A SURFACE VORTICITY METHOD
FOR WAKE–BODY INTERACTIONS

A Dissertation
Presented to
The Academic Faculty

By

David Joyner Pate

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology

May 2017

A SURFACE VORTICITY METHOD FOR WAKE–BODY INTERACTIONS

Approved by:

Professor Brian J. German, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Professor Lakshmi Sankar
School of Aerospace Engineering
Georgia Institute of Technology

Professor Marilyn Smith
School of Aerospace Engineering
Georgia Institute of Technology

Professor Graeme Kennedy
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Erik D. Olson
Aeronautics Systems Analysis Branch
NASA Langley Research Center

Date Approved: March 31, 2017

ACKNOWLEDGEMENTS

While this thesis represents about two years of work, my journey really started a long time ago, and there are many people I would like to thank. Thanks to my parents for all their support, especially Mom for taking an active role in my education. My wife, Cindy, for gently reminding me that seven years is a long time to spend as a grad student. My brother, Michael, for guiding the countless hours of music I listened to during grad school. And, of course, my thesis advisor, Prof. Brian German, for taking a chance on me and helping me become the researcher I am today.

Thanks Matt Daskilewicz for helping me get started and for raising the bar. Youngjun Choi for being a little bit crazy. Michael Patterson for helping me through those first few years—I think we complemented each other well. Andrea and Giada for teaching me how to speak proper Italian. Marc Canellas for always being that ray of sunshine when I'm feeling blue. Xiaofan Fei for playing a few games with me every now and then. And thanks to the rest of the past and present GRG members. We've had some good times.

Of course, I must also acknowledge the teachers and professors I've had over the years who took the time to get to know me, and, as a result, helped me immensely.

TABLE OF CONTENTS

LIST OF FIGURES

NOMENCLATURE

The following list describes the symbols used in this thesis. Bold notion indicates a vector in 3-space. The overhead arrow notation indicates a column matrix.

$\boldsymbol{\alpha}$	vortex particle strength
$\boldsymbol{\gamma}$	surface vorticity
$\boldsymbol{\omega}$	vorticity
$\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3$	fixed inertial reference frame and orthonormal basis vectors
$\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3$	body-fixed reference frame and orthonormal basis vectors
$\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3$	triangle orthonormal basis vectors
$\boldsymbol{d}_1, \boldsymbol{d}_2, \boldsymbol{d}_3$	vertex orthonormal basis vectors
$\boldsymbol{n}$	normal vector
$\boldsymbol{p}$	vertex position vector
$\boldsymbol{r}$	relative vector
$\boldsymbol{v}$	velocity
$\boldsymbol{x}$	position
δ	smoothing parameter; Dirac delta function
Γ	circulation
$\Gamma_{1a}, \ldots, \Gamma_{3b}$	half-edge circulation strengths
ν	kinematic viscosity
ϕ	scalar potential field
ρ	density
$\vec{\lambda}$	list of Lagrange multipliers
$\vec{b}$	list of right-hand-side terms
$\vec{x}$	list of degrees of freedom
$\vec{z}$	list of degrees of freedom in nonlinear optimization scheme
A	matrix

A_Γ	circulation loops matrix
A_{eq}	equality constraints matrix
A_{nf}	normal flow matrix
A_{ppc}	panel perimeter circulation matrix
$a_{00}, \ldots, b_{01}$	linear surface vorticity coefficients
C	path
C_p	pressure coefficient
f, g	surface vorticity scalare functions
M	equality-constrained quadratic program matrix
N_{dof}	number of degrees of freedom
N_T	number of triangles
N_V	number of vertices
P	vertex
p	pressure
S	surface
s	path parameter; wing semispan
t	time
u, v	triangle Cartesian coordinates
V	volume
v_∞	freestream velocity
W	weighting matrix
w	weighting factor
x, y, z	Cartesian coordinates

SUMMARY

The objective of this dissertation research is to develop a surface vorticity method for simulating high Reynolds number incompressible aerodynamic flows with strong unsteady interactions between wakes and lifting bodies. Examples of these types of flows include rotors in hover, propeller/wing installations, and impingement of vortex cores shed from wing strakes or flaps on downstream surfaces. Although higher-order panel codes provide good representation of potential flow around lifting bodies, their treatment of wakes is inadequate for our purpose. In the absence of significant boundary layer separation, the vorticity in these flows concentrates into thin shear layers. Therefore, vortex sheets are a natural mathematical representation of these flows.

We leverage and extend rigorous methods from the vortex methods literature to model a wake as a free vortex sheet discretized as a triangulation of panels with linearly varying surface vorticity. The vorticity evolution equation is solved approximately by maintaining constant circulation along each half-edge in the triangulation, an approach that generalizes current methods for constant-strength elements. The vortex sheet is regularized with a smoothing parameter which provides an apparent thickness that mimics the limited viscous mixing in high Reynolds number flow. An adaptive paneling algorithm is implemented to maintain the desired level of detail as the wake triangulation stretches and deforms. The induced velocities from the wake vortex sheet are computed with a treecode implemented on a graphics processing unit (GPU) to allow computations with millions of panels.

Lifting bodies are modeled with bound vortex sheets that are also triangulated with linear strength panels. These higher-order vorticity elements provide accurate velocity predictions on and near the surface, allowing for high resolution streamline tracing. Surface vorticity is determined by enforcing flow tangency constraints at each triangle centroid, zero circulation around each panel perimeter, and the unsteady pressure matching Kutta condition. These constraints result in an overdetermined system that is solved in a least squares formulation. Thus, our method is a second-order surface vorticity boundary element method that combines both solid bodies and wakes in a rigorous and consistent manner.

The results of the method are shown to compare favorably to wind tunnel experimental results, including wake profiles, for a rectangular wing in a steady freestream, and for a horizontal axis wind turbine. Finally, we demonstrate the capabilities of our method in the context of strong wake–body interactions by simulating two flying wing aircraft in close formation, with the wake from the leading aircraft impacting the tailing aircraft.

CHAPTER 1

INTRODUCTION

Aerodynamic flows in which lifting bodies are tightly coupled with the wakes they shed are particularly challenging to simulate. A few common examples include: a hovering helicopter rotor shedding a wake that remains very close to the rotor; multiple propellers closely distributed along a wing shedding wakes that flow around the wing; and a wing with deflected flaps shedding wakes that form strong vortex cores and then pass near a tail surface. In each of these cases, there is a strong mutual interaction between the bodies and the wakes, which implies that the accurate simulation of one (e.g., unsteady pressures acting on a body) requires the accurate simulation of the other (e.g., wake strength and location). These types of flows are also inherently unsteady and do not admit a relaxation-based solution. The purpose of this thesis is to develop a numerical method capable of simulating these interacting lifting systems for incompressible, high Reynolds number flows.

Problems of this nature have traditionally been solved by Navier–Stokes methods based on Eulerian grids. Difficulties with these methods arise from numerical dissipation of vorticity and the challenges with creating intricate volume grids. Panel methods based on Laplace's equation have not been a viable alternative for high-fidelity simulations because the typical techniques for wake modeling are insufficient at adapting to the complex wake structures in these flows.

However, Lagrangian methods based on the Helmholtz vorticity transport equation have matured to become a realistic alternative to grid-based Navier–Stokes methods for simulating vorticity-dominated flows [1]. Indeed, incompressible flow can be described by vorticity alone, and in high Reynolds number flow, the vorticity is confined to thin layers because it diffuses much slower than it advects. Thus, the flow may be described by the distribution of vorticity concentrated in thin sheets instead of by distributions of primitive flow variables in a volume grid throughout the entire flow field. Additionally, Lagrangian methods have the benefit of automatically adapting the computational elements to concentrate where they are needed [2].

One challenge with Lagrangian methods has been the interaction with solid boundaries. Also, these methods have historically suffered from the high computational cost of calculating the mutual influence of elements with the Biot–Savart law. Fortunately, fast summation techniques originally developed for N-body problems in gravitation and electrostatics have recently alleviated this limitation.

1.1 Literature Review

Among the first numerical models for simulating unsteady wakes is that of Belotserkovskii and Nisht [3]. They use vortex filaments arranged in a quadrilateral-based lattice that is algebraically equivalent to vortex rings. The vortex rings convect under their own mutual influence while maintaining a constant circulation strength. A wing is treated as thin and also represented by vortex rings with a no-normal-flow condition enforced at the centroids. New vortex rings are shed from the trailing edge and are assigned a circulation strength to satisfy the unsteady Kutta condition. This method, most often called the *unsteady vortex lattice method*, was further developed [4–12], including in PMARC [13], in which the thick wing is modeled. Katz and Plotkin [14] codified the standard form of this method, and it remains popular due to its simplicity [15–20]. Versions of this method have also been applied to helicopter rotors, including the work of Quackenbush et al. [21], Wachspress et al. [22], and others [23–25]. Unfortunately, the unsteady vortex lattice method does not model vortex sheet evolution in a sufficiently rigorous and robust manner to simulate the motion of a vortex sheet as it approaches the immediate neighborhood of a body.

Several authors have paired high-order panel methods with wakes represented by a lattice of vortex filaments or vortex particles [26–34]. For example, Willis [35, 36] implemented this approach with a fast multipole tree algorithm to accelerate the vortex particle induced velocity computations. Similar to the unsteady vortex lattice method, these works also use a fixed grid-based representation of the wake. By not refining the wake as it evolves, the panels are stretched beyond any meaningful physical representation. For particle methods, this behavior manifests in unacceptably large gaps between nearby particles. As stated by Voutsinas [37], "the stretching of the wake surface grid is not totally unphysical . . . what is unphysical is to keep the same grid and disregard the changes due to deformation."

Low-order panel methods have also been paired with particle-based wakes, as exemplified by the work of Calabretta and McDonald [38]. In contrast with the heretofore discussed works, Martin [39] adaptively refined the wake based on a vortex particle splitting scheme introduced by Mansfield et al. [40]. In this scheme, a newly created particle is assigned three mutually orthogonal vectors that evolve alongside the particle strength vector. When the length of a vector exceeds a chosen threshold, the particle is split into two particles placed in-line with the vector. Unfortunately, Martin experienced numerical stability issues when applying this method to a boat propeller. The velocities induced by the constant doublet panels are singular at the panel edges; thus, these panels produce large, nonphysical velocities and velocity gradients nearby. This behavior adversely affects the particles that recirculate back toward the propeller hub, especially with the inclusion of the particle splitting scheme. Also, Martin did not incorporate fast-summation techniques and was therefore limited to a relatively small number of particles. Opoku et al. [41] employed a thin wing low-order panel method and vortex particle wakes in a particle–mesh method. Zhao and He [42–44] paired a vortex particle method with lifting-line theory to represent a helicopter rotor in hover and forward flight, and included the adaptive particle splitting scheme of Mansfield et al. [40]. Lastly, Park et al. [45] paired a vortex lattice method with vortex particles to represent a helicopter rotor in hover.

To simulate helicopter rotor–wake interactions, some authors have coupled a primitive-variable-based Navier–Stokes methods, which capture the compressible and viscous effects near the rotor, with a vortex particle method, which captures the vorticity in the wake. Stock et al. paired OVERFLOW, a Navier–Stokes method with Chimera overset grids, with a Lagrangian vortex particle method [46]. By including a fast multipole treecode and executing their programs on an Intel/Tesla cluster computer, they simulated a helicopter rotor in forward flight with over 55M particles and 46M grid points. Zhao and He [47] also coupled OVERFLOW with a vortex particle method and saw improved predictions for a UH-60A rotor as compared to a solitary OVERFLOW simulation. Additionally, Rajmohan et al. [48] implemented a coupled method with OpenFoam, an unstructured finite volume Navier–Stokes flow solver, and He and Rajmohan [49] used FUN3D, an unstructured Reynolds-averaged Navier–Stokes (RANS) method, to model a helicopter fuselage.

Several authors have noted the difficulty faced by traditional Navier–Stokes CFD methods when modeling vorticity-dominated flows, such as the blade–wake interactions with a helicopter rotor. Caradonna [50] states: "…it is clear that resolving scales of the order of the vortex core diameter imposes an enormous burden on such a solution," and that the main reason for this grid resolution is to avoid the numerical dissipation of vorticity inherent in a finite-differencing method. Similarly, Strawn and Djomehri [51] state: "…current numerical schemes and computational grids are unable to preserve the vortices long enough to predict accurately the noise and airloads caused by blade–vortex interactions." For more recent work in this area, see Eshcol et al. [52], and for a review of the literature see Strawn et al. [53] and Hariharan et al. [54].

However, the numerical dissipation of vorticity can be mitigated by solving the incompressible Navier–Stokes equations in vorticity–velocity form (see Equation 2.4). As noted by Speziale [55] and Gatski [56], an additional benefit of this approach is a simpler handling of the far field boundary conditions. Brown [57] applied the vorticity–velocity approach to capture blade–wake interactions for a helicopter rotor. An efficient solution on a volume grid is obtained via a fast Poisson solver. Brown noted that one of the main challenges in these simulations is that the vehicle dynamics of interest are on a much larger timescale than the aerodynamics. They applied the fast multipole method to calculate velocity on the volume grid, and they included an adaptive grid generation technique that created and destroyed grid nodes to follow the regions of intense vorticity. Brown and Line model a helicopter fuselage with vortex rings (i.e., vortex filaments), and they represent the rotor with an unsteady lifting line formulation. Additional applications of the vorticity transport method include Scheurich and Brown [58], and Whitehouse and Boschitsch [59]. Smith et al. [60] compared the vorticity transport method to the primitive-variable Navier–Stokes solver FUN3D.

Finally, the vorticity transport method has also been coupled with traditional CFD methods to solve the compressible, viscous flow near the blade with primitive variables, and then to solve the external vorticity-dominated flow in vorticity–velocity form. Whitehouse et al. [61] paired the vorticity transport method with OVERFLOW, FUN3D, and RSA3D (an unstructured RANS method) to model helicopter aerodynamics. Similarly, Whitehouse and Tadghighi [62] paired a vorticity–velocity formulation with OVERFLOW.

This overview of related works is not comprehensive. Indeed, these and similar methods for modeling wings and wakes have been combined in numerous ways to varying degrees of success. Nonetheless, at present, the aerodynamic simulation of tightly coupled wakes and lifting bodies remains challenging, regardless of the computational approach.

1.2 Research Hypothesis and Contributions

As evidenced by this diverse set of literature, there are many methods and hybrid combinations of methods for modeling wings and wakes, each with certain advantages and disadvantages, such as accuracy, computational expense, and ease of use. A gap within these prior works is the lack of a higher-order surface element representation of the vorticity within the flow.

The fundamental hypothesis of this research is that interacting aerodynamic flows with strong couplings between wakes and lifting bodies can be modeled efficiently and effectively as vortex sheets discretized into triangular panels with linearly varying surface vorticity. Accordingly, the prime contribution of this thesis is a method for accurate simulations of wake–body interactions, which is enabled by multiple individual mathematical and numerical contributions.

To this end, we develop a boundary element method for which the surface of an arbitrary body is represented by a *bound* vortex sheet discretized into a triangular mesh with a linear variation of surface vorticity along each triangle. This approach is an original contribution, as compared to previous low-order or thin-surface methods. Additional contributions comprise a zero panel perimeter circulation constraint to achieve a solenoidal vorticity field, a vector rotation scheme that provides a fully continuous surface vorticity field, and a least squares solution approach.

A wake is represented by a *free* vortex sheet, also discretized in the same manner. This higher-order representation of the wake elements and the corresponding vorticity update scheme is a novel extension of the constant strength elements used by Stock [63] and Feng [64]. An additional contribution is a novel Delaunay edge flipping scheme, which helps maintain a more regular triangulation. Finally, our GPU-based implementation of a treecode is also an important contribution, enabling the large number of computational elements required for high resolution simulations.

In Chapters 2 and 3, the free and bound vortex sheet formulations and simulation methods are derived independently from the Helmholtz decomposition of velocity, with incremental validation tests presented along the way. In Chapter 4 we present our full numerical method, combining bound and free vortex sheets to model complex wake–body interactions. Then, we compare predictions from this method to two sets of wind tunnel experiments. Finally, we demonstrate the application of this method with an example simulation in which two aircraft are flying in close formation with the wake of the lead aircraft impinging upon the tailing aircraft.

CHAPTER 2

FREE VORTEX SHEETS

Vortex sheets are often used to model incompressible fluid flows in which vorticity is confined to thin shear layers [65]. A vortