic solver for complex configurations.

8.4.1 Future developments for VKI-BELL

A first natural continuation for VKI-BELL is the validation under axial-flow propeller conditions and for geometries with spanwise-varying chord, twist, and airfoil sections. In the present thesis, the validation of VKI-BELL has focused on the Caradonna–Tung hover rotor, which is geometrically simple and uses a constant-section blade. A more demanding validation can be performed using the more complex geometry of the VX4 propeller considered in the eVTOL project [65], [66]. The VX4 propeller has been analysed by van Wijk, whose work provides a RANS-based dataset for both axial-flow and hover conditions [67]. These results could provide a higher-fidelity reference for the validation of VKI-BELL beyond the already available experimental data.

A second development path for VKI-BELL concerns the modelling of the tip-vortex trajectory. The validation results showed that the prediction of the sectional loading is sensitive to the position of the tip-vortex. For this reason, future work should focus on expanding the library of wake models available in the code to improve the representation of rotor wakes, especially in hover. The possibility of selecting among different contraction laws, axial-descent laws, and core-growth models would increase both the flexibility and the accuracy of the solver.

The treatment of sectional aerodynamic data may also be extended. For applications involving high angles of attack, local stall, or propeller off-design operation, the polar database should be extended with post-stall models. Classical extrapolation methods such as the Viterna model and the Montgomerie method represent suitable starting points for extending the aerodynamic coefficients to high angles of attack [68], [69], [70].

8.4.2 Future developments for VKI-SDPM

For VKI-SDPM, an important development concerns the treatment of the far wake. In the present implementation, all wake elements generated during the time-marching solution are explicitly convected. This approach is physically consistent, but it becomes increasingly expensive as the wake grows in time. A possible improvement would be to replace the far wake with an equivalent distribution of point doublets once the wake elements are sufficiently far from the body. The near wake would continue to be represented by wake panels, while the far wake would be described by a reduced set of singularities.

This development should be accompanied by a more robust geometrical treatment of wake–body interactions. Although the impermeability of the body is enforced through the potential-flow boundary condition, the present implementation does not prevent wake panels from geometrically intersecting the body during the time-marching evolution. Future work should therefore include an explicit geometrical non-penetration algorithm. Such an algorithm should detect intersections between wake panels and body surfaces, prevent wake singularities and far-wake point doublets from entering the solid domain, and enforce a minimum admissible clearance. The work by Oppizzi [71] represents a suitable starting point for this development.

A second natural evolution of VKI-SDPM is the development and subsequent systematic validation of multibody capabilities. The source–doublet formulation is well suited to configurations involving multiple lifting bodies, such as rotor–rotor and wing–rotor interactions, as well as lifting/non-lifting body interactions, such as rotor–fuselage and wing–fuselage configurations [72]. The current code already contains preliminary data structures that could support a future multiple-body treatment, particularly at the level of geometry assembly, body indexing, and wake management. However, these capabilities have not yet been systematically exploited or validated. The far-wake treatment through point doublets and the geometrical non-penetration algorithm described above would be particularly useful in this context, since they would reduce the computational cost of larger multibody meshes and improve the physical consistency of wake evolution near solid surfaces.

A further extension of VKI-SDPM concerns the treatment of compressibility effects. A first and relatively straightforward development would be the introduction of a Prandtl–Glauert correction. This would preserve the present incompressible potential-flow formulation while providing a first-order correction of the aerodynamic coefficients and pressure distributions in low-to-moderate subsonic conditions. A more general treatment would instead require reformulating the governing potential-flow problem to account directly for compressibility. Depending on the adopted approximation, this could range from a linearised compressible potential formulation to a non-linear full-potential model in which density becomes coupled to the local velocity field. Such an extension would substantially increase the range of operating conditions accessible to VKI-SDPM, particularly for high-speed propellers and rotors, but would also require significant modifications to the present influence formulation and solution strategy, since the linear Laplace problem on which the current implementation is based would no longer be directly applicable.

A final major extension would be the introduction of viscous–inviscid coupling. A possible implementation could couple the outer potential-flow solution with an integral boundary-layer solver. The inviscid surface-velocity distribution computed by VKI-SDPM would provide the external velocity required by the boundary-layer calculation, while the resulting displacement thickness could be used to modify the effective body geometry or boundary condition [73]. Such a coupling would extend the applicability of the solver to cases in which mild viscous effects influence the pressure field and the aerodynamic loads [5], [74]. A robust viscous treatment would therefore represent a long-term milestone for the evolution of VKI-SDPM.

Appendix A

Additional Theoretical Derivations

A.1 Derivation of Kelvin's Circulation Theorem and Helmholtz Vortex Theorems

This appendix reports the mathematical derivation of Kelvin's circulation theorem and of the Helmholtz vortex theorems. The discussion is restricted to inviscid, incompressible flow with conservative body forces, consistently with the assumptions adopted in the potential-flow formulation [5].

A.1.1 Kelvin's Circulation Theorem

Let $\mathcal{C}(t)$ be a closed material curve moving with the fluid. The circulation around $\mathcal{C}(t)$ is defined as

$$\Gamma(t) = \oint_{\mathcal{C}(t)} \mathbf{V} \cdot d\mathbf{l}. \tag{A.1}$$

To evaluate its time derivative, the curve is parametrized by a material coordinate s, so that

$$d\mathbf{l} = \frac{\partial \mathbf{x}}{\partial s} ds. \tag{A.2}$$

Since the curve is material, its points move with the local fluid velocity,

$$\frac{\partial \mathbf{x}}{\partial t} = \mathbf{V}(\mathbf{x}, t). \tag{A.3}$$

Taking the derivative of Eq. (A.1) gives

$$\frac{d\Gamma}{dt} = \frac{d}{dt} \oint_{\mathcal{C}(t)} \mathbf{V} \cdot d\mathbf{l}. \tag{A.4}$$

Using the material derivative,

$$\frac{d\Gamma}{dt} = \oint_{\mathcal{C}(t)} \left[\frac{D\mathbf{V}}{Dt} \cdot d\mathbf{l} + \mathbf{V} \cdot \frac{D(d\mathbf{l})}{Dt} \right]. \tag{A.5}$$

The material derivative of the line element is

$$\frac{D(d\mathbf{l})}{Dt} = d\mathbf{V}, \tag{A.6}$$

and therefore

$$\mathbf{V} \cdot \frac{D(d\mathbf{l})}{Dt} = \mathbf{V} \cdot d\mathbf{V} = d\left(\frac{1}{2}|\mathbf{V}|^2\right). \tag{A.7}$$

Since $\mathcal{C}(t)$ is closed, the integral of an exact differential around the contour is zero,

$$\oint_{\mathcal{C}(t)} d\left(\frac{1}{2}|\mathbf{V}|^2\right) = 0. \tag{A.8}$$

Thus,

$$\frac{d\Gamma}{dt} = \oint_{\mathcal{C}(t)} \frac{D\mathbf{V}}{Dt} \cdot d\mathbf{l}. \tag{A.9}$$

For inviscid flow, the Euler equation is

$$\frac{D\mathbf{V}}{Dt} = -\frac{1}{\rho}\nabla p + \mathbf{f}, \tag{A.10}$$

where ρ is the density, p is the pressure, and $\mathbf{f}$ is the body force per unit mass. For incompressible flow, ρ is constant. If the body force is conservative, it can be written as

$$\mathbf{f} = -\nabla\Phi, \tag{A.11}$$

where Φ is a potential. Substitution into Eq. (A.9) gives

$$\frac{d\Gamma}{dt} = -\oint_{\mathcal{C}(t)} \nabla\left(\frac{p}{\rho} + \Phi\right) \cdot d\mathbf{l}. \tag{A.12}$$

The integral of a gradient around a closed contour is zero. Therefore,

$$\frac{d\Gamma}{dt} = 0. \tag{A.13}$$

Equation (A.13) is Kelvin's circulation theorem: under the stated assumptions, the circulation around a closed material curve is conserved.

A.1.2 Vorticity Transport Equation

The vorticity is defined as

$$\boldsymbol{\zeta} = \nabla \times \mathbf{V}. \tag{A.14}$$

Taking the curl of the Euler equation gives the vorticity transport equation. For inviscid incompressible flow with conservative body forces, this equation reduces to

$$\frac{D\boldsymbol{\zeta}}{Dt} = (\boldsymbol{\zeta} \cdot \nabla)\,\mathbf{V}. \tag{A.15}$$

This relation may also be written in Eulerian form as

$$\frac{\partial \boldsymbol{\zeta}}{\partial t} + (\mathbf{V} \cdot \nabla)\,\boldsymbol{\zeta} = (\boldsymbol{\zeta} \cdot \nabla)\,\mathbf{V}. \tag{A.16}$$

The right-hand side is the vortex-stretching term. It represents the change of vorticity due to velocity gradients along the vorticity direction.

A second identity follows directly from the definition of vorticity. Since the divergence of a curl is identically zero,

$$\nabla \cdot \boldsymbol{\zeta} = \nabla \cdot (\nabla \times \mathbf{V}) = 0. \tag{A.17}$$

Equations (A.15) and (A.17) provide the mathematical basis for the Helmholtz vortex theorems.

A.1.3 Conservation of Vortex-Tube Strength

Consider a vortex tube bounded laterally by vortex lines and closed by two cross-sections $\mathcal{S}_1$ and $\mathcal{S}_2$. Since vortex lines are tangent to $\boldsymbol{\zeta}$, the vorticity flux through the lateral surface is zero. Applying the divergence theorem to the closed surface of the vortex tube gives

$$\int_{\mathcal{V}} \nabla \cdot \boldsymbol{\zeta}\, d\mathcal{V} = \int_{\partial\mathcal{V}} \boldsymbol{\zeta} \cdot \mathbf{n}\, dS = 0. \tag{A.18}$$

Using Eq. (A.17), the flux through the complete closed surface is zero. Since the contribution of the lateral surface vanishes, only the two cross-sections remain,

$$\int_{\mathcal{S}_1} \boldsymbol{\zeta} \cdot \mathbf{n}_1\, dS + \int_{\mathcal{S}_2} \boldsymbol{\zeta} \cdot \mathbf{n}_2\, dS = 0. \tag{A.19}$$

With opposite outward orientations on the two cross-sections, this implies

$$\Gamma_1 = \Gamma_2, \tag{A.20}$$

where

$$\Gamma_i = \int_{\mathcal{S}_i} \boldsymbol{\zeta} \cdot \mathbf{n} \, dS. \tag{A.21}$$

Thus, the strength of a vortex tube is constant along its length.

The same solenoidal property of vorticity also implies that a vortex tube cannot terminate inside the fluid. Suppose that a vortex tube of non-zero strength ended at an interior point of the fluid. Enclosing the endpoint by a small closed surface $\mathcal{S}$, the vorticity flux through that surface would be non-zero if the tube ended inside it. However,

$$\int_{\mathcal{S}} \boldsymbol{\zeta} \cdot \mathbf{n} \, dS = \int_{\mathcal{V}} \nabla \cdot \boldsymbol{\zeta} \, d\mathcal{V} = 0. \tag{A.22}$$

This contradiction shows that a vortex tube of finite strength cannot have an endpoint inside the fluid domain. It must either close on itself, intersect a boundary, or extend outside the domain considered.

A.1.4 Material Transport of Vortex Lines

Let $d\mathbf{l}$ be an infinitesimal material line element. Its material derivative is

$$\frac{D(d\mathbf{l})}{Dt} = (d\mathbf{l} \cdot \nabla) \, \mathbf{V}. \tag{A.23}$$

For incompressible inviscid flow, the vorticity satisfies Eq. (A.15),

$$\frac{D\boldsymbol{\zeta}}{Dt} = (\boldsymbol{\zeta} \cdot \nabla) \, \mathbf{V}. \tag{A.24}$$

Equations (A.23) and (A.24) have the same form. Therefore, if a material line element is initially parallel to the vorticity vector, it remains parallel to the vorticity vector during the motion. In other words, vortex lines are transported with the fluid.

The three results derived above correspond to the Helmholtz vortex theorems: vortex-tube strength is conserved along the tube, vortex tubes cannot terminate inside the fluid, and vortex lines move with the fluid.

A.2 Green's Identities

This appendix reports the derivation of Green's identities for the Laplace equation. The derivation follows the standard treatment used in [5].

Let $\mathcal{V}$ be a sufficiently smooth domain bounded by the surface $\partial\mathcal{V}$, and let $\mathbf{n}_\mathcal{V}$ be the outward unit normal to $\partial\mathcal{V}$. For a sufficiently smooth vector field $\mathbf{F}$, the divergence theorem states that

$$\int_\mathcal{V} \nabla \cdot \mathbf{F} \, dV = \int_{\partial\mathcal{V}} \mathbf{F} \cdot \mathbf{n}_\mathcal{V} \, dS. \tag{A.25}$$

This result is the starting point for the derivation of Green's identities.

A.2.1 Green's First Identity

Let f and g be scalar functions with continuous first and second derivatives in $\mathcal{V}$. Applying the divergence theorem to the vector field

$$\mathbf{F} = f \nabla g \tag{A.26}$$

gives

$$\int_\mathcal{V} \nabla \cdot (f \nabla g) \, dV = \int_{\partial\mathcal{V}} f \nabla g \cdot \mathbf{n}_\mathcal{V} \, dS. \tag{A.27}$$

Using the product rule,

$$\nabla \cdot (f \nabla g) = \nabla f \cdot \nabla g + f \nabla^2 g, \tag{A.28}$$

and introducing the outward normal derivative

$$\frac{\partial g}{\partial n_\mathcal{V}} = \nabla g \cdot \mathbf{n}_\mathcal{V}, \tag{A.29}$$

one obtains

$$\int_\mathcal{V} \left(\nabla f \cdot \nabla g + f \nabla^2 g \right) dV = \int_{\partial\mathcal{V}} f \frac{\partial g}{\partial n_\mathcal{V}}, dS. \tag{A.30}$$

Equation (A.30) is Green's first identity.

A.2.2 Green's Second Identity

Green's second identity is obtained by writing Eq. (A.30) twice: first for the ordered pair (f, g), and then for the ordered pair (g, f). Therefore,

$$\int_\mathcal{V} \left(\nabla f \cdot \nabla g + f \nabla^2 g \right) dV = \int_{\partial\mathcal{V}} f \frac{\partial g}{\partial n_\mathcal{V}} \, dS, \tag{A.31}$$

and

$$\int_{\mathcal{V}} \left(\nabla g \cdot \nabla f + g\nabla^2 f\right) dV = \int_{\partial\mathcal{V}} g \frac{\partial f}{\partial n_{\mathcal{V}}}\, dS. \tag{A.32}$$

Subtracting Eq. (A.32) from Eq. (A.31), and noting that

$$\nabla f \cdot \nabla g = \nabla g \cdot \nabla f, \tag{A.33}$$

gives

$$\int_{\mathcal{V}} \left(f\nabla^2 g - g\nabla^2 f\right) dV = \int_{\partial\mathcal{V}} \left(f \frac{\partial g}{\partial n_{\mathcal{V}}} - g \frac{\partial f}{\partial n_{\mathcal{V}}}\right) dS. \tag{A.34}$$

Equation (A.34) is Green's second identity.

A.2.3 Green's Third Identity

To obtain Green's third identity, one of the scalar functions in Green's second identity is chosen as the fundamental solution of the Laplace equation.

Let $\mathbf{x}$ be the field point and let $\mathbf{x}_0$ denote the source point and integration variable. Their distance is defined as

$$r = \left|\mathbf{x} - \mathbf{x}_0\right|. \tag{A.35}$$

The three-dimensional fundamental solution of the Laplace equation is

$$G(\mathbf{x}, \mathbf{x}_0) = \frac{1}{4\pi r}. \tag{A.36}$$

It satisfies, in the distributional sense,

$$\nabla_0^2 G(\mathbf{x}, \mathbf{x}_0) = -\delta(\mathbf{x} - \mathbf{x}_0), \tag{A.37}$$

where ∇_0^2 denotes the Laplacian with respect to the source coordinates $\mathbf{x}_0$, and δ is the Dirac delta distribution.

Since G is singular at $\mathbf{x}_0 = \mathbf{x}$, Green's second identity cannot be applied directly over the entire domain $\mathcal{V}$. For an interior field point $\mathbf{x} \in \mathcal{V}$, consider instead the punctured domain

$$\mathcal{V}_\varepsilon = \mathcal{V} \setminus \mathcal{B}_\varepsilon(\mathbf{x}), \tag{A.38}$$

where $\mathcal{B}_\varepsilon(\mathbf{x})$ is a sphere of radius ε centred at $\mathbf{x}$. The boundary of the punctured domain is

$$\partial\mathcal{V}_\varepsilon = \partial\mathcal{V} \cup \mathcal{S}_\varepsilon, \tag{A.39}$$

where $\mathcal{S}_\varepsilon$ is the spherical surface bounding the removed region.

On $\partial\mathcal{V}$, the outward normal to $\mathcal{V}_\varepsilon$ coincides with $\mathbf{n}_\mathcal{V}$. On $\mathcal{S}_\varepsilon$, the outward normal to the punctured domain, denoted by $\mathbf{n}_\varepsilon$, points toward the centre of the removed sphere. Therefore,

$$\mathbf{n}_\varepsilon = -\mathbf{e}_r, \qquad \mathbf{e}_r = \frac{\mathbf{x}_0 - \mathbf{x}}{r}. \tag{A.40}$$

Let

$$f(\mathbf{x}_0) = \phi(\mathbf{x}_0), \qquad g(\mathbf{x}_0) = G(\mathbf{x}, \mathbf{x}_0). \tag{A.41}$$

Within $\mathcal{V}_\varepsilon$, the fundamental solution is regular and satisfies

$$\nabla_0^2 G(\mathbf{x}, \mathbf{x}_0) = 0. \tag{A.42}$$

Applying Green's second identity over $\mathcal{V}_\varepsilon$ gives

$$\begin{aligned} -\int_{\mathcal{V}_\varepsilon} G(\mathbf{x}, \mathbf{x}_0)\nabla_0^2\phi(\mathbf{x}_0), dV_0 = &\int_{\partial\mathcal{V}} \left[\phi(\mathbf{x}_0)\frac{\partial G}{\partial n_\mathcal{V}} - G(\mathbf{x}, \mathbf{x}_0)\frac{\partial \phi}{\partial n_\mathcal{V}}\right] dS_0 \\ &+ \int_{\mathcal{S}_\varepsilon} \left[\phi(\mathbf{x}_0)\frac{\partial G}{\partial n_\varepsilon} - G(\mathbf{x}, \mathbf{x}_0)\frac{\partial \phi}{\partial n_\varepsilon}\right] dS_0. \end{aligned} \tag{A.43}$$

The contribution of the inner spherical boundary is now evaluated in the limit $\varepsilon \to 0$. On $\mathcal{S}_\varepsilon$,

$$G = \frac{1}{4\pi\varepsilon}, \qquad \nabla_0 G = -\frac{\mathbf{e}_r}{4\pi\varepsilon^2}. \tag{A.44}$$

Since $\mathbf{n}_\varepsilon = -\mathbf{e}_r$, the normal derivative of the fundamental solution is

$$\frac{\partial G}{\partial n_\varepsilon} = \nabla_0 G \cdot \mathbf{n}_\varepsilon = \frac{1}{4\pi\varepsilon^2}. \tag{A.45}$$

Because ϕ is continuous at $\mathbf{x}$, the first contribution over the inner spherical surface satisfies

$$\begin{aligned} \lim_{\varepsilon\to 0}\int_{\mathcal{S}_\varepsilon} \phi(\mathbf{x}_0)\frac{\partial G}{\partial n_\varepsilon} dS_0 &= \phi(\mathbf{x})\lim_{\varepsilon\to 0}\frac{1}{4\pi\varepsilon^2}\int_{\mathcal{S}_\varepsilon} dS_0 \\ &= \phi(\mathbf{x}). \end{aligned} \tag{A.46}$$

The second contribution vanishes because G is of order $1/\varepsilon$, the normal derivative of ϕ remains bounded, and the area of $\mathcal{S}_\varepsilon$ is of order ε^2. Therefore,

$$\lim_{\varepsilon\to 0}\int_{\mathcal{S}_\varepsilon} G(\mathbf{x}, \mathbf{x}_0)\frac{\partial \phi}{\partial n_\varepsilon} dS_0 = 0. \tag{A.47}$$

Consequently,

$$\lim_{\varepsilon \to 0} \int_{\mathcal{S}_\varepsilon} \left[\phi(\mathbf{x}_0) \frac{\partial G}{\partial n_\varepsilon} - G(\mathbf{x}, \mathbf{x}_0) \frac{\partial \phi}{\partial n_\varepsilon} \right] dS_0 = \phi(\mathbf{x}). \tag{A.48}$$

Taking the limit $\varepsilon \to 0$ in Eq. (A.43) gives

$$-\int_{\mathcal{V}} G(\mathbf{x}, \mathbf{x}_0) \nabla_0^2 \phi(\mathbf{x}_0) dV_0 = \int_{\partial \mathcal{V}} \left[\phi(\mathbf{x}_0) \frac{\partial G}{\partial n_{\mathcal{V}}} - G(\mathbf{x}, \mathbf{x}_0) \frac{\partial \phi}{\partial n_{\mathcal{V}}} \right] dS_0 + \phi(\mathbf{x}). \tag{A.49}$$

Rearranging the terms yields

$$\phi(\mathbf{x}) + \int_{\mathcal{V}} G(\mathbf{x}, \mathbf{x}_0) \nabla_0^2 \phi(\mathbf{x}_0) dV_0 = \int_{\partial \mathcal{V}} \left[G(\mathbf{x}, \mathbf{x}_0) \frac{\partial \phi}{\partial n_{\mathcal{V}}} - \phi(\mathbf{x}_0) \frac{\partial G}{\partial n_{\mathcal{V}}} \right] dS_0. \tag{A.50}$$

Equation (A.50) is Green's third identity for a field point $\mathbf{x}$ located inside $\mathcal{V}$. If ϕ is harmonic in $\mathcal{V}$, namely

$$\nabla_0^2 \phi(\mathbf{x}_0) = 0 \qquad \text{in } \mathcal{V}, \tag{A.51}$$

then Eq. (A.50) reduces to

$$\phi(\mathbf{x}) = \int_{\partial \mathcal{V}} \left[G(\mathbf{x}, \mathbf{x}_0) \frac{\partial \phi}{\partial n_{\mathcal{V}}} - \phi(\mathbf{x}_0) \frac{\partial G}{\partial n_{\mathcal{V}}} \right] dS_0, \qquad \mathbf{x} \in \mathcal{V}. \tag{A.52}$$

Equation (A.52) is the boundary representation used for potential-flow problems. When the field point approaches a smooth point of $\partial \mathcal{V}$, the corresponding limiting process gives $E(\mathbf{x}) = 1/2$, recovering the boundary form introduced in Eqs. (2.87) and (2.88).

A.3 Doublet-Panel and Vortex-Ring Equivalence

This appendix reports the mathematical derivation of the equivalence between a constant-strength doublet panel and a vortex ring distributed along its boundary. The derivation follows the potential-flow treatment of source-and-doublet panel methods [5].

Let $\mathcal{S}$ be an oriented surface with boundary $\partial \mathcal{S}$, unit normal $\mathbf{n}_0$, and constant doublet strength μ. Let $\mathbf{x}$ be the field point and $\mathbf{x}_0$ a point on the surface. Define

$$\mathbf{r} = \mathbf{x} - \mathbf{x}_0, \qquad r = |\mathbf{r}|. \tag{A.53}$$

The potential induced by a constant-strength doublet distribution over $\mathcal{S}$ may be written as

$$\phi_\mu(\mathbf{x}) = -\frac{\mu}{4\pi}\int_{\mathcal{S}} \frac{\partial}{\partial n_0}\left(\frac{1}{r}\right) dS_0, \tag{A.54}$$

where

$$\frac{\partial}{\partial n_0}\left(\frac{1}{r}\right) = \mathbf{n}_0 \cdot \nabla_0 \left(\frac{1}{r}\right). \tag{A.55}$$

Since the derivative is taken with respect to the source coordinates $\mathbf{x}_0$,

$$\nabla_0 \left(\frac{1}{r}\right) = \frac{\mathbf{r}}{r^3}, \tag{A.56}$$

and therefore Eq. (A.54) becomes

$$\phi_\mu(\mathbf{x}) = -\frac{\mu}{4\pi}\int_{\mathcal{S}} \frac{\mathbf{n}_0 \cdot \mathbf{r}}{r^3} dS_0. \tag{A.57}$$

The integral

$$I(\mathbf{x}) = \int_{\mathcal{S}} \frac{\mathbf{n}_0 \cdot \mathbf{r}}{r^3} dS_0 \tag{A.58}$$

is the solid angle subtended by $\mathcal{S}$ at the field point $\mathbf{x}$. Thus,

$$\phi_\mu(\mathbf{x}) = -\frac{\mu}{4\pi} I(\mathbf{x}). \tag{A.59}$$

The velocity induced by the doublet panel is obtained by differentiating the potential with respect to the field-point coordinates,

$$\mathbf{V}_\mu(\mathbf{x}) = \nabla \phi_\mu(\mathbf{x}) = -\frac{\mu}{4\pi} \nabla_{\mathbf{x}} I(\mathbf{x}). \tag{A.60}$$

It remains to express the gradient of the solid angle as a line integral along the boundary $\partial\mathcal{S}$. Differentiating Eq. (A.58) with respect to the field-point coordinates gives

$$\nabla_{\mathbf{x}} I(\mathbf{x}) = \int_{\mathcal{S}} \left[\frac{\mathbf{n}_0}{r^3} - 3\frac{(\mathbf{n}_0 \cdot \mathbf{r})\,\mathbf{r}}{r^5}\right] dS_0. \tag{A.61}$$

Using Stokes' theorem, the same surface integral can be written as a boundary integral,

$$\nabla_{\mathbf{x}} I(\mathbf{x}) = -\oint_{\partial\mathcal{S}} \frac{d\mathbf{l}_0 \times \mathbf{r}}{r^3}, \tag{A.62}$$

where $d\mathbf{l}_0$ is oriented consistently with the surface normal $\mathbf{n}_0$ according to the right-hand rule.

Substitution of Eq. (A.62) into Eq. (A.60) gives

$$\mathbf{V}_\mu(\mathbf{x}) = \frac{\mu}{4\pi} \oint_{\partial \mathcal{S}} \frac{d\mathbf{l}_0 \times \mathbf{r}}{r^3}. \tag{A.63}$$

The velocity induced by a closed vortex filament of circulation Γ coincident with the same boundary $\partial\mathcal{S}$ is given by the Biot–Savart law,

$$\mathbf{V}_\Gamma(\mathbf{x}) = \frac{\Gamma}{4\pi} \oint_{\partial \mathcal{S}} \frac{d\mathbf{l}_0 \times \mathbf{r}}{r^3}. \tag{A.64}$$

Comparison between Eq. (A.63) and Eq. (A.64) shows that the velocity field induced by a constant-strength doublet panel is identical to the velocity field induced by a vortex ring placed along the panel boundary, provided that

$$\Gamma = \mu. \tag{A.65}$$

A.4 BEMT Formulation

Blade Element Momentum Theory combines the sectional blade-element relations introduced in Sec. 2.3 with momentum balances applied to the flow through the rotor disk. Historically, this formulation is associated with the combination of Rankine–Froude actuator-disk theory and blade-element theory, later formalized in the classical work of Glauert [75], [76] [33], [77] [34] [78].

The purpose of this appendix is to summarize the mathematical formulation used to obtain the axial and tangential induction factors. The derivation is written for a propulsive convention, in which the axial induced velocity increases the axial velocity through the disk. Other conventions, such as those commonly adopted for wind turbines, lead to equivalent equations after a consistent change of sign.

A.4.1 Actuator-Disk and Annular Momentum Relations

Momentum theory models the rotor as an actuator disk that introduces a pressure jump in the flow. In its simplest form, the flow is assumed inviscid, incompressible, axisymmetric, and steady, and the disk is assumed to act on a streamtube passing through the rotor plane [3], [79].

Let V_∞ be the far-upstream axial velocity and let V_{ax} be the axial velocity at the rotor plane. The axial induction factor a_{ind} is defined, for the propulsive convention adopted here, as

$$V_{\mathrm{ax}} = V_\infty \left(1 + a_{\mathrm{ind}}\right). \tag{A.66}$$

The corresponding far-wake axial velocity is

$$V_{\mathrm{w}} = V_\infty \left(1 + 2a\right), \tag{A.67}$$

which follows from the standard actuator-disk result stating that the velocity at the disk is the average between the far-upstream and far-wake velocities.

For an annular streamtube of radius r and thickness dr, the differential area is

$$dS = 2\pi r\, dr. \tag{A.68}$$

Including the tip factor F_{tip}, the axial momentum balance gives the elemental thrust

$$dT_{\mathrm{mom}} = 4\pi\rho F_{\mathrm{tip}} V_\infty^2 a_{\mathrm{ind}} \left(1 + a_{\mathrm{ind}}\right) r\, dr. \tag{A.69}$$

To include torque, the annular momentum balance is extended to account for slipstream rotation [79], [80]. The upstream flow is assumed to have no tangential velocity, while the rotor induces swirl in the wake. The tangential induction factor a'_{ind} is defined through the relative tangential velocity at the rotor plane as

$$V_{\tan} = \Omega r \left(1 - a'_{\mathrm{ind}}\right), \tag{A.70}$$

where Ω is the angular velocity of the rotor. With the same annular-streamtube approximation, the angular momentum balance gives

$$dQ_{\mathrm{mom}} = 4\pi\rho F_{\mathrm{tip}} V_\infty \Omega a'_{\mathrm{ind}} \left(1 + a_{\mathrm{ind}}\right) r^3\, dr. \tag{A.71}$$

Equations (A.69) and (A.71) provide the momentum-theory relations to be coupled with the blade-element sectional loads.

A.4.2 Blade-Element Closure

For a blade section at radius r, the relative velocity is decomposed into axial and tangential components,

$$V_{\mathrm{rel}} = \sqrt{V_{\mathrm{ax}}^2 + V_{\tan}^2}, \tag{A.72}$$

and the local inflow angle is

$$\tan\varphi = \frac{V_{\mathrm{ax}}}{V_{\tan}} = \frac{V_\infty \left(1 + a_{\mathrm{ind}}\right)}{\Omega r \left(1 - a'_{\mathrm{ind}}\right)}. \tag{A.73}$$

The angle of attack is obtained from the local blade pitch angle θ as

$$\alpha = \theta - \varphi. \tag{A.74}$$

The sectional aerodynamic coefficients are obtained from the airfoil polar data,

$$C_l = C_l(\alpha), \qquad C_d = C_d(\alpha). \tag{A.75}$$

For the present propulsive convention, the normal and tangential force coefficients are defined as

$$C_T = C_l \cos\varphi - C_d \sin\varphi, \tag{A.76}$$

and

$$C_{\text{tan}} = C_l \sin\varphi + C_d \cos\varphi. \tag{A.77}$$

The local blade solidity is defined as

$$\sigma' = \frac{N_b c}{2\pi r}, \tag{A.78}$$

where N_b is the number of blades and $c = c(r)$ is the local chord.
The elemental thrust and torque from blade-element theory are

$$dT_{\text{BE}} = \frac{1}{2}\rho V_{\text{rel}}^2 N_b c C_T \, dr, \tag{A.79}$$

and

$$dQ_{\text{BE}} = \frac{1}{2}\rho V_{\text{rel}}^2 N_b c C_{\text{tan}} r \, dr. \tag{A.80}$$

These two relations provide the blade-element closure required to determine the induction factors.

A.4.3 Induction Factors and Scalar Residual Formulation

The BEMT equations are obtained by equating the momentum and blade-element expressions for thrust and torque. Using

$$V_{\text{rel}} = \frac{V_\infty (1 + a_{\text{ind}})}{\sin\varphi}, \tag{A.81}$$

and equating Eq. (A.69) with Eq. (A.79) gives

$$4F a_{\text{ind}} = \sigma' C_T \frac{1 + a_{\text{ind}}}{\sin^2\varphi}. \tag{A.82}$$

Solving for a_{ind} yields

$$a_{\text{ind}}(\varphi) = \left(\frac{4F \sin^2\varphi}{\sigma' C_n} - 1 \right)^{-1}. \tag{A.83}$$

Similarly, using

$$V_{\text{rel}}^2 = \frac{V_\infty\,(1+a_{\text{ind}})\,\Omega r\,(1-a'_{\text{ind}})}{\sin\varphi\cos\varphi}, \tag{A.84}$$

and equating Eq. (A.71) with Eq. (A.80) gives

$$4Fa'_{\text{ind}} = \sigma' C_{\tan}\frac{1-a'_{\text{ind}}}{\sin\varphi\cos\varphi}. \tag{A.85}$$

Solving for a'_{ind} gives

$$a'_{\text{ind}}(\varphi) = \left(\frac{4F\sin\varphi\cos\varphi}{\sigma' C_{\tan}} + 1\right)^{-1}. \tag{A.86}$$

The coupled BEMT problem can therefore be reduced to a scalar nonlinear problem in the inflow angle φ. Following the formulation of Ning [2], the induction factors are treated as explicit functions of φ, through Eqs. (A.83) and (A.86). Substitution into the kinematic relation in Eq. (A.73) gives the residual

$$\mathcal{R}(\varphi) = \frac{\sin\varphi}{1+a_{\text{ind}}(\varphi)} - \frac{V_\infty}{\Omega r}\frac{\cos\varphi}{1-a'_{\text{ind}}(\varphi)} = 0. \tag{A.87}$$

Once Eq. (A.87) is solved for φ, the induction factors follow from Eqs. (A.83) and (A.86). The local aerodynamic coefficients are then evaluated from the corresponding angle of attack, and the elemental loads are obtained from Eqs. (A.79) and (A.80).

Appendix B

VKI-BELL Code Reference

This appendix provides a practical reference for the MATLAB implementation of VKI-BELL used in the present thesis. The theoretical formulation of the code has been discussed in Chapter 3. The purpose of this appendix is instead to document how the code is organised, how a simulation is launched, which files are normally edited by the user, and how the saved results are structured.

The guide refers to the version of VKI-BELL used for the simulations reported in this work. Most variable names match those used in Chapter 3; where the two differ, for instance the code refers to the outer-loop thrust coefficient as cfg.wake.CTcurrent where the chapter uses $C_{\mathrm{T,w}}$, the naming follows the actual MATLAB implementation and should be used as the practical reference for code usage.

B.1 Code Structure and Workflow Map

The source files are grouped according to the main solver tasks. The most relevant folders are:

$$\textsf{configuration}, \qquad \textsf{core}, \qquad \textsf{blade element}, \qquad \textsf{lifting line}. \tag{B.1}$$

Each folder corresponds to a block of the implementation described in Chapter 3: configuration holds the case-specific input files; core contains rotor-geometry generation, wake construction, the fixed-point solver, and load evaluation; blade element contains the sectional aerodynamic closure; lifting line contains the vortex-segment assembly and the Biot–Savart routines. The remaining files, including the main driver main_BELL.m and the launch script run_BELL.m reside in the project root.

Unlike a time-marching solver, VKI-BELL does not iterate over physical time: the driver executes an inner fixed-point loop that solves for the circulation at a frozen wake and an optional outer loop that reconciles the wake shape with the converged solution. The block sequence executed by main_BELL.m is:

```
% Block 1: configuration starting
cfg = feval(config_name);   % cfg = simulation function evaluation
cfg = validate_config(cfg); % cfg input validation
% Block 2: rotor/body geometry
build_rotor_geometry.m
% Block 3: polar data
load_polars.m
% Block 4: initial wake geometry
build_wake_geometry.m
% Block 5: steady solver (inner + outer loop)
run_solver.m
% Block 6: post-processing + run summary
compute_loads.m
% Block 7: diagnostic figures ('explain' mode only)
```

Listing B.1: Configuration loading and workflow blocks in main_BELL.m.

The configuration is a MATLAB function whose name coincides with the file name, so adding a new case only requires writing a new file in configuration folder, without modifying the driver.

In Lst. B.2, the $\boldsymbol{M}$ matrix assembly and circulation update are reported within the context of the inner-loop contained in run_solver.m.

```
[Mx, My, Mz] = build_influence_matrix(bladeGeom, wakeGeom, cfg);
for iter = 1:maxIter
    g = Gamma(:);
    Vind_CP = reshape([Mx*g, My*g, Mz*g], nBlades, nPanels, 3);
    [Gamma_target, sectionData] = compute_section_aerodynamics( ...
        bladeGeom, Vind_CP, polarData, cfg);
    delta_Gamma = Gamma_target - Gamma;
    residual    = max(abs(delta_Gamma(:))) / (mean(abs(Gamma_target(:))) + eps);
    Gamma       = Gamma + relax * delta_Gamma;
end
```

Listing B.2: Influence-matrix assembly and inner-loop circulation update in run_solver.m.

If the outer loop is enabled (always in hover, optional in axial flight), the converged inner solution is used to update the mean induced velocity and, if cfg.wake.CTadaptive is set, the thrust coefficient feeding the empirical wake-shape laws; the wake is then rebuilt and the inner loop is solved again:

```
for outer = 1:maxOuter
    [Gamma, sectionData, history, wakeGeom] = ...
        run_solver(bladeGeom, wakeGeom, polarData, cfg);
    viComp_outer = sum(sectionData.Vind .* reshape(
   eOmega_outer,1,1,3), 3);
    if cfg.wake.CTadaptive
        loads_outer = compute_loads(bladeGeom, sectionData,
   Gamma, cfg);
        cfg.wake.CTcurrent = cfg.wake.CTcurrent + cfg.wake.
   CTadaptRelax * ...
            (loads_outer.CT - cfg.wake.CTcurrent);
    end
    vi_new   = max(mean(viComp_outer(:)), 1e-3);
    vi_outer = vi_outer + relaxVi * (vi_new - vi_outer);
    Vind_edges_outer = repmat(reshape(vi_outer*eOmega_outer
   ,1,1,3), ...
        nBlades_outer, nRedges_outer, 1);
    wakeGeom = build_wake_geometry(bladeGeom,
   Vind_edges_outer, cfg);
end
```

Listing B.3: Outer wake-reconciliation loop in main_BELL.m.

A compact map of the solver routines is reported in Table B.1. The table is intended as a navigation aid for the code and as a link between the implementation files and the corresponding parts of the thesis.

Table B.1: Workflow-to-code mapping of the VKI-BELL solver.

Workflow block	**Main role**	**Main routines**
Simulation setup	Case parameters, execution options, and field validation with defaults	main_BELL.m run_BELL.m <case_name>.m validate_config.m
Rotor geometry	Blade panels, control points, local frames, and radial-edge geometry	build_rotor_geometry.m
Polar data	Loading and angle-of-attack interpolation of the sectional aerodynamic polars	load_polars.m lookup_polar.m
Initial wake geometry	Prescribed helical wake from an initial induced-velocity guess	build_wake_geometry.m apply_wake_contraction.m
Inner fixed-point loop (frozen wake)		
Influence-matrix assembly	Vortex-segment topology and Biot–Savart kernel evaluated once per outer iteration	assemble_segments.m build_influence_matrix.m compute_biot_savart_kernel.m
Induced velocity	Control-point induced velocity from the influence matrix, $\mathbf{V}_{\text{ind,CP}} = \mathbf{M}\,\Gamma$	run_solver.m (Step 1)
Sectional aerodynamics	Angle of attack, polar lookup, and target circulation from the Kutta–Joukowsky theorem	compute_section_aerodynamics.m
Circulation update	Residual evaluation and under-relaxation of the bound circulation	run_solver.m (Steps 3–4)
Outer wake-reconciliation loop (optional)		
Induced-velocity / C_T update	Mean or local induced velocity and, if enabled, the adaptive thrust coefficient feeding the wake-shape laws	main_BELL.m (Block 5)
Wake rebuild	Reconstruction of the prescribed wake with the updated velocity and/or C_T	build_wake_geometry.m apply_wake_contraction.m
Post-processing		
Load evaluation	Sectional and integrated rotor loads, performance coefficients	compute_loads.m
Result storage	Packing and saving of the results structure	save_BELL_results.m
Diagnostics and plotting	Rotor and wake visualisation, load and residual-history plots	plot_rotor_3d.m plot_radii_twist_distribution.m plot_wake_panels_3d.m plot_loads.m plot_solver_history.m
Post-processing	Reload a saved run and recompute or re-plot loads	postprocess_BELL.m

B.2 Simulation Main Files

The standard execution path assumes that the code is launched from the main VKI-BELL folder. main_BELL.m itself adds all subfolders to the MATLAB path at the top of the function, so the geometry, wake, aerodynamic, and diagnostic routines become available without any manual configuration.

The test case is selected by passing the name of the configuration function, as a string, to the driver:

```
results = main_BELL('template_test');
```

Listing B.4: Standard launch call, as used in run_BELL.m.

The driver calls this function directly with feval: there is no internal case list to extend. Parameters sweep can be performed by passing a pre-built cfg struct instead of a name.

B.2.1 Input File Template

A VKI-BELL simulation is controlled through a case-specific configuration file. The file name and the function it defines must coincide,

$$\textsf{<case\_name>.m} \quad \Longrightarrow \quad \textsf{function cfg} = \textsf{<case\_name>()}, \tag{B.2}$$

and the file must be placed in the configuration folder. The configuration fields are grouped into seven categories: geometry, operatingPoint, wake, vortex, numerics, aero, and results.

Listing B.5 reports an abbreviated excerpt of template_config.m, the reference template distributed with the code. The excerpt is not intended to reproduce the full, extensively commented file, but to show the structure of the input file and the main groups of parameters expected by the driver.

```
function cfg = template_config()
% ============================================================
% GEOMETRY
% ============================================================
cfg.geometry.nBlades          = 2;    % number of blades
cfg.geometry.nPanels          = 20;   % spanwise panels
cfg.geometry.spanDistribution = 'cosine';   % 'uniform' | 'cosine'
cfg.geometry.radii      = [];     % [m] array of radial stations, root to tip
cfg.geometry.chords     = [];     % [m] array of chords at each station
cfg.geometry.twist_deg  = [];     % [deg] array of pitch angle at each station
```

```
cfg.geometry.collectivePitch_deg    = 0.0;
cfg.geometry.pitchAxisChordFraction = 0.30;
cfg.geometry.airfoilNames = {};  % .dat string array for airfoil names
% ==========================================================
% OPERATING POINT
% ==========================================================
cfg.operatingPoint.OmegaVector_rpm = [0, 0, 0];   % [rpm]
cfg.operatingPoint.VinfVector      = [0, 0, 0];   % [m/s]
cfg.operatingPoint.hubPosition     = [0, 0, 0];   % [m]
cfg.operatingPoint.blade1RefVector = [0, 0, 0];   % radial vector for the reference blade
% ==========================================================
% WAKE
% ==========================================================
cfg.wake.model             = 'helical';
cfg.wake.nTurns            = 12;      % wake turns
cfg.wake.nSegmentsPerTurn = 36;      % azimuthal discretisation per turn
cfg.wake.contractionModel = 'none'; % 'none' | 'landgrebe' | 'mccormick'
cfg.wake.tipAxialModel     = 'convection'; % 'convection' | 'landgrebe'
cfg.wake.farWakeFactor     = 2.0;
cfg.wake.CTadaptive        = false;  % see wake-shape laws, Sec. 4.4 (chapter)
cfg.wake.CTadaptRelax      = 0.4;
% ==========================================================
% VORTEX
% ==========================================================
cfg.vortex.coreRadiusWake   = 0.005;      % [m] emission core radius
cfg.vortex.coreSpreadDelta = 100;         % 0 = constant core
cfg.vortex.regularization   = 'lambOseen';
% ==========================================================
% NUMERICS
% ==========================================================
cfg.numerics.maxIter     = 400;
cfg.numerics.tolerance   = 1e-4;
cfg.numerics.relaxation = 0.2;
cfg.numerics.OuterLoop   = true;
cfg.numerics.gammaInit   = 'aeroInit';    % 'zero' | 'aeroInit', initial guess on Gamma
% ==========================================================
% AERO
```

```
% ============================================================
cfg.aero.polarDir      = './airfoil polars';
cfg.aero.extrapPolicy = 'flatPlate';    % 'clamp'|'
    extrapolate'|'flatPlate'
% ============================================================
% RESULTS SAVING
% ============================================================
cfg.results.name       = 'template_test_case';
cfg.results.autoSave   = true;
cfg.results.overwrite = true; % false = timestamp is
    appended
```

Listing B.5: VKI-BELL input file template.

B.2.2 Output File Template

At the end of the simulation, if cfg.results.autoSave is true, save_BELL_results.m writes a MATLAB .mat file containing the results structure. The destination folder and base file name are read from cfg.results.dir and cfg.results.name.

The main fields are summarised in Listing B.6. The field results.cfg stores the validated configuration file, including the defaults filled in by validate_config.m. The fields results.bladeGeom and results.wakeGeom store the blade and wake geometry used for the last inner-loop solution, results.sectionData stores the sectional flow and aerodynamic quantities, and results.loads stores the integrated and radial performance quantities.

```
% Main saved variable
results.cfg           % validated configuration
results.bladeGeom     % blade-panel geometry, control points,
     local frames
results.wakeGeom      % final wake node lattice and panels
results.Gamma         % converged bound-circulation
    distribution [nBlades x nPanels]
results.history       % inner-loop convergence: residual,
    nIter, converged
results.sectionData   % sectional flow and aerodynamic
    quantities:
results.loads         % integrated and radial rotor loads
```

Listing B.6: Typical structure of the saved VKI-BELL output file.

Appendix C

VKI-SDPM Code Reference

This appendix provides a practical reference for the MATLAB implementation of VKI-SDPM used in the present thesis. The theoretical formulation of the source–doublet panel method, the wake model, and the load-evaluation procedure have been discussed in the main chapters. The purpose of this appendix is instead to document how the code is organised, how a simulation is launched, which files are normally modified by the user, and how the saved results are converted into post-processing and validation quantities.

The guide refers to the version of VKI-SDPM used for the simulations reported in this work. Some parameter and variable names may not exactly match those used in Chapter 5, where a more homogeneous notation was adopted for clarity. In this appendix, the naming follows the actual MATLAB implementation and should therefore be used as the practical reference for code usage.

C.1 Code Structure and Workflow Map

The source files are grouped according to the main solver tasks. The most relevant folders are:

$$\mathsf{geometry}, \qquad \mathsf{solver}, \qquad \mathsf{influence}, \qquad \mathsf{wake}, \qquad \mathsf{loads}, \qquad \mathsf{post\ process}. \tag{C.1}$$

Each folder corresponds to a block of the implementation described in Chapter 5. The main driver calls these routines in a fixed order: geometry generation, body influence assembly, wake initialisation, time marching, pressure evaluation, result storage, and post-processing.

The geometry block is selected according to the simulation class:

```
switch lower(cfg.sim_class)
case 'wing'
    semispan    = feval(cfg.geometry_func, cfg.geometry_params{:});
    wing        = assemble_wing(semispan);
    panels_all = build_panel_geometry(wing);
case 'rotor'
    blade       = feval(cfg.geometry_func, cfg.geometry_params{:});
    rotor       = assemble_propeller(blade, cfg.Nb);
    panels_all = build_panel_geometry(rotor);
end
```

Listing C.1: Geometry selection in the main driver.

The geometry functions generate the raw rotor or wing geometry, while build_panel_geometry.m converts it into the panel structure used by the solver.

The body influence matrices are computed immediately after geometry generation.

The time loop contains the gathers the body motion, source update, wake shedding, wake influence, algebraic solver, wake convection, and surface computations. A compact map of the main solver routines is reported in Table C.1. The table is intended as a navigation aid for the code and as a link between the implementation files and the corresponding parts of the thesis. Here, follows parts of the code that summarize the time loop:

```
if strcmp(simulation_class, 'rotor')
panels_all = update_rotor_position(panels_all, dphi_deg);
end
Q_now      = Q_inf_fun(t_n);
Omega_now = Omega_fun(t_n);
sigma_now = compute_sources(panels_all, Q_now, Omega_now);
wake   = create_wake_panels_exp(wake, panels_all, mu_prev);
C_phi = compute_C_phi_vectorial(panels_all, wake);
RHS     = - A_phi * sigma_now - C_phi * mu_wake_vec;
mu_now = U_phi \ (L_phi \ (P_phi * RHS));
```

Listing C.2: Linear system assemble and solution

The wake is then convected according to the selected wake mode. In free-wake mode, the induced velocity is first evaluated on the wake nodes. In prescribed-wake mode, the same routine is used, but the induced-velocity field is set to zero and the wake is convected only by the freestream contribution.

```
if strcmp(lower(wake_mode), 'free')
V_wake_nodes = compute_on_wake_velocities_vectorial(wake,
   panels_all, mu_now, sigma_now);
wake = convect_wake(wake, V_wake_nodes, dt, time_int_method,
   t_step, Q_now);
end
```

Listing C.3: Free-wake induced-velocity evaluation and convection.

After the body solution is obtained, the surface velocity and pressure are computed and stored:

```
q_surf = compute_surface_velocity(panels_all, mu_now,
   sigma_now, Nc);
p_surf = compute_pressure(panels_all, q_surf, ...
mu_now, mu_prev, mu_prev2, ...
rho, Q_now, Omega_now, dt, t_step, ...
dphi_dt_method, p_ref);
```

Listing C.4: Surface velocity and pressure evaluation.

Table C.1: Workflow-to-code mapping of the VKI-SDPM solver.

Workflow block	**Main role**	**Main routines**
Simulation setup	Case parameters and execution options	Main_SDPM.m sim_config_*.m
Geometry generation	Panel mesh, collocation points, normals, and areas	assemble_propeller.m assemble_wing.m build_panel_geometry.m
Body AIC assembly	Body source and doublet influence matrices	compute_A_phi_LS_vectorial.m compute_B_phi_LS_vectorial.m assemble_matrices.m
Initial body solution	Initial source vector and body doublet distribution	compute_sources.m LU solve
Wake initialisation	Initial trailing-edge wake structure	initialize_wake.m
	Time-marching loop	
Body kinematics	Current body position and angular velocity	update_rotor_position.m
Source update	Current source distribution from the non-penetration condition	compute_sources.m
Wake shedding	Newly shed wake panels and wake doublet strengths	create_wake_panels_exp.m
Wake AIC update	Current wake influence matrix on the body collocation points	compute_C_phi_vectorial.m
Linear solve	Current body doublet distribution	LU solve
Wake vortex segments	Equivalent vortex-segment wake network	update_wake_vortex_segments.m
Wake induced velocity	Induced velocity on wake nodes	compute_wake_velocities_vectorial.m
Wake convection	Updated wake-node positions	convect_wake.m
Surface velocity	Surface perturbation velocity on body panels	compute_surface_velocity.m
Pressure evaluation	Dimensional pressure field on body panels	compute_pressure.m
Data storage	Time-step quantities saved in the result structure	results save
	Post-processing	
Pressure coefficient	Conversion from dimensional pressure to C_p	compute_Cp.m
Load integration	Integrated forces, moments, and aerodynamic coefficients	compute_forces_moments.m
Plotting routines	Pressure, loading, convergence, and wake visualisation	plot_Cp_sections.m plot_spanwise_loading.m plot_convergence.m plot_wake_animation.m

C.2 Simulation Main Files

The standard execution path assumes that the code is launched from the main VKI-SDPM folder. At startup, the driver adds all subfolders to the MATLAB path making the geometry, influence, wake, load, utility, and post-processing routines available without manually adding each folder.

The test case is selected in the first block of Main_SDPM.m by changing the string test_case. The corresponding configuration function is then called through a switch statement based on the selected string. The user can modify the simulation parameters from the case related sim_config_*.m file.

The code uses MATLAB parfor feature to access to parallel loops for the computation of AIC matrices and to convect the wake in the free-wake simulations. Therefore, the MATLAB parallel toolbox is necessary to run the code.

The driver initialises the parallel pool and the user can select whether the pool is based on MATLAB threads or processes. The processes have to be preferred in case of large computations.

The main execution options that are independent of the physical test case are collected in the first block of the driver. In the current version, these include saving options, parallel-pool type, and the list of bodies allowed to shed wake. The field use_holder is retained for future interacting-body extensions. In the simulations reported in this thesis, the active workflow uses lifting surfaces only.

```
mode_save            = true;
pool_type            = 'Processes';
use_holder           = false;
shedding_bodies      = [1];
```

Listing C.5: Execution options defined in the driver.

C.2.1 Input File Template

A VKI-SDPM simulation is controlled through a case-specific configuration file

$$\text{sim\_config\_*.m.} \tag{C.2}$$

In the standard workflow, the user selects the desired test case in Main_SDPM.m, the corresponding configuration file is loaded, the simulation is executed, and the output is saved as a MATLAB .mat file.

Listing C.6 reports a representative template for a rotor simulation of a VKI-SDPM configuration file. The template is not intended to reproduce every validation case, but to show the structure of the input file and the main groups of parameters expected by the driver.

```
function cfg = sim_config_template()
% ==========================================================
% CASE IDENTIFICATION
% ==========================================================
cfg.test_case     = 'template_case';
cfg.sim_class     = 'rotor';    % 'rotor' | 'wing'
cfg.geometry_func = 'template_geometry_function';
% ==========================================================
% GEOMETRY AND MESH
% ==========================================================
cfg.R              = 1.0;          % rotor radius  [m]
cfg.r_R_start    = 0.15;          % inner cutout for rotor [-]
cfg.chord_rotor = 0.10;          % rotor reference chord [m]
cfg.collective_deg = 5.0;        % collective pitch [deg]
cfg.Nb = 2;          % number of blades
cfg.Nc = 40;         % chordwise panels per surf side
cfg.Nr = 20;         % spanwise panels per blade
cfg.chord_spacing  = 'cosine'; % 'uniform' | 'cosine' | '
    sine'
cfg.span_spacing   = 'cosine'; % 'uniform' | 'cosine'
cfg.rotor_axis     = [0.0, 0.0, 1.0];
cfg.moment_ref_point = [0.0, 0.0, 0.0];
% ==========================================================
% OPERATING CONDITIONS
% ==========================================================
cfg.RPM       = 1000.0;                    % rotor speed [rev/min]
cfg.Omega_z   = cfg.RPM * 2*pi/60;     % angular speed [rad/s]
cfg.V_tip     = cfg.Omega_z * cfg.R;  % tip speed [m/s]
cfg.rho       = 1.225;                  % density [kg/m^3]
cfg.p_ref     = 101325.0;             % reference pressure [Pa]
cfg.q_ref     = 0.5 * cfg.rho * cfg.V_tip^2;
cfg.V_ref     = cfg.V_tip;
cfg.c_ref     = cfg.chord_rotor;
cfg.c_mean    = cfg.chord_rotor;
% Freestream and angular-velocity functions.
cfg.Q_inf_fun = @(t) [0.0, 0.0, 0.0];
cfg.Omega_fun = @(t) [0.0, 0.0, cfg.Omega_z];
cfg.Q_inf_vec = cfg.Q_inf_fun(0.0);
% ==========================================================
% WAKE-LENGTH AND TIME-STEP PARAMETERS
% ==========================================================
cfg.N_wake_length = 6;      % rotor: revolutions;
cfg.solver.dphi_deg = 15.0; % azimuthal step [deg]
% ==========================================================
```

```
% SOLVER SETTINGS
% ============================================================
cfg.solver.wake_mode        = 'free';         % 'free' | '
    prescribed'
cfg.solver.time_int_method = 'ET';           % Time
    integration scheme
cfg.solver.vortex_core_model  = 'vatistas'; % 'none' | '
    lamb_oseen' | 'vatistas'
cfg.solver.core_evolution     = 'wake_age'; % 'constant' | '
    stretching' | 'wake_age' | 'stretching_age'
cfg.solver.init_core_size_factor = 0.10;     % initial core
    factor (%c)
cfg.solver.nu_eff                = 1.46e-5; % kinematic
    viscosity [m^2/s]
cfg.solver.turbulent_viscosity_model = 'constant'; % '
    constant' | 'squire'|'stepped'
% ============================================================
% OUTPUT PATHS
% ============================================================
cfg.solver.save_path = fullfile('data', '
    stored_numerical_results', ...
'template_case', ...
'template_result.mat');
```

Listing C.6: Representative template of a VKI-SDPM simulation configuration file.

C.2.2 Output File Template

At the end of the simulation, the main output is a MATLAB .mat file, that contains the structure results. The main fields stored here are summarised in Listing C.7. The field results.params stores the input configuration together with derived quantities such as the number of panels, the time step, and the number of simulated time steps. The arrays results.mu, results.sigma, results.p_surf, and results.q_surf contain the main time-dependent solution variables. The optional field results.wake stores the entirety or just snapshots of the wake.

```
% Main saved variable
% Simulation parameters and results
results.params  % copy of cfg plus derived fields
results.panels  % body-panel structure
results.t       % time vector
results.mu      % body doublet strengths
results.sigma   % source strengths
results.p_surf  % surface dimensional pressure
results.q_surf  % surface perturbation velocity
% Wake data
results.wake.dt
results.wake.Nr
results.wake.nodes
results.wake.n_rows
results.wake.circulation
results.wake.core_radii
```

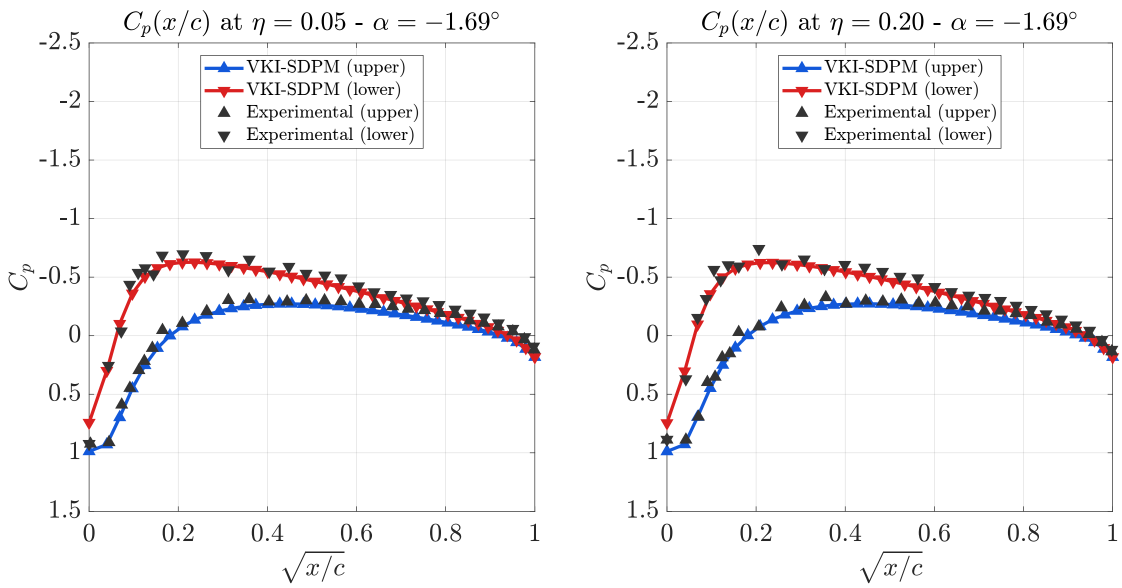
Listing C.7: Typical structure of the saved VKI-SDPM output file.

Appendix D

Extra Validation Results for VKI-SDPM

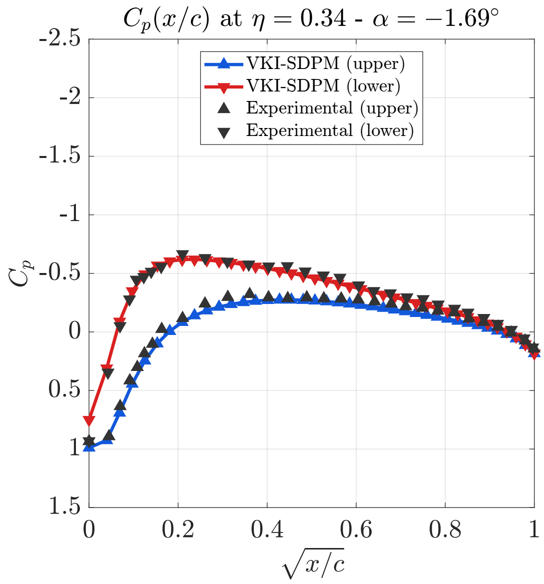
D.1 Yip-Shubert Test Case

D.1.1 C_p Distributions for $\alpha = -1.69°$ and $q_\infty = 1.47$ kPa

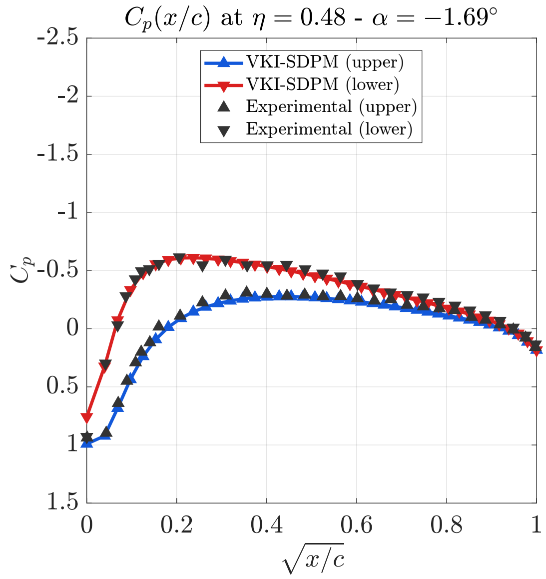


(a) $\eta = 0.05$ **(b)** $\eta = 0.20$

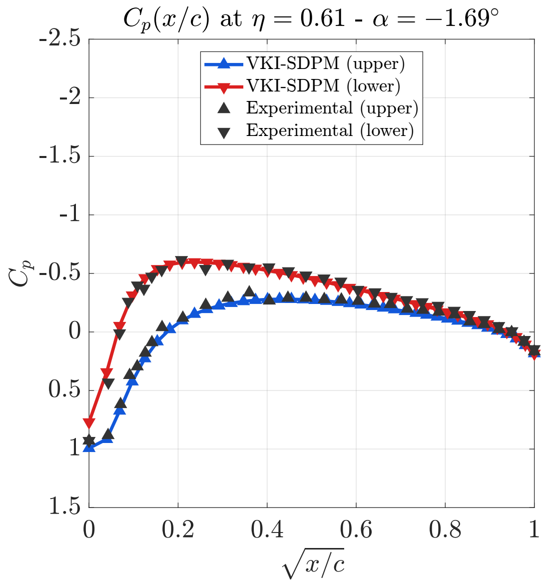
Figure D.1: Experimental and numerical comparison of C_p distributions for the Yip–Shubert test case at $\alpha = -1.69°$ ($\eta = 0.05$, 0.20).



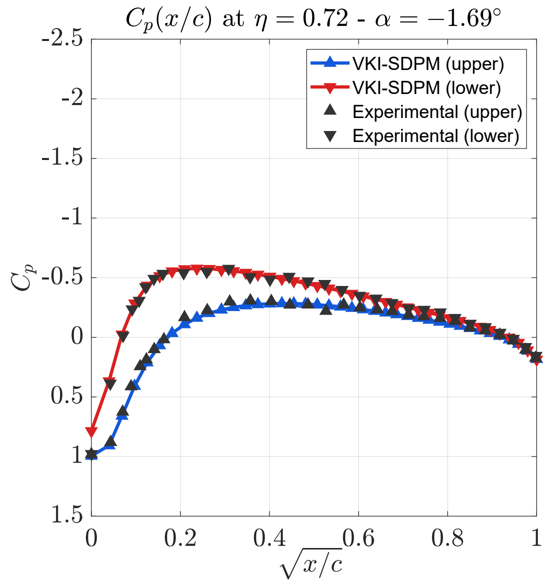
(c) $\eta = 0.34$



(d) $\eta = 0.48$

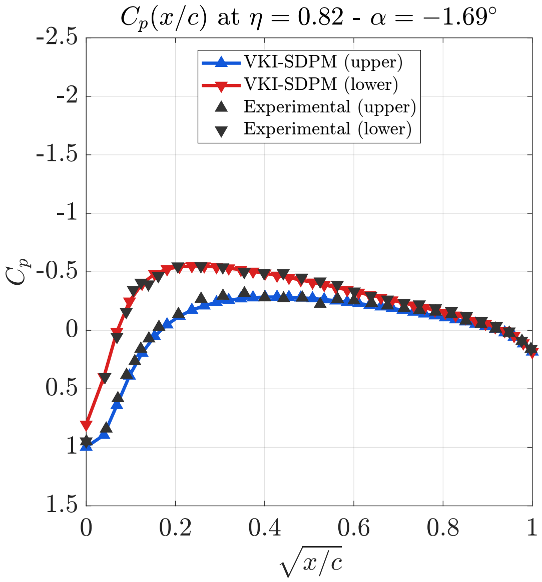


(e) $\eta = 0.61$

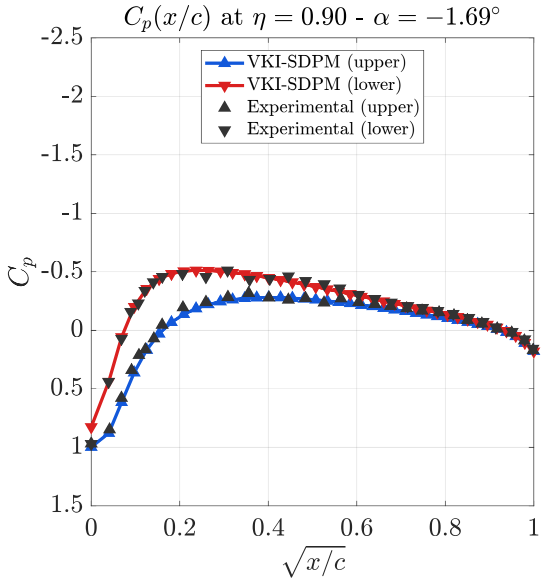


(f) $\eta = 0.72$

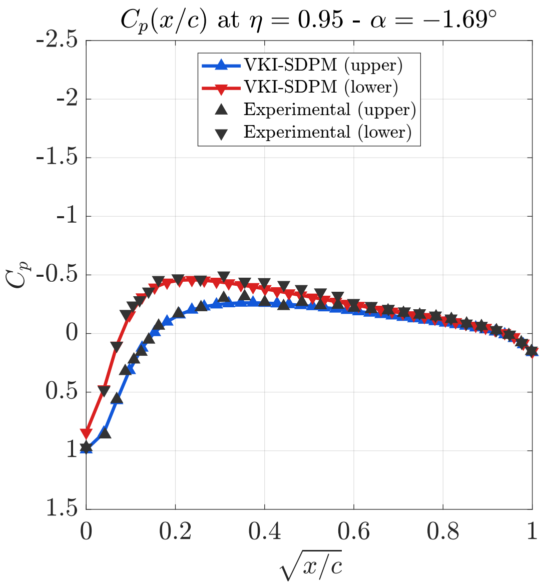
Figure D.1 (continued) ($\eta = 0.34,\ 0.48,\ 0.61,\ 0.72$).



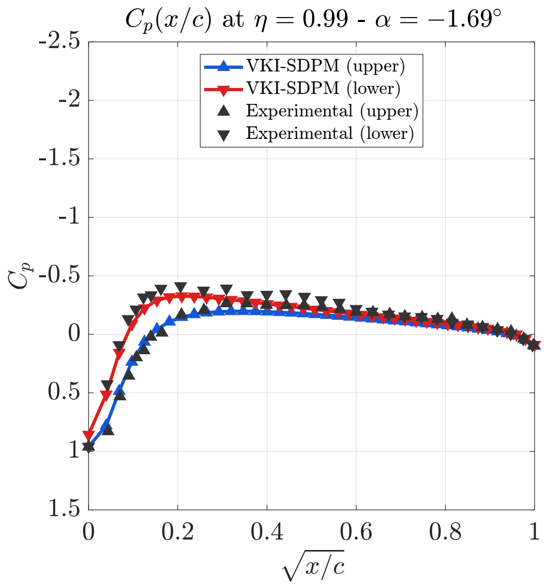
(g) $\eta = 0.82$



(h) $\eta = 0.90$



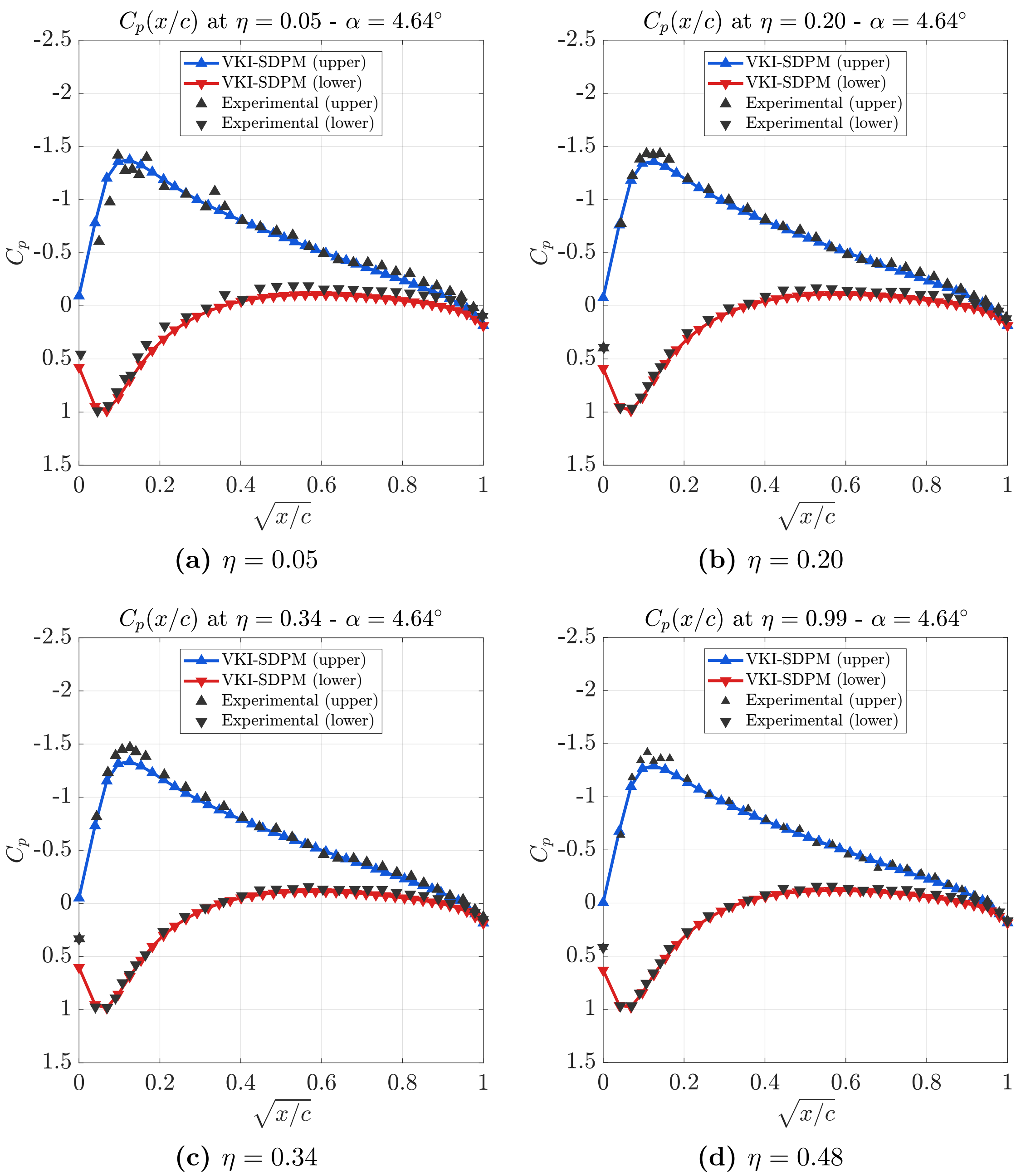
(i) $\eta = 0.95$



(j) $\eta = 0.99$

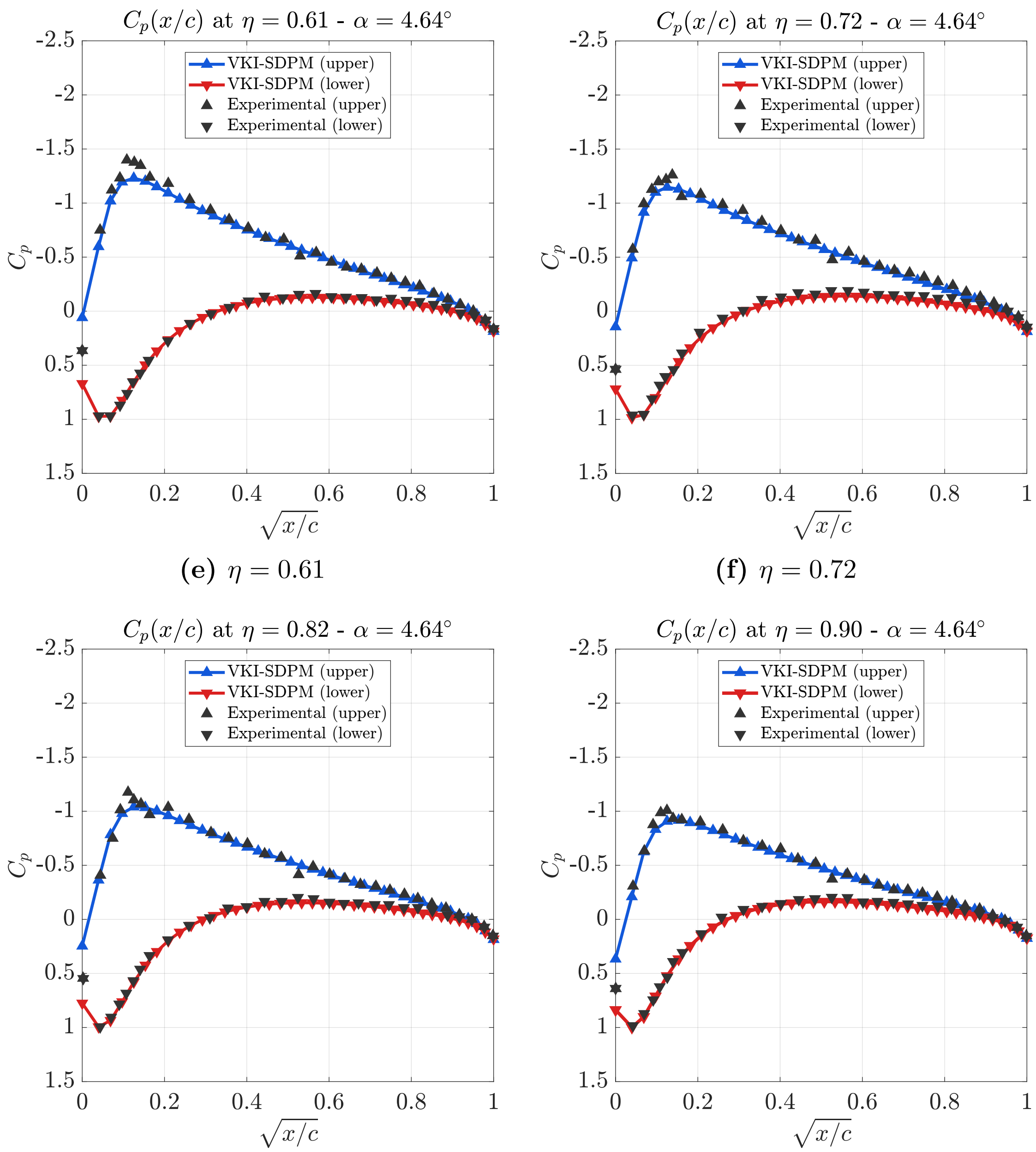
Figure D.1 (continued) ($\eta = 0.82$, 0.90, 0.95, 0.99).

D.1.2 C_p Distributions for $\alpha = 4.64°$ and $q_\infty = 1.48$ kPa



(a) $\eta = 0.05$ **(b)** $\eta = 0.20$ **(c)** $\eta = 0.34$ **(d)** $\eta = 0.48$

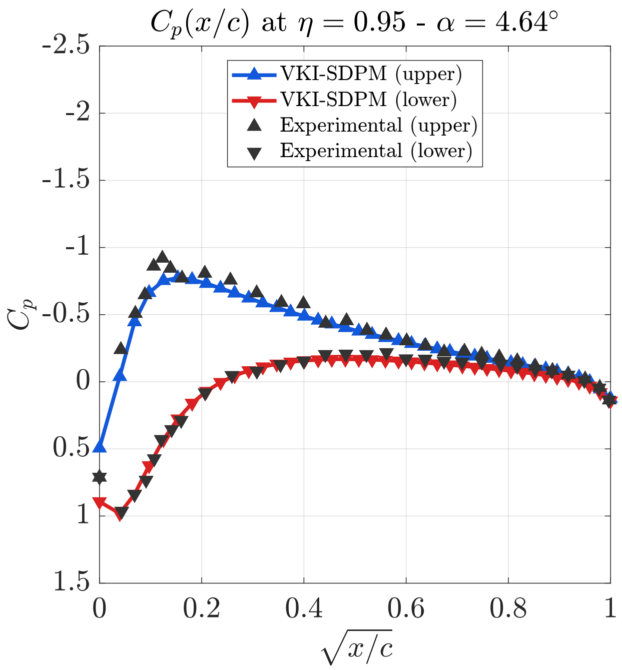
Figure D.2: Experimental and numerical comparison of C_p distributions for the Yip–Shubert test case at $\alpha = 4.64°$ ($\eta = 0.05$, 0.20, 0.34, 0.48).



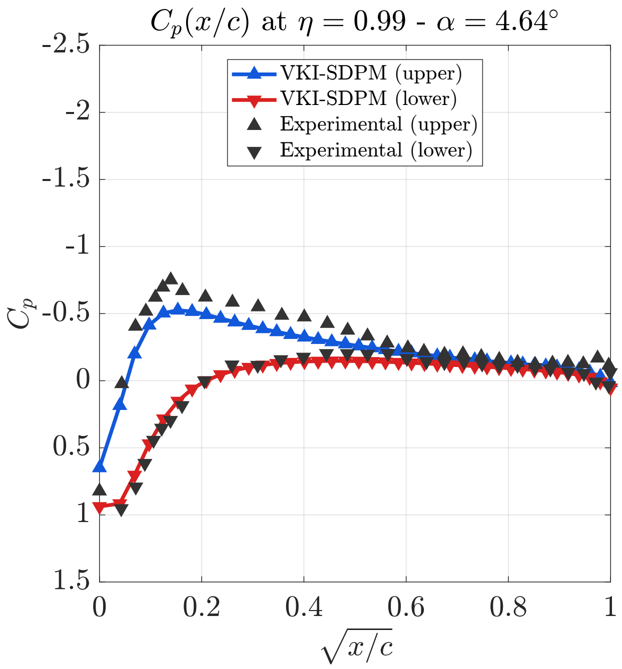
(e) $\eta = 0.61$ **(f)** $\eta = 0.72$

(g) $\eta = 0.82$ **(h)** $\eta = 0.90$

Figure D.2 (continued) ($\eta = 0.61,\ 0.72,\ 0.82,\ 0.90$).



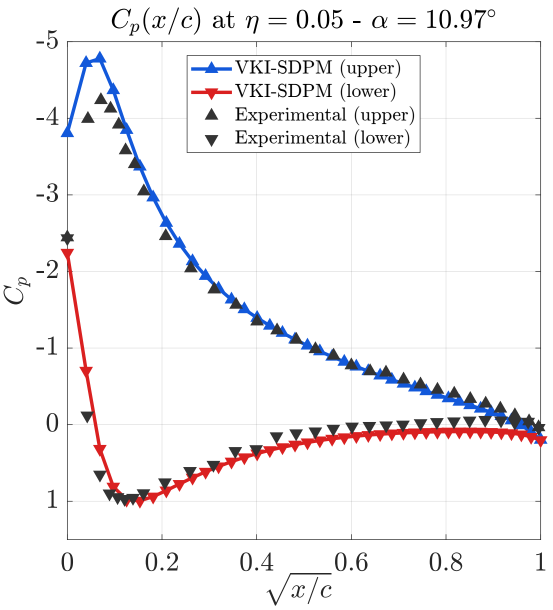
(i) $\eta = 0.95$



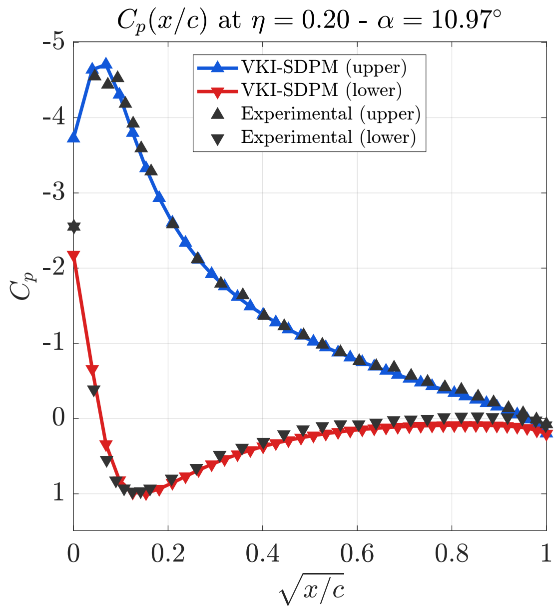
(j) $\eta = 0.99$

Figure D.2 (continued) ($\eta = 0.95$, 0.99).

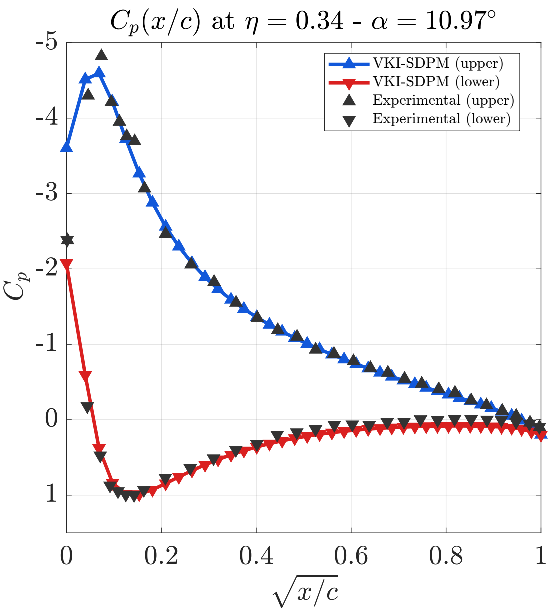
D.1.3 C_p Distributions for $\alpha = 10.97°$ and $q_\infty = 1.48$ kPa



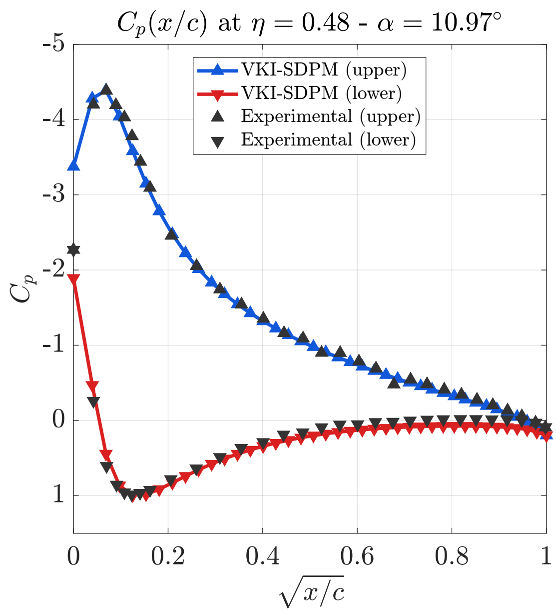
(a) $\eta = 0.05$



(b) $\eta = 0.20$

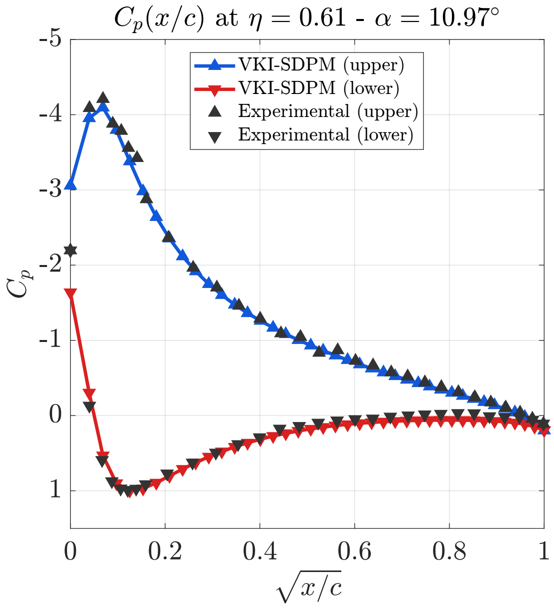


(c) $\eta = 0.34$

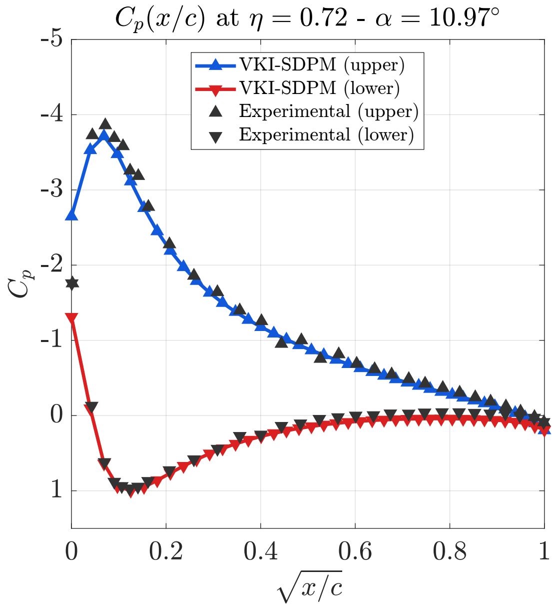


(d) $\eta = 0.48$

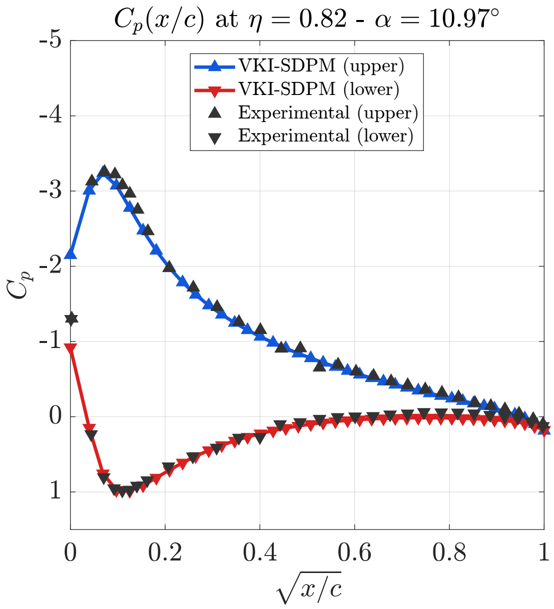
Figure D.3: $C_p(x/c)$ distributions at $\alpha = 10.97°$ ($\eta = 0.05$, 0.20, 0.34, 0.48).



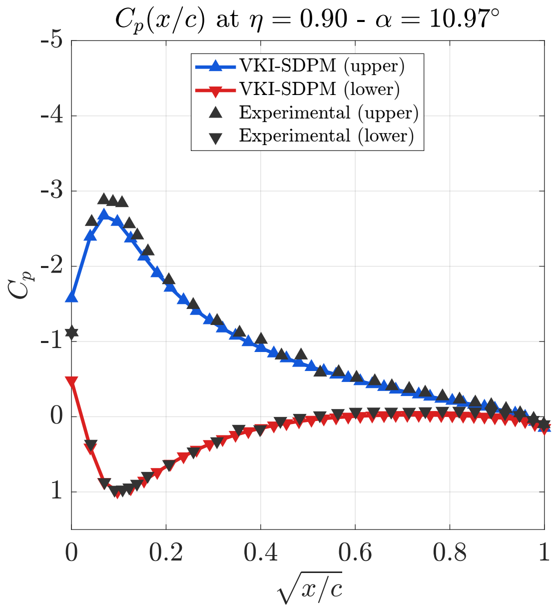
(e) $\eta = 0.61$



(f) $\eta = 0.72$

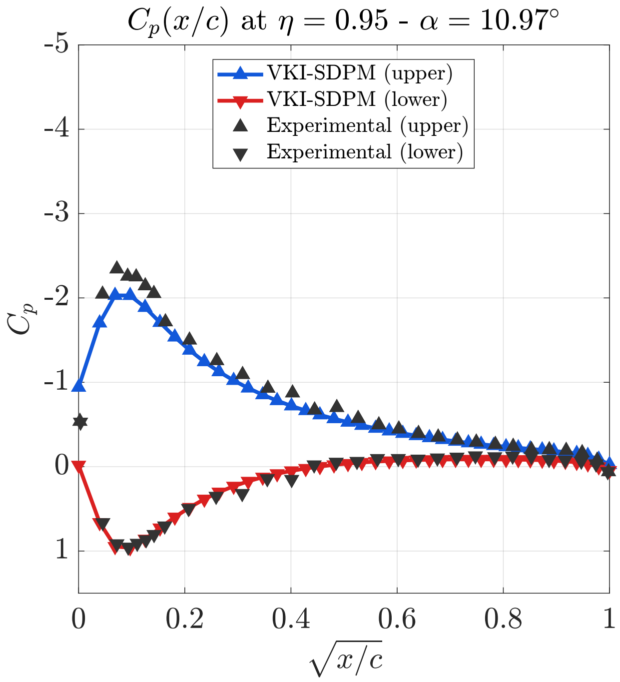


(g) $\eta = 0.82$

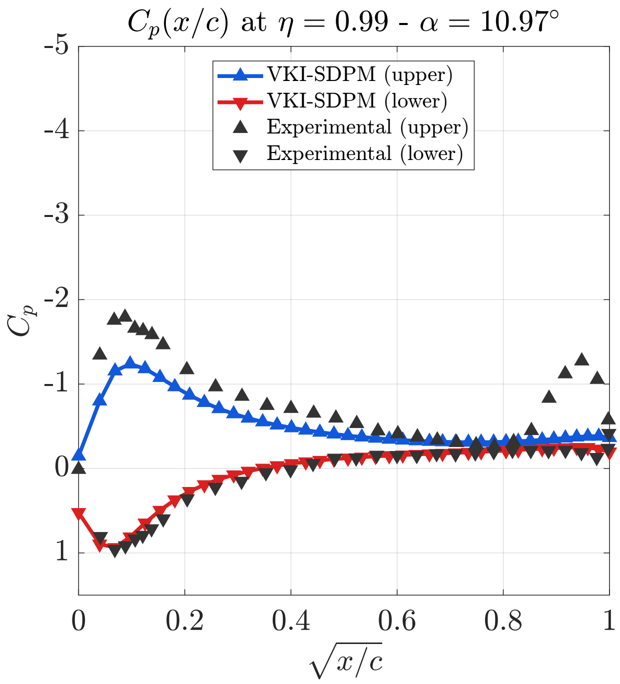


(h) $\eta = 0.90$

Figure D.3 (continued) ($\eta = 0.61$, 0.72, 0.82, 0.90).



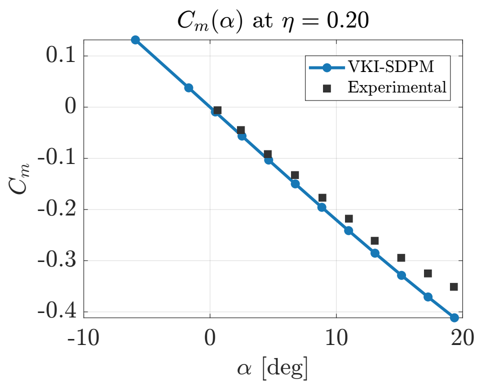
(i) $\eta = 0.95$



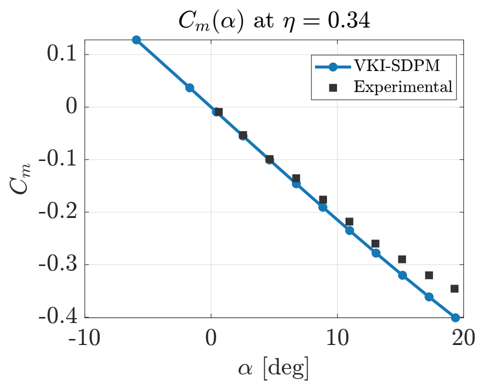
(j) $\eta = 0.99$

Figure D.2 (continued) ($\eta = 0.95$, 0.99).

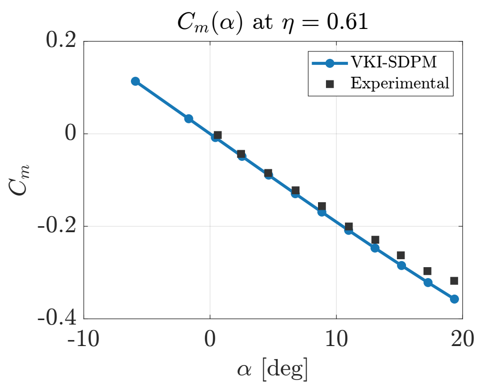
D.1.4 $C_m(\alpha)$ Spanwise Distribution



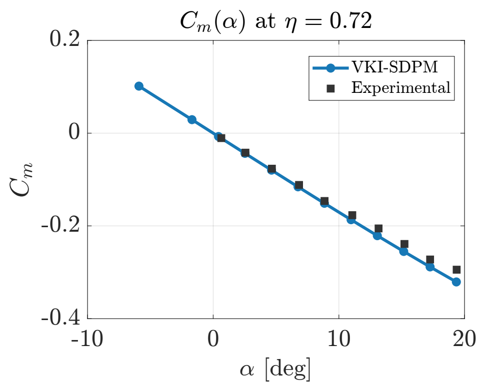
(k) $\eta = 0.20$



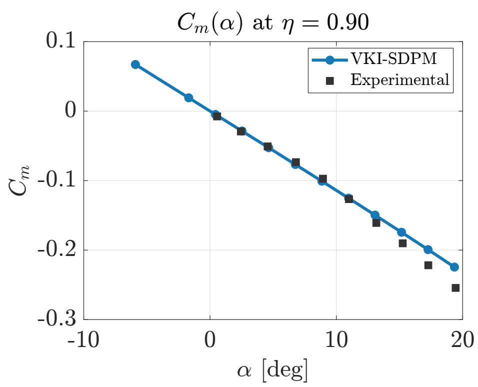
(l) $\eta = 0.34$



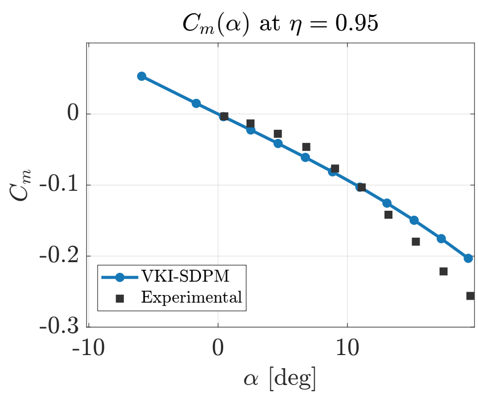
(m) $\eta = 0.61$



(n) $\eta = 0.72$



(o) $\eta = 0.90$

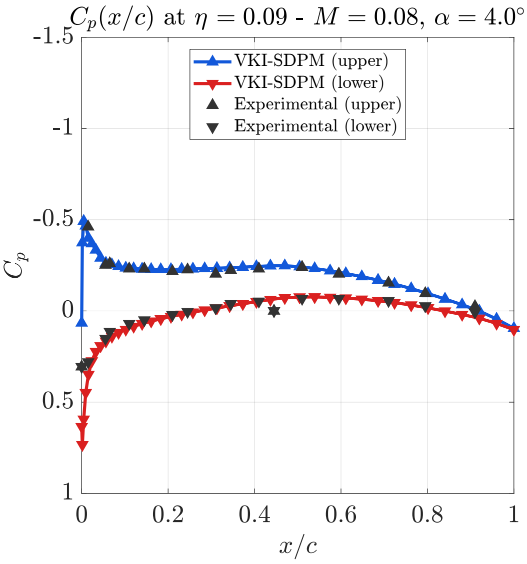


(p) $\eta = 0.95$

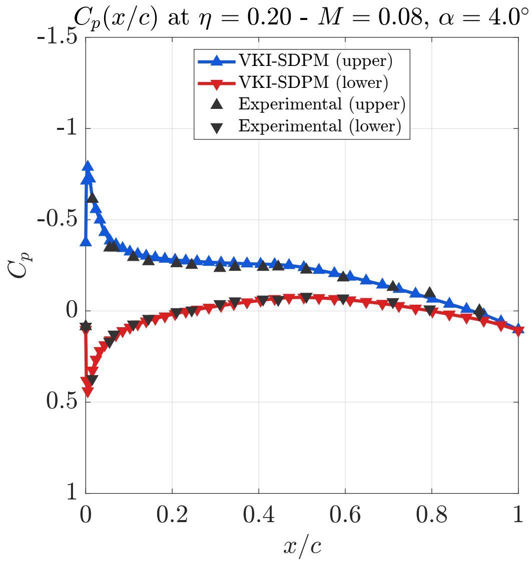
Figure D.3: $C_m(\alpha)$ curves along the spanwise sections for the Yip–Shubert test cases. ($\eta = 0.20,\ 0.34,\ 0.61,\ 0.72,\ 0.90,\ 0.95$).

D.2 Kolbe-Boltz Test Case

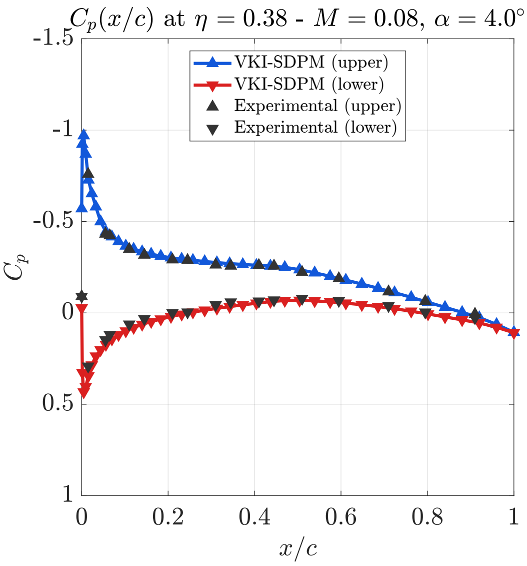
D.2.1 C_p Distribution for $M = 0.08$, $Re = 8 \cdot 10^6$ and $\alpha_u = 4°$



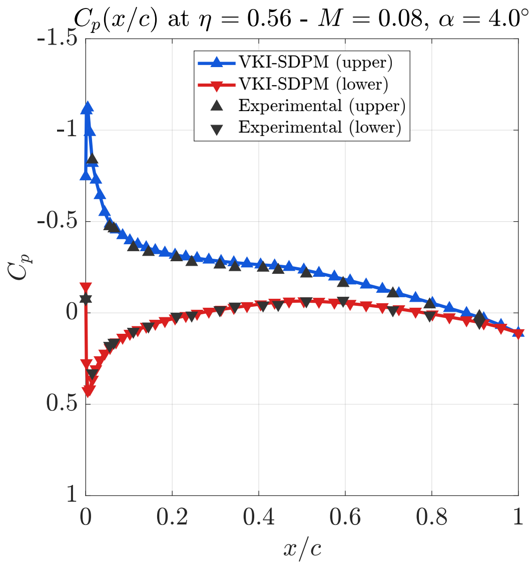
(a) $\eta = 0.09$



(b) $\eta = 0.20$

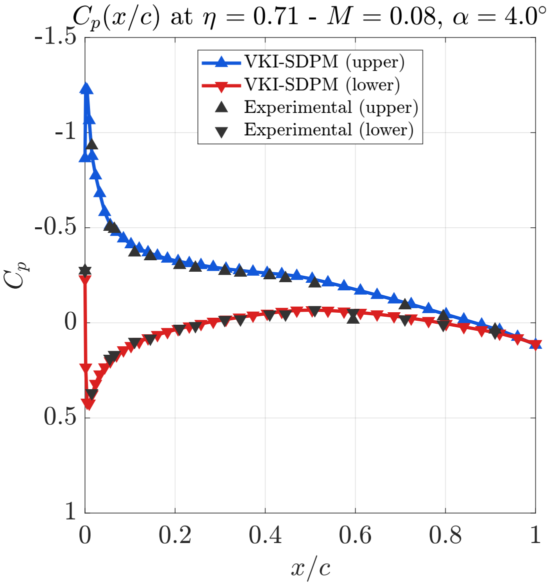


(c) $\eta = 0.38$

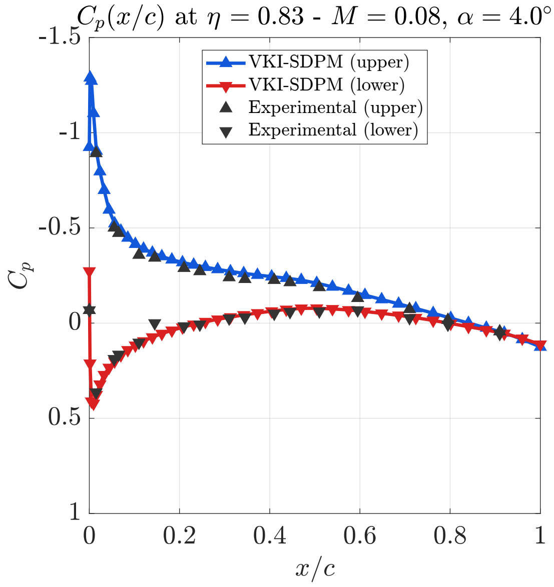


(d) $\eta = 0.56$

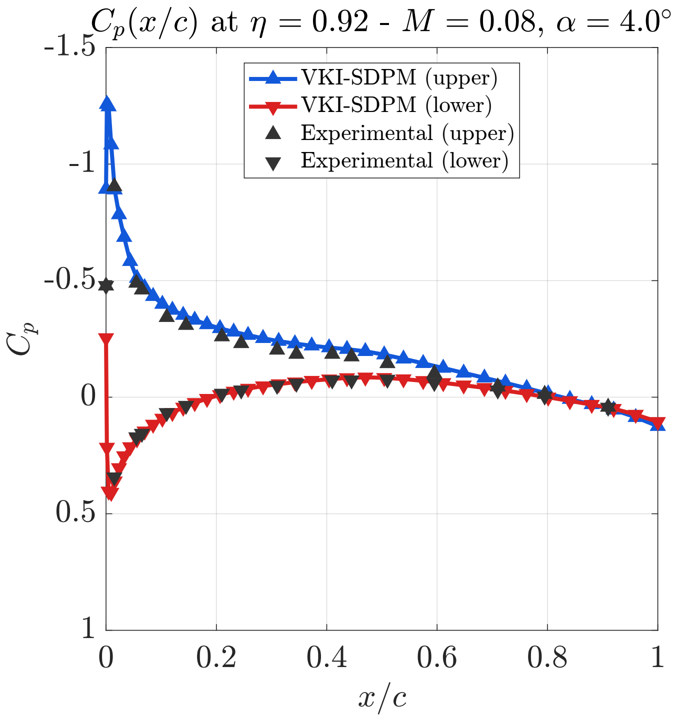
Figure D.4: Experimental and numerical comparison of C_p distributions for the Kolbe–Boltz test case at $M = 0.08$, $Re = 8 \cdot 10^6$ and $\alpha_u = 4°$. ($\eta =$ 0.09, 0.20, 0.38, 0.56).



(e) $\eta = 0.71$



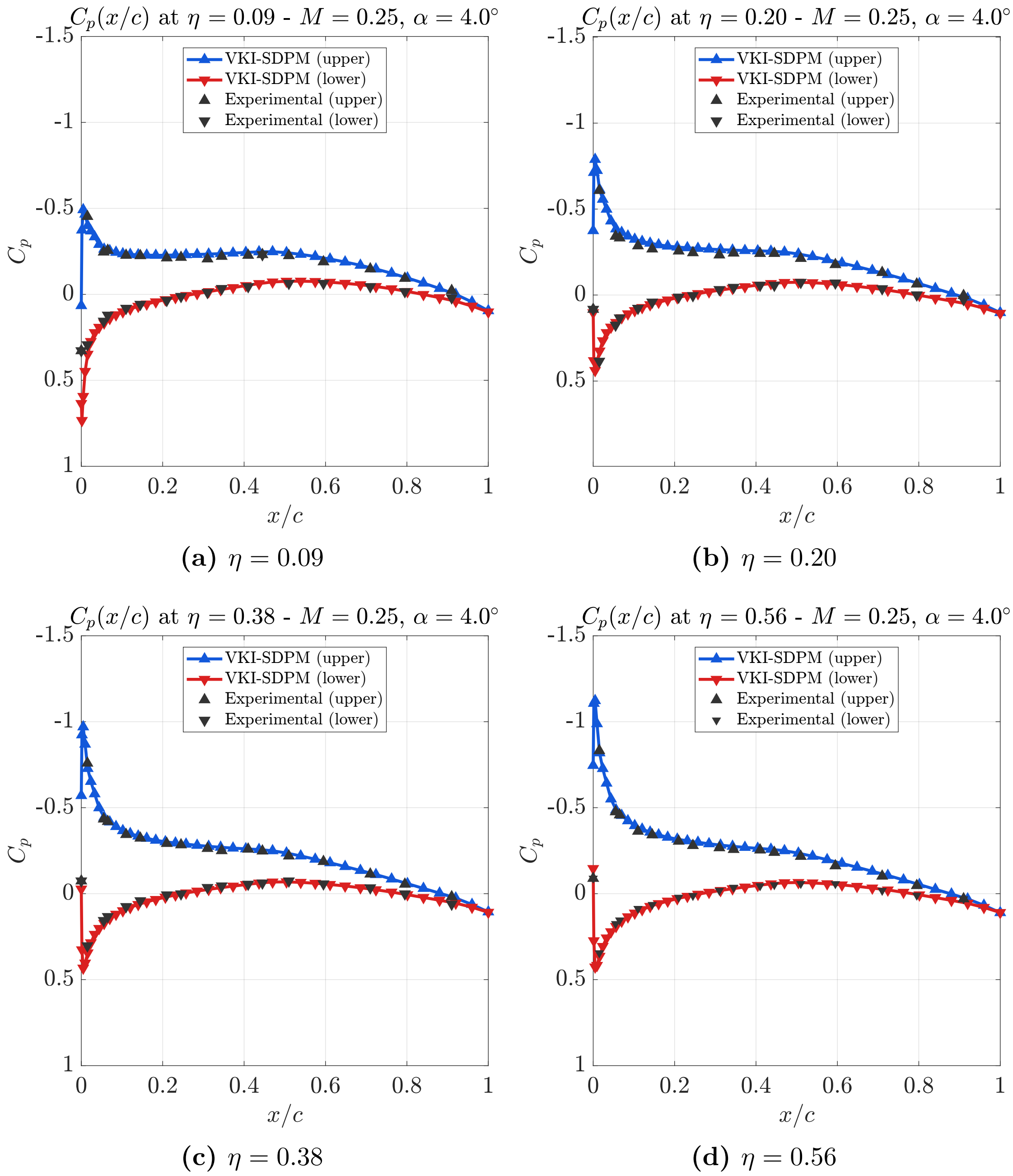
(f) $\eta = 0.83$



(g) $\eta = 0.92$

Figure D.4 (continued) ($\eta = 0.71$, 0.83, 0.92).

D.2.2 C_p Distribution for $M = 0.25$, $Re = 4 \cdot 10^6$ and $\alpha_u = 4°$



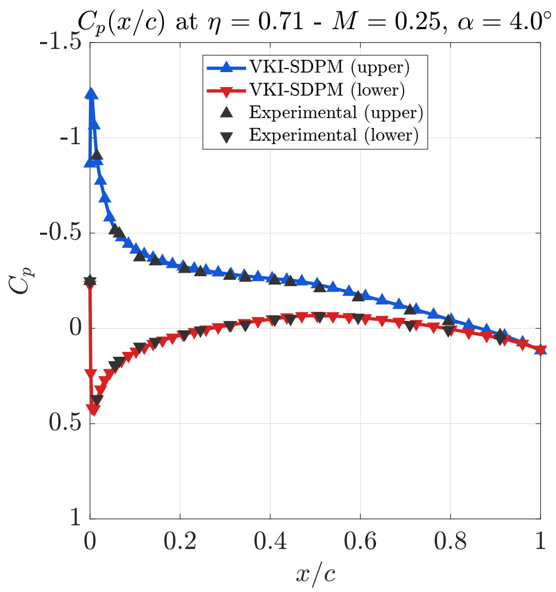
(a) $\eta = 0.09$

(b) $\eta = 0.20$

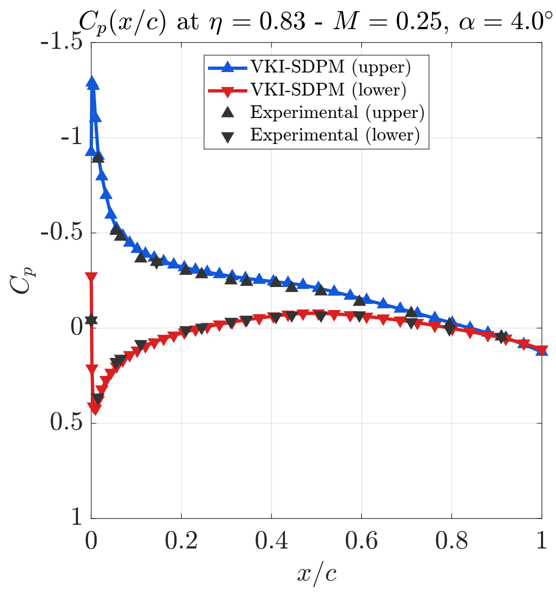
(c) $\eta = 0.38$

(d) $\eta = 0.56$

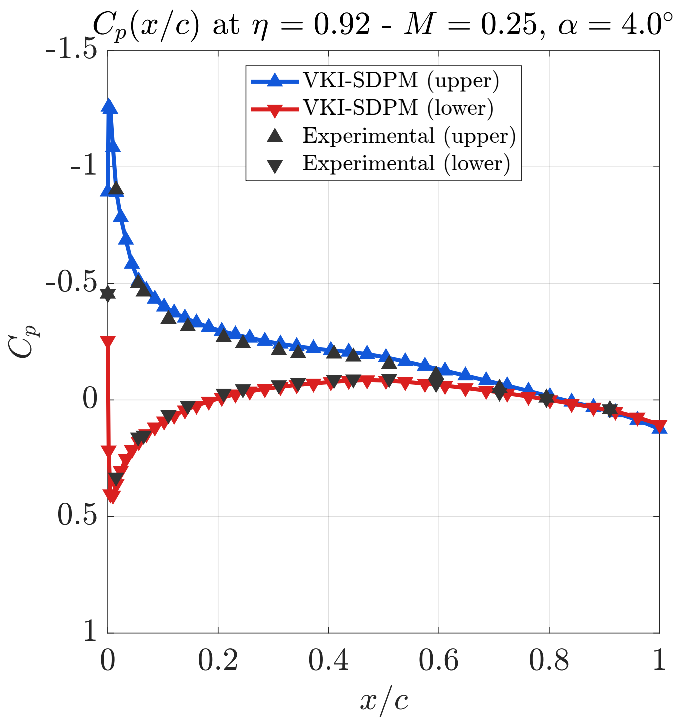
Figure D.5: C_p distributions for the Kolbe–Boltz case at $M = 0.25$, $Re = 4 \cdot 10^6$, $\alpha_u = 4°$ ($\eta = 0.09$, 0.20, 0.38, 0.56).



(e) $\eta = 0.71$



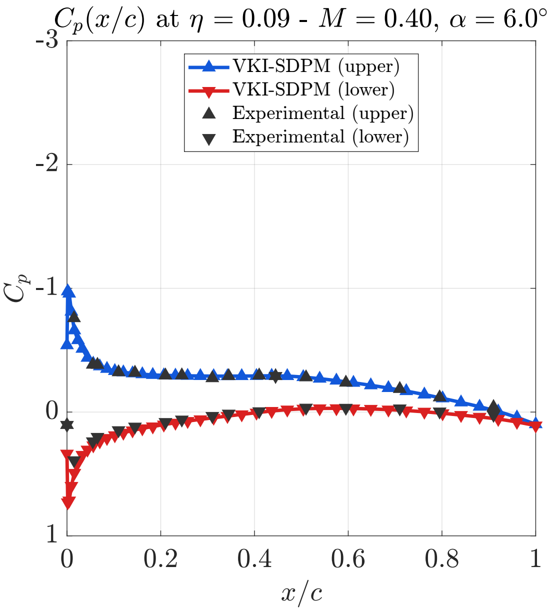
(f) $\eta = 0.83$



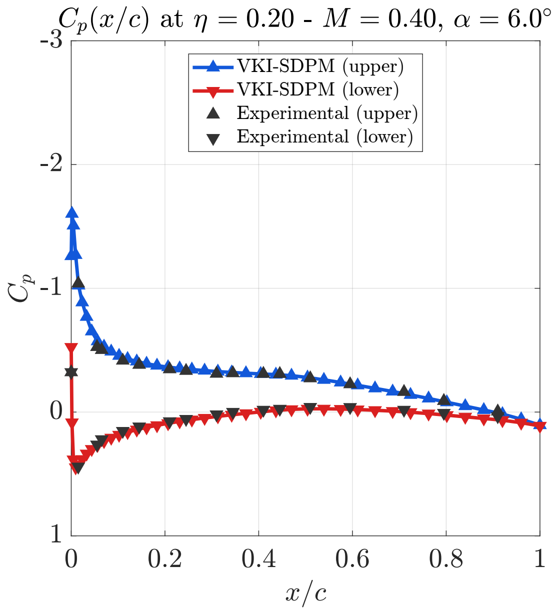
(g) $\eta = 0.92$

Figure D.5 (continued) ($\eta = 0.71$, 0.83, 0.92).

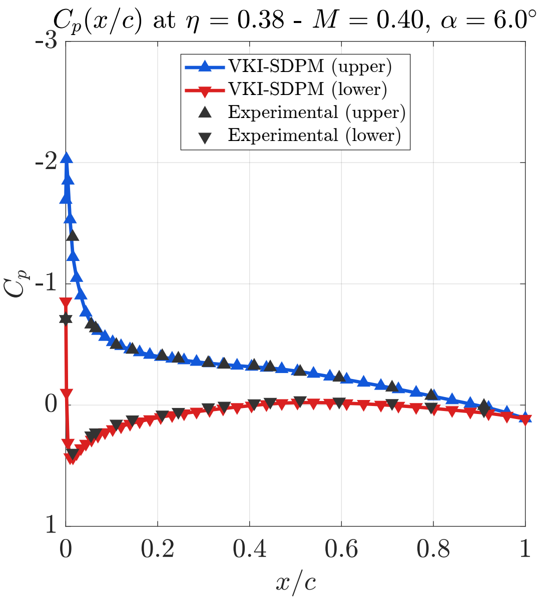
D.2.3 C_p Distribution for $M = 0.40$, $Re = 4 \cdot 10^6$ and $\alpha_u = 6°$



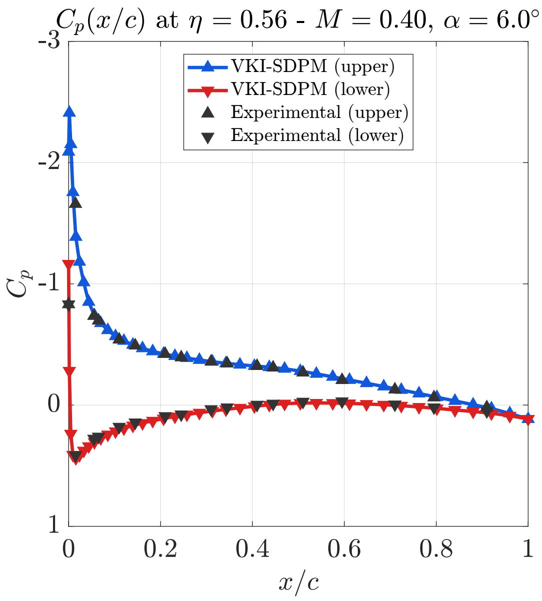
(a) $\eta = 0.09$



(b) $\eta = 0.20$

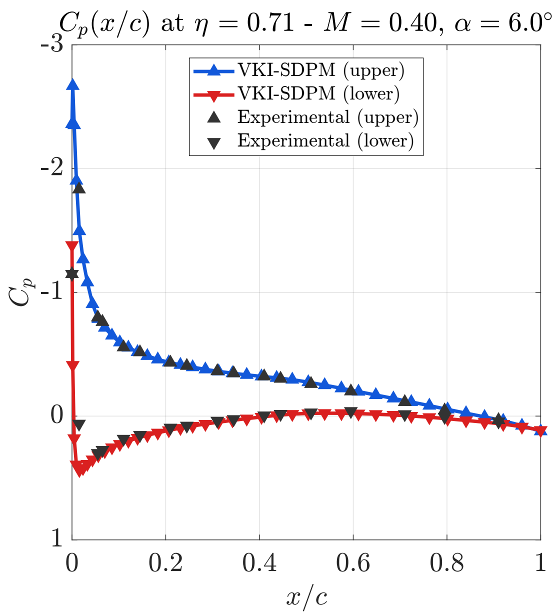


(c) $\eta = 0.38$

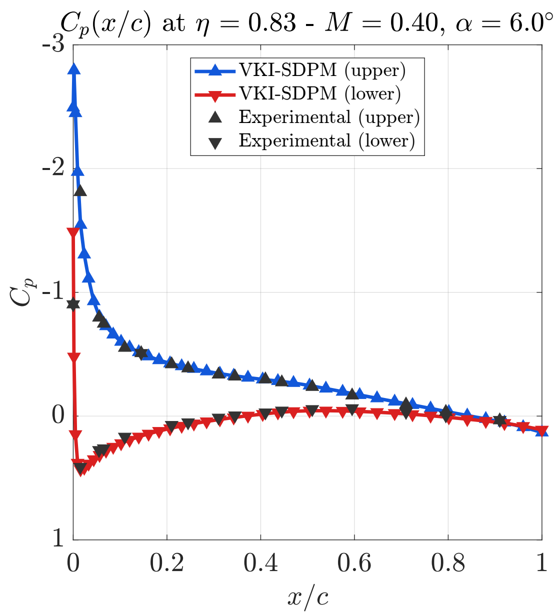


(d) $\eta = 0.56$

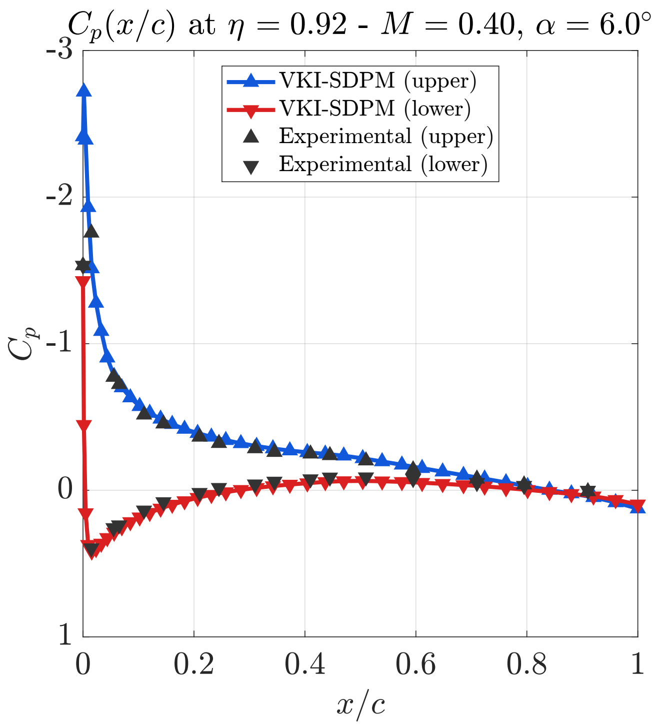
Figure D.6: C_p distributions for the Kolbe–Boltz test case at $M = 0.40$, $Re = 4 \cdot 10^6$, $\alpha_u = 6°$ ($\eta = 0.09$, 0.20, 0.38, 0.56).



(e) $\eta = 0.71$



(f) $\eta = 0.83$

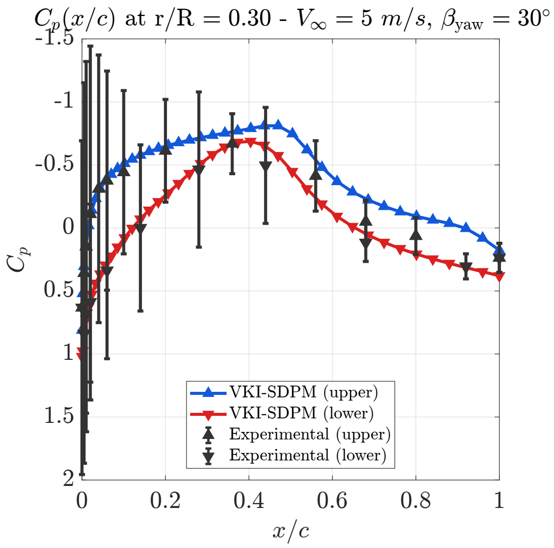


(g) $\eta = 0.92$

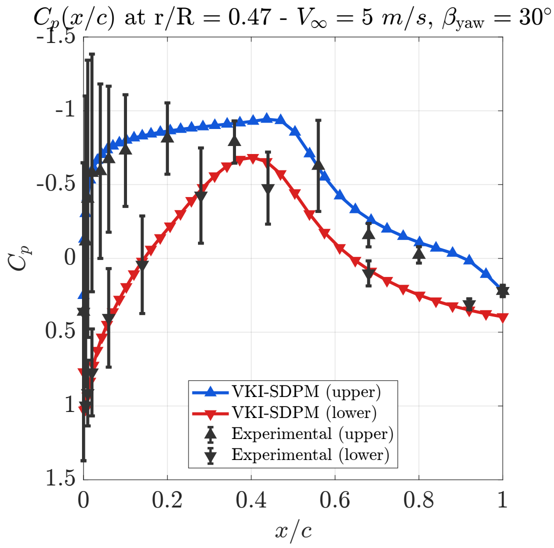
Figure D.6 (continued) ($\eta = 0.71$, 0.83, 0.92).

D.3 NREL UAE Phase VI

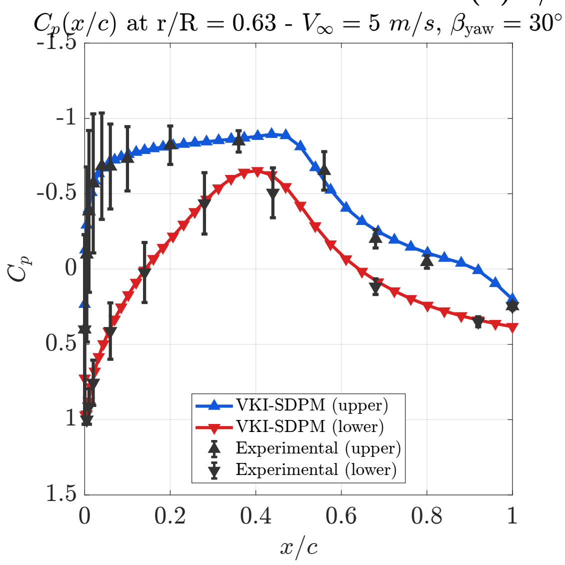
D.3.1 C_p Distribution for S0500300



(a) $r/R = 0.30$

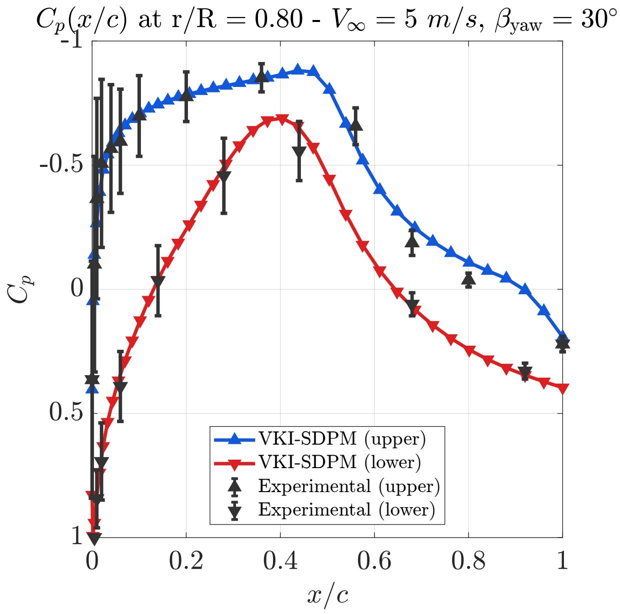


(b) $r/R = 0.47$

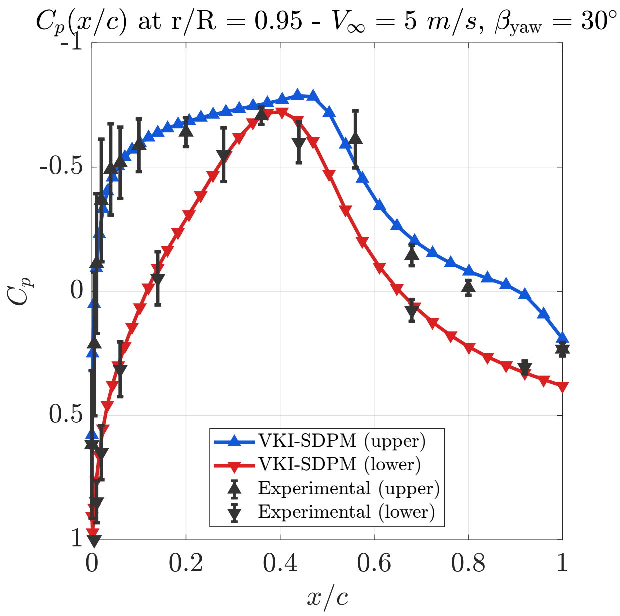


(c) $r/R = 0.63$

Figure D.7: C_p distributions for the NREL UAE Phase VI case at $V_\infty = 5$ m/s, $\beta_{\text{yaw}} = 30°$ ($r/R = 0.30$, 0.47, 0.63).



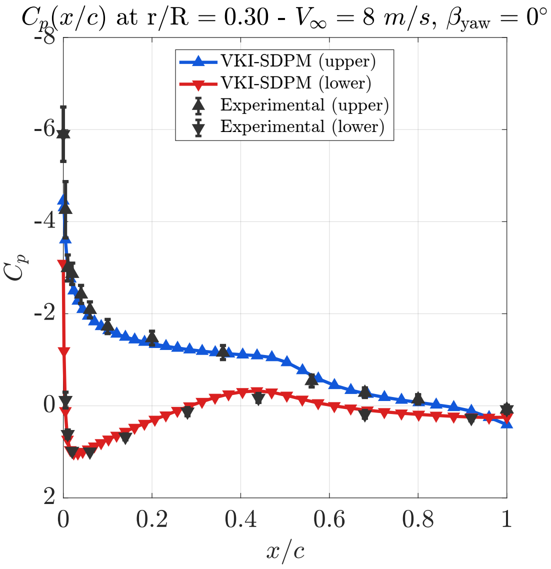
(d) $r/R = 0.80$



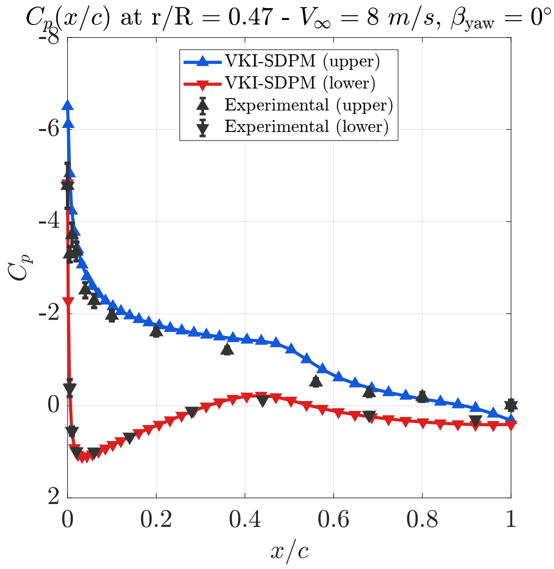
(e) $r/R = 0.95$

Figure D.7 (continued) ($r/R = 0.80$, 0.95).

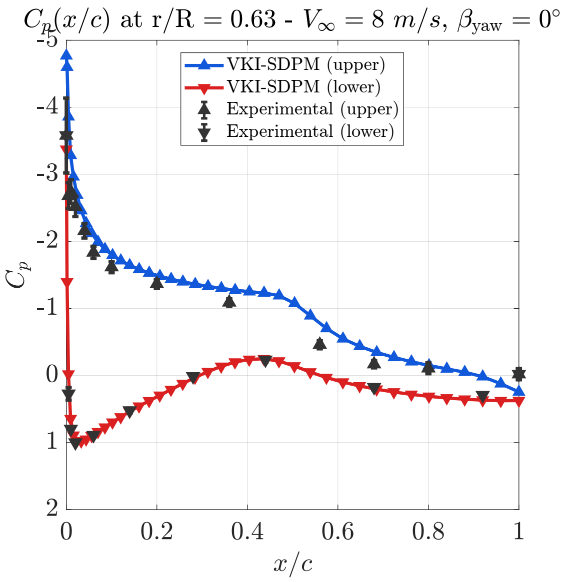
D.3.2 C_p Distribution for S0800000



(a) $r/R = 0.30$

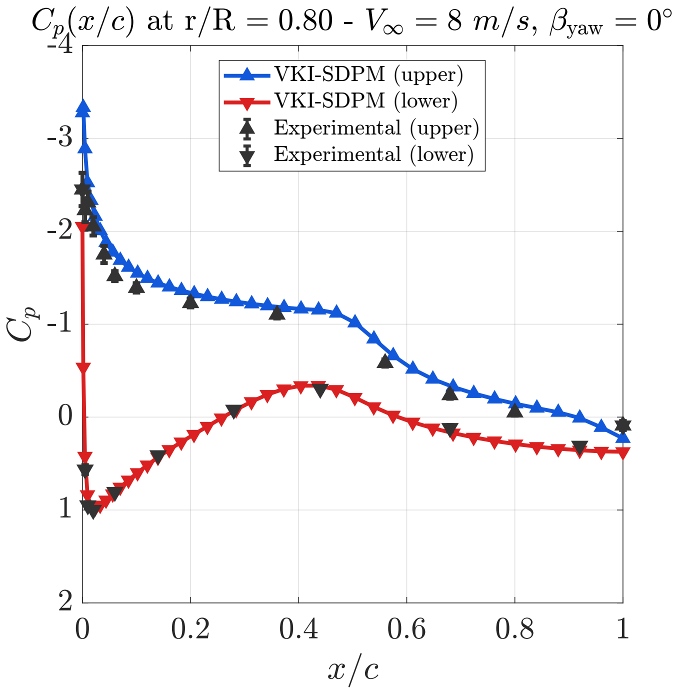


(b) $r/R = 0.47$

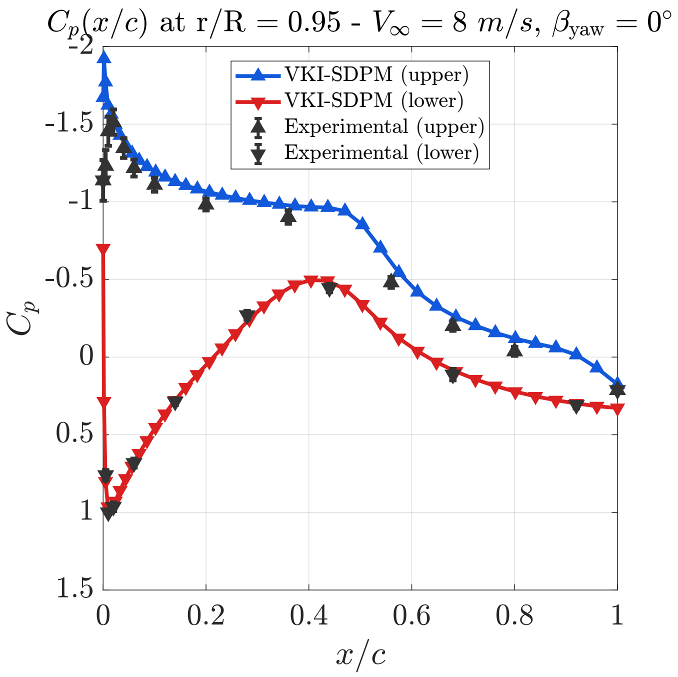


(c) $r/R = 0.63$

Figure D.8: C_p distributions for the NREL UAE Phase VI case at $V_\infty = 8$ m/s, $\beta_{\text{yaw}} = 0°$ ($r/R = 0.30$, 0.47, 0.63).



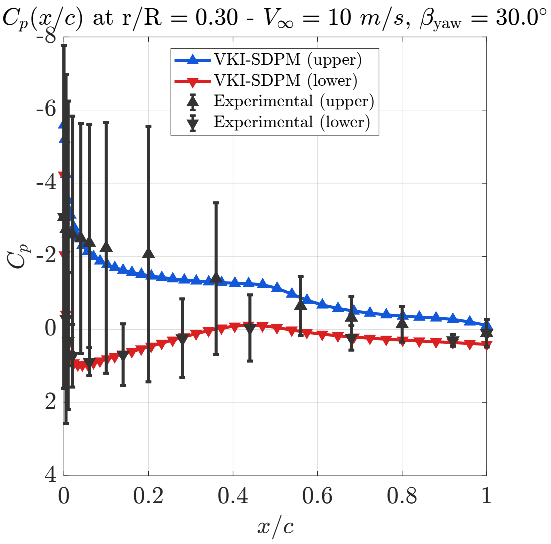
(d) $r/R = 0.80$



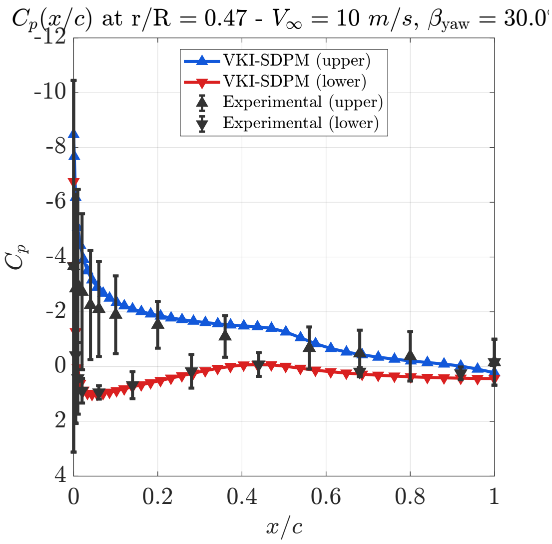
(e) $r/R = 0.95$

Figure D.8 (continued) ($r/R = 0.80$, 0.95).

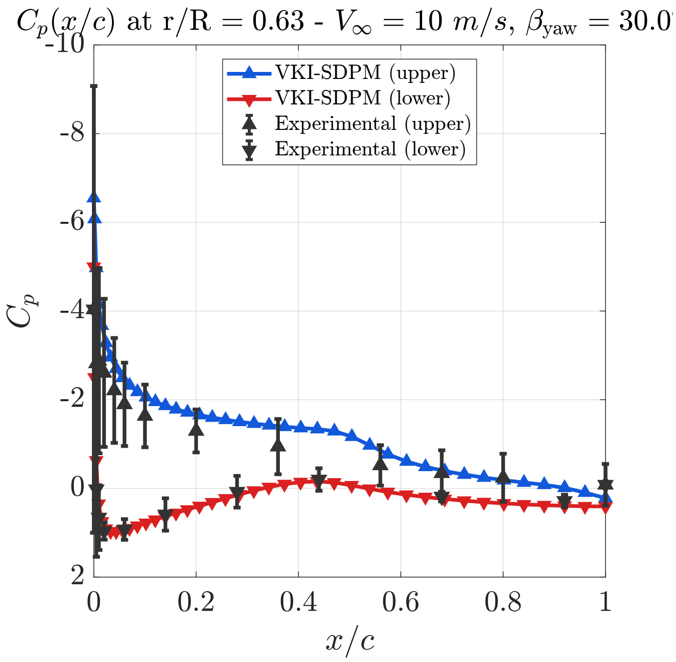
D.3.3 C_p Distribution for S1000300



(a) $r/R = 0.30$

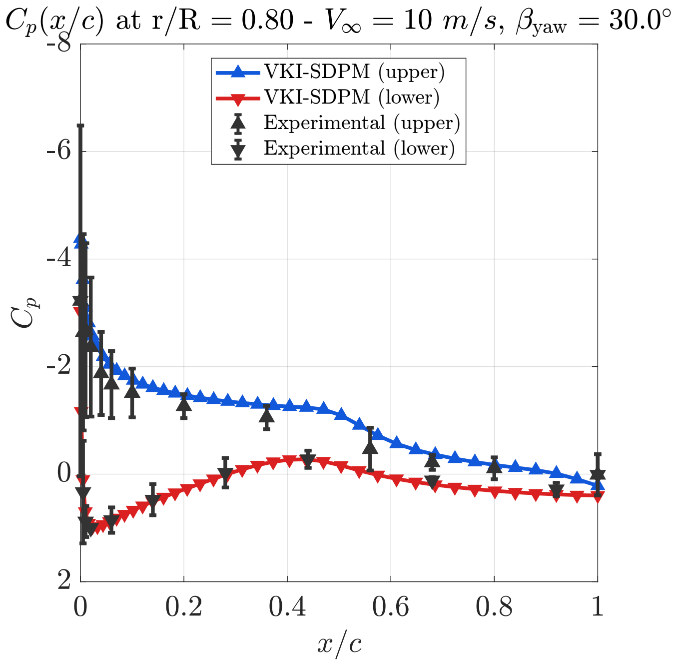


(b) $r/R = 0.47$

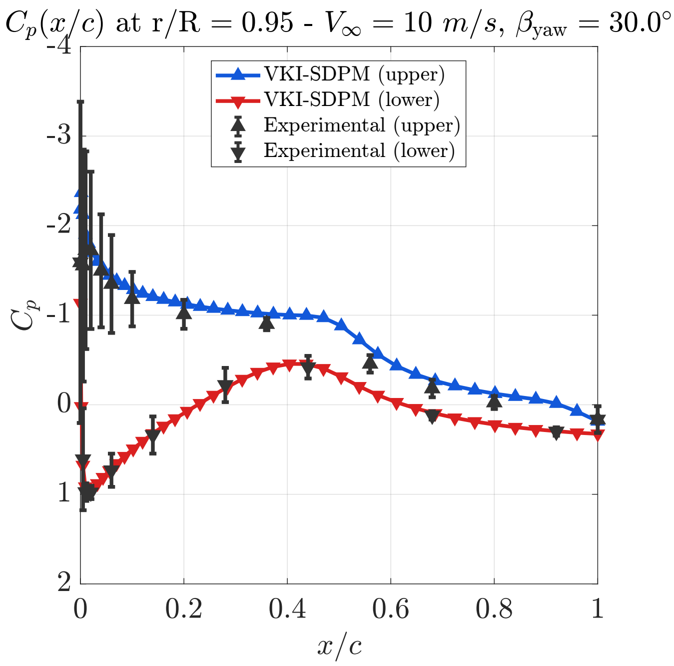


(c) $r/R = 0.63$

Figure D.9: C_p distributions for the NREL UAE Phase VI at $V_\infty = 10$ m/s, $\beta_{\text{yaw}} = 30°$ ($r/R = 0.30$, 0.47, 0.63).



(d) $r/R = 0.80$

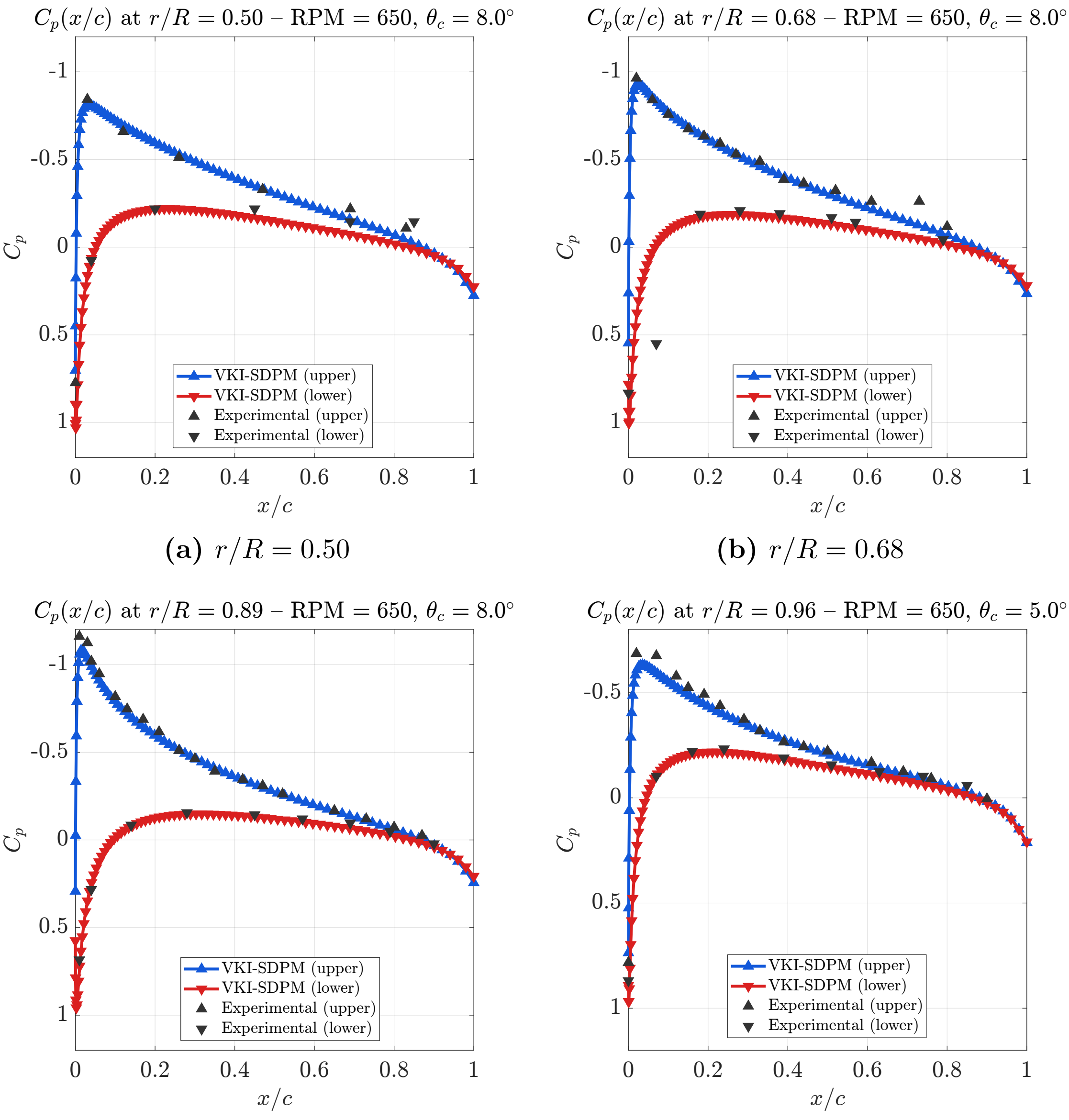


(e) $r/R = 0.95$

Figure D.9 (continued) ($r/R = 0.80$, 0.95).

D.4 Caradonna–Tung test case

D.4.1 C_p Distributions for $\Omega = 650$ rpm and $\theta_c = 8°$

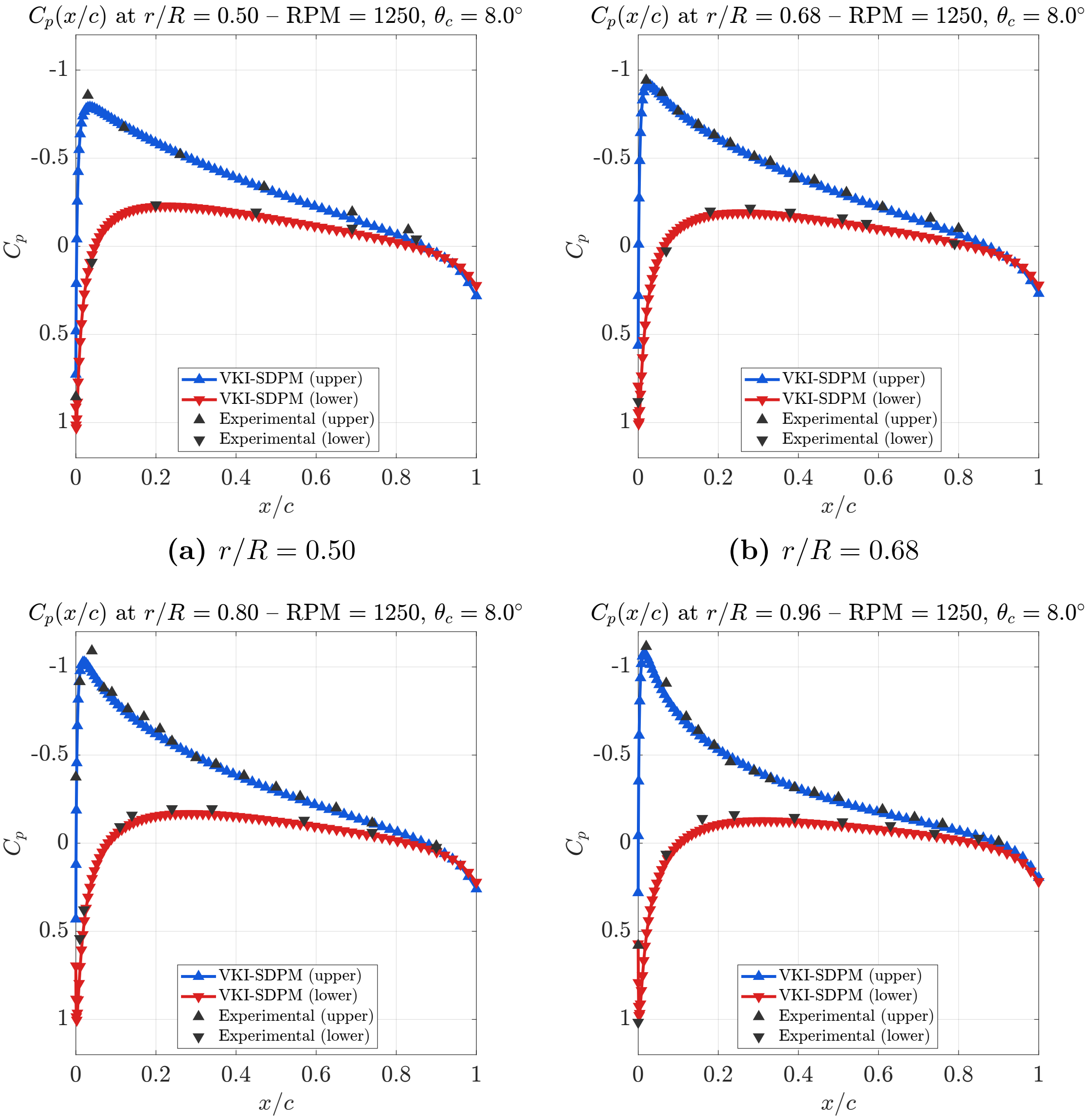


(a) $r/R = 0.50$ **(b)** $r/R = 0.68$

(c) $r/R = 0.89$ **(d)** $r/R = 0.96$

Figure D.10: C_p distributions for the Caradonna–Tung test case rotor at $\Omega = 650$ rpm and $\theta_c = 8°$.

D.4.2 C_p Distributions for $\Omega = 1250$ rpm and $\theta_c = 5°$



(a) $r/R = 0.50$ (b) $r/R = 0.68$

(c) $r/R = 0.80$ (d) $r/R = 0.96$

Figure D.11: C_p distributions for the Caradonna–Tung test case rotor at $\Omega = 1250$ rpm and $\theta_c = 5°$.

Bibliography

[1] E. J. Alvarez, J. Mehr, and A. Ning, «FLOWUnsteady: An interactional aerodynamics solver for multirotor aircraft and wind energy», in *AIAA AVIATION 2022 Forum*, Chicago, IL: American Institute of Aeronautics and Astronautics, 2022. DOI: 10.2514/6.2022-3218.

[2] A. Ning, «Using blade element momentum methods with gradient-based design optimization», *Structural and Multidisciplinary Optimization*, vol. 64, pp. 991–1014, Aug. 2021. DOI: 10.1007/s00158-021-02883-6.

[3] J. N. Sørensen, *General Momentum Theory for Horizontal Axis Wind Turbines* (Research Topics in Wind Energy). Cham, Switzerland: Springer, 2016, vol. 4. DOI: 10.1007/978-3-319-22114-4.

[4] J. J. Bertin and R. M. Cummings, *Aerodynamics for Engineers*, 5th ed. Pearson/Prentice Hall, 2009, ISBN: 9780132272681.

[5] J. Katz and A. Plotkin, *Low-Speed Aerodynamics* (Cambridge Aerospace Series), 2nd ed. Cambridge University Press, 2001, vol. 13, ISBN: 9780521665520. DOI: 10.1017/CBO9780511810329.

[6] J. G. Leishman, *Principles of Helicopter Aerodynamics* (Cambridge Aerospace Series), 2nd ed. Cambridge: Cambridge University Press, 2006.

[7] M. Cerny and C. Breitsamter, «Investigation of small-scale propellers under non-axial inflow conditions», *Aerospace Science and Technology*, vol. 106, Nov. 2020. DOI: 10.1016/j.ast.2020.106048.

[8] E. Karaca, «Accuracy and robustness of drone trajectories using BEMT models», von Karman Institute for Fluid Dynamics, Short Training Program Final Report, 2024, Department of Environmental and Applied Fluid Dynamics, Aeroacoustics and Low-Speed Aerodynamics Group. Supervisor: Prof. Dr. Christophe Schram. Internal unpublished report.

[9] L. P. Yip and G. L. Shubert, «Pressure distributions on a 1- by 3-meter semispan wing at sweep angles from 0° to 40° in subsonic flow», NASA Langley Research Center, Hampton, VA, NASA Technical Note NASA TN D-8307, Dec. 1976. [Online]. Available: `https://ntrs.nasa.gov/api/citations/19770009036/downloads/19770009036.pdf`.

[10] C. D. Kolbe and F. W. Boltz, «The forces and pressure distribution at subsonic speeds on a plane wing having 45 degrees of sweepback, an aspect ratio of 3, and a taper ratio of 0.5», National Advisory Committee for Aeronautics, NACA Research Memorandum NACA RM A51G31, Oct. 1951. [Online]. Available: `https://ntrs.nasa.gov/citations/19930086688`.

[11] M. M. Hand, D. A. Simms, L. J. Fingersh, D. W. Jager, J. R. Cotrell, S. Schreck, and S. M. Larwood, «Unsteady aerodynamics experiment phase VI: Wind tunnel test configurations and available data campaigns», National Renewable Energy Laboratory, Golden, CO, NREL Technical Report NREL/TP-500-29955, Dec. 2001. DOI: `10.2172/15000240`. [Online]. Available: `https://docs.nrel.gov/docs/fy02osti/29955.pdf`.

[12] F. X. Caradonna and C. Tung, «Experimental and analytical studies of a model helicopter rotor in hover», National Aeronautics and Space Administration, Ames Research Center, Moffett Field, CA, NASA Technical Memorandum NASA TM-81232, Sep. 1981.

[13] W. M. Kutta, «Auftriebskrafte in stromenden flussigkeiten», *Illustrierte Aeronautische Mitteilungen*, vol. 6, no. 133, pp. 133–135, 1902.

[14] W. Thomson, «On vortex motion», *Transactions of the Royal Society of Edinburgh*, vol. 25, no. 1, pp. 217–260, 1868. DOI: `10.1017/S0080456800028179`.

[15] H. Glauert, «The effect of compressibility on the lift of an aerofoil», *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, vol. 118, no. 779, pp. 113–119, 1928. DOI: `10.1098/rspa.1928.0039`.

[16] H. von Helmholtz, «On integrals of the hydrodynamical equations, which express vortex-motion», *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 4th ser., vol. 33, pp. 485–512, 1867, Translated by P. G. Tait.

[17] P. Saffman, *Vortex Dynamics*. Cambridge University Press, 1992.

[18] K. Shaler, E. Branlard, and A. Platt, «OLAF user's guide and theory manual», National Renewable Energy Laboratory, Golden, CO, Technical Report NREL/TP-5000-75959, 2020. DOI: `10.2172/1659853`.

[19] S. Gupta, «Development of a time-accurate viscous lagrangian vortex wake model for wind turbine applications», Ph.D. dissertation, University of Maryland, College Park, MD, 2006. [Online]. Available: `https://drum.lib.umd.edu/items/cfb4ac11-3c7b-4ebb-a5fa-6793210c9241`.

[20] C. W. Oseen, «Uber die Wirbelbewegung in einer reibenden Flussigkeit», *Arkiv for Matematik, Astronomi och Fysik*, vol. 7, pp. 14–26, 1912, In German.

[21] H. Lamb, *Hydrodynamics*, 6th ed. Cambridge: Cambridge University Press, 1932.

[22] W. Kaufmann, «Uber die Ausbreitung kreiszylindrischer Wirbel in zahen (viskosen) Flussigkeiten», *Ingenieur-Archiv*, vol. 31, no. 1, pp. 1–9, 1962. DOI: `10.1007/BF00538235`.

[23] M. P. Scully, «Computation of helicopter rotor wake geometry and its influence on rotor harmonic airloads», Also issued as ASRL TR 178-1, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 1975.

[24] M. P. Scully and J. P. Sullivan, «Helicopter rotor wake geometry and airloads and development of laser doppler velocimeter for use in helicopter rotor wakes», Massachusetts Institute of Technology Aerophysics Laboratory, Cambridge, MA, Technical Report 183, MIT DSR No. 73032, Aug. 1972.

[25] G. H. Vatistas, V. Kozel, and W. C. Mih, «A simpler model for concentrated vortices», *Experiments in Fluids*, vol. 11, no. 1, pp. 73–76, 1991. DOI: `10.1007/BF00198434`.

[26] L. A. Young, «Vortex core size in the rotor near-wake», NASA Ames Research Center, Moffett Field, CA, NASA Technical Memorandum NASA/TM-2003-212275, Jun. 2003. [Online]. Available: `https://ntrs.nasa.gov/api/citations/20040082235/downloads/20040082235.pdf`.

[27] H. B. Squire, «The growth of a vortex in turbulent flow», *Aeronautical Quarterly*, vol. 16, no. 3, pp. 302–306, Aug. 1965. DOI: `10.1017/S0001925900003516`.

[28] M. Ramasamy and J. G. Leishman, «The interdependence of straining and viscous diffusion effects on vorticity in rotor flow fields», in *Proceedings of the 59th Annual Forum of the American Helicopter Society International*, vol. 59, Phoenix, AZ: American Helicopter Society International, May 2003, pp. 1787–1801.

[29] S. Ananthan and J. G. Leishman, «Role of filament strain in the free-vortex modeling of rotor wakes», *Journal of the American Helicopter Society*, vol. 49, no. 2, pp. 176–191, Apr. 2004. DOI: `10.4050/JAHS.49.176`.

[30] L. Prandtl, «Tragflugeltheorie. i. mitteilung», *Nachrichten von der Gesellschaft der Wissenschaften zu Gottingen, Mathematisch-Physikalische Klasse*, vol. 1918, pp. 451–477, 1918. [Online]. Available: http://eudml.org/doc/59036.

[31] L. Prandtl, «Tragflugeltheorie. ii. mitteilung», *Nachrichten von der Gesellschaft der Wissenschaften zu Gottingen, Mathematisch-Physikalische Klasse*, vol. 1919, pp. 107–137, 1919. [Online]. Available: http://eudml.org/doc/59045.

[32] Z. S. Montgomery, «A propeller model based on a modern numerical lifting-line algorithm with an iterative semi-free wake solver», Master's thesis, Utah State University, Logan, UT, May 2018. [Online]. Available: https://digitalcommons.usu.edu/etd/7001.

[33] S. Drzewiecki, «Méthode pour la détermination des éléments mécaniques des propulseurs hélicoı¨daux», *Bulletin de l'Association technique maritime*, no. 3, 1892.

[34] S. Drzewiecki, «Du choix des éléments déterminant les hélices propulsives, permettant leur facile comparaison entre elles», *Bulletin de l'Association technique maritime*, no. 12, pp. 71–92, 1901, Session de 1901. [Online]. Available: https://archive.org/details/bulletindelasso02marigoog.

[35] I. H. Abbott and A. E. von Doenhoff, *Theory of Wing Sections: Including a Summary of Airfoil Data*. New York, NY, USA: Dover Publications, 1959, ISBN: 9780486605869.

[36] M. Drela, «XFOIL: An analysis and design system for low reynolds number airfoils», in *Low Reynolds Number Aerodynamics*, T. J. Mueller, Ed., ser. Lecture Notes in Engineering, vol. 54, Berlin, Heidelberg: Springer, 1989, pp. 1–12, ISBN: 9783540518846. DOI: 10.1007/978-3-642-84010-4_1.

[37] N. E. Joukowski, «On annexed vortices», *Proceedings of the Physical Section of the Natural Science Society*, vol. 13, no. 2, pp. 12–25, 1906.

[38] W. M. Kutta, «Uber ebene Zirkulationsstromungen nebst flugtechnischen Anwendungen», *Sitzungsberichte der Koniglich Bayerischen Akademie der Wissenschaften, Mathematisch-Physikalische Klasse*, vol. 41, pp. 65–125, 1911.

[39] J. L. Hess and A. M. O. Smith, «Calculation of potential flow about arbitrary bodies», *Progress in Aerospace Sciences*, vol. 8, pp. 1–138, 1967. DOI: 10.1016/0376-0421(67)90003-6.

[40] G. Dimitriadis, *SDPMflut user's guide*, Version 0.6, University of Liége, Jan. 2025. [Online]. Available: https://aeroconsult.gr/SDPMflut/SDPMflut_User_guide_v_0_6.pdf.

[41] A. van Garrel, «Development of a wind turbine aerodynamics simulation module», Energy Research Centre of the Netherlands, Petten, The Netherlands, ECN Technical Report ECN-C–03-079, Aug. 2003. [Online]. Available: https://publications.ecn.nl/E/2003/ECN-C--03-079.

[42] O. Gur and A. Rosen, «Comparison between blade-element models of propellers», *The Aeronautical Journal*, vol. 112, no. 1138, pp. 689–704, Dec. 2008. DOI: 10.1017/S0001924000002669.

[43] D. Marten, J. Wendler, G. Pechlivanoglou, C. N. Nayeri, and C. O. Paschereit, «QBlade: An open source tool for design and simulation of horizontal and vertical axis wind turbines», *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. Special Issue 3, pp. 264–269, 2013.

[44] B. W. McCormick, *Aerodynamics of V/STOL Flight*. New York: Academic Press, 1967, ISBN: 0124823505.

[45] A. J. Landgrebe, «An analytical and experimental investigation of helicopter rotor hover performance and wake geometry characteristics», U.S. Army Air Mobility Research and Development Laboratory, USAAMRDL Technical Report USAAMRDL TR 71-24, Jun. 1971.

[46] A. J. Landgrebe, «The wake geometry of a hovering helicopter rotor and its influence on rotor performance», *Journal of the American Helicopter Society*, vol. 17, no. 4, pp. 3–15, Oct. 1972. DOI: 10.4050/JAHS.17.4.3.

[47] F. Mahmuddin, «The effect of flat plate theory assumption in post-stall lift and drag coefficients extrapolation with viterna method», *Journal of Subsea and Offshore -Science and Engineering-*, vol. 6, pp. 9–13, Jun. 2016. [Online]. Available: https://isomase.org/JSOse/Vol.6%20Jun%202016/6-2.pdf.

[48] M. L. Verstraete, L. R. Ceballos, C. Hente, B. A. Roccia, and C. G. Gebhardt, «Code-to-code benchmark for simulation tools based on the unsteady vortex-lattice method», *Journal of Aerospace Information Systems*, vol. 20, no. 11, pp. 719–746, 2023. DOI: 10.2514/1.I011184.

[49] P. Doerffer and O. Szulc, «Numerical simulation of model helicopter rotor in hover», *TASK Quarterly*, vol. 12, no. 3–4, pp. 227–236, 2008. [Online]. Available: https://journal.mostwiedzy.pl/TASKQuarterly/article/view/2037.

[50] S. Lee, H. Choi, L. Cho, and J. Cho, «Aerodynamic analysis of the helicopter rotor using the time-domain panel method», in *Proceedings of the 27th International Congress of the Aeronautical Sciences*, International Council of the Aeronautical Sciences, Nice, France, 2010. [Online]. Available: https://www.icas.org/icas_archive/ICAS2010/PAPERS/115.PDF.

[51] J. D. Kocurek and J. L. Tangler, «A prescribed wake lifting surface hover performance analysis», in *Proceedings of the 32nd Annual National Forum of the American Helicopter Society*, Preprint 1001, American Helicopter Society, May 1976.

[52] J. M. Summa and D. R. Clark, «A lifting-surface method for hover/climb loads», in *Proceedings of the 35th Annual Forum of the American Helicopter Society*, Preprint 79-1, American Helicopter Society, Washington, DC, May 1979.

[53] M. Sanchez Martinez and G. Dimitriadis, «Subsonic source and doublet panel methods», *Journal of Fluids and Structures*, vol. 113, p. 103 624, 2022. DOI: `10.1016/j.jfluidstructs.2022.103624`.

[54] A. Grama, A. Gupta, G. Karypis, and V. Kumar, *Introduction to Parallel Computing*, 2nd ed. Addison-Wesley, 2003.

[55] G. M. Amdahl, «Validity of the single processor approach to achieving large scale computing capabilities», in *Proceedings of the AFIPS Spring Joint Computer Conference*, AFIPS Press, 1967, pp. 483–485.

[56] E. C. Polhamus, «Jet-boundary-induced upwash velocities for swept reflection-plane models mounted vertically in 7- by 10-foot, closed, rectangular wind tunnels», National Advisory Committee for Aeronautics, NACA Technical Note 1752, 1948.

[57] J. G. Herriot, «Blockage corrections for three-dimensional-flow closed-throat wind tunnels, with consideration of the effect of compressibility», National Advisory Committee for Aeronautics, NACA Report 995, 1950, Supersedes NACA RM A7B28.

[58] J. L. Hess, «Calculation of potential flow about arbitrary three-dimensional lifting bodies», Naval Air Systems Command, Report MDC 55679-01, Oct. 1972, Contract N00019-71-C-0524.

[59] J. Roskam, *Airplane Design, Part I: Preliminary Sizing of Airplanes.* Lawrence, KS: DARcorporation, 1997, ISBN: 9781884885426.

[60] J. C. Sivells and O. J. Deters, «Jet-boundary and plan-form corrections for partial-span models with reflection-plane, end-plate, or no end-plate in a closed circular wind tunnel», National Advisory Committee for Aeronautics, NACA Report 843, 1946. [Online]. Available: `https://ntrs.nasa.gov/citations/19930091918`.

[61] J. DeYoung and C. W. Harper, «Theoretical symmetric span loading at subsonic speeds for wings having arbitrary plan form», National Advisory Committee for Aeronautics, NACA Report 921, 1948. [Online]. Available: https://ntrs.nasa.gov/api/citations/19930091985/downloads/19930091985.pdf.

[62] D. M. Somers, «Design and experimental results for the S809 airfoil», National Renewable Energy Laboratory, Golden, CO, NREL Subcontract Report NREL/SR-440-6918, 1997. DOI: 10.2172/437668. [Online]. Available: https://docs.nrel.gov/docs/legosti/old/6918.pdf.

[63] H. Glauert, *The Elements of Aerofoil and Airscrew Theory*, 2nd ed. Cambridge: Cambridge University Press, 1947.

[64] D. E. Shipley, M. S. Miller, M. C. Robinson, M. W. Luttges, and D. A. Simms, «Techniques for the determination of local dynamic pressure and angle of attack on a horizontal axis wind turbine», National Renewable Energy Laboratory, Golden, CO, NREL Technical Report NREL/TP-442-7393, May 1995. DOI: 10.2172/61151. [Online]. Available: https://www.osti.gov/biblio/61151.

[65] Vertical Aerospace. «Meet the VX4», Accessed: Sep. 22, 2025. [Online]. Available: https://vertical-aerospace.com/meet-the-vx4/.

[66] C. Alcock. «Vertical raises $60 million for VX4 eVTOL aircraft development», Accessed: Sep. 22, 2025. [Online]. Available: https://www.ainonline.com/aviation-news/futureflight/2025-07-15/vertical-raises-60-million-vx4-evtol-aircraft-development.

[67] D. van Wijk, «A low-order approach to predict the tip-on-tip interaction noise in distributed electric-propulsion systems in vtol vehicles», Faculty of Engineering Technology, Department of Engineering Fluid Dynamics (484), in collaboration with the von Karman Institute for Fluid Dynamics, Master's thesis, University of Twente, Enschede, The Netherlands, 2025.

[68] L. A. Viterna and D. C. Janetzke, «Theoretical and experimental power from large horizontal-axis wind turbines», National Aeronautics and Space Administration, Lewis Research Center, Cleveland, OH, NASA Technical Memorandum DOE/NASA/20320-41; NASA-TM-82944, Sep. 1982. DOI: 10.2172/6763041. [Online]. Available: https://www.osti.gov/biblio/6763041.

[69] B. Montgomerie, «Methods for root effects, tip effects and extending the angle of attack range to ±180°, with application to aerodynamics for blades on wind turbines and propellers», FOI Swedish Defence Research Agency, Stockholm, Sweden, Scientific Report FOI-R–1305–SE, Jun. 2004. [Online]. Available: https://www.foi.se/rest-api/report/FOI-R--1305--SE.

[70] J. Tangler and D. Kocurek, «Wind turbine post-stall airfoil performance characteristics guidelines for blade-element momentum methods», in *43rd AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2005-591; NREL/CP-500-36900, Reno, NV, USA, Jan. 2005, pp. 146–155. DOI: 10.2514/6.2005-591.

[71] M. Oppizzi, «Sviluppo di un codice a vortici discreti per il calcolo della scia del rotore dell'elicottero», Academic year 2016–2017, Master's thesis, Politecnico di Milano, Milan, Italy, 2017.

[72] L. L. Erickson, «Panel methods: An introduction», NASA, Moffett Field, CA, USA, NASA Technical Paper 2995, 1990.

[73] M. Drela and M. B. Giles, «Viscous-inviscid analysis of transonic and low reynolds number airfoils», *AIAA Journal*, vol. 25, no. 10, pp. 1347–1355, 1987. DOI: 10.2514/3.9789.

[74] P. Kunze, «Approximate boundary layer methods for a 3d panel code for helicopter simulations», in *45th European Rotorcraft Forum*, Warsaw, Poland, 2019.

[75] W. J. M. Rankine, «On the mechanical principles of the action of propellers», *Transactions of the Institution of Naval Architects*, vol. 6, pp. 21–34, 1865.

[76] W. Froude, «On the elementary relation between pitch, slip, and propulsive efficiency», *Transactions of the Institution of Naval Architects*, vol. 19, pp. 47–65, 1878.

[77] R. E. Froude, «On the part played in propulsion by differences of fluid pressure», *Transactions of the Institution of Naval Architects*, vol. 30, pp. 390–405, 1889.

[78] H. Glauert, «Airplane propellers», in *Aerodynamic Theory: A General Review of Progress Under a Grant of the Guggenheim Fund for the Promotion of Aeronautics*, ser. Division L, W. F. Durand, Ed., vol. IV, Berlin, Germany: Julius Springer, 1935, pp. 169–360. DOI: 10.1007/978-3-642-91487-4_3.

[79] M. K. Rwigema, «Propeller blade element momentum theory with vortex wake deflection», in *Proceedings of the 27th Congress of the International Council of the Aeronautical Sciences (ICAS)*, Nice, France, Sep. 2010, pp. 727–735. [Online]. Available: https://www.icas.org/icas_archive/ICAS2010/PAPERS/434.PDF.

[80] J. M. Jonkman, «Modeling of the uae wind turbine for refinement of fast_ad», National Renewable Energy Laboratory (NREL), Golden, CO, USA, Tech. Rep. NREL/TP-500-34755, Dec. 2003, Tech. Rep. [Online]. Available: https://www.nrel.gov/docs/fy04osti/34755.pdf.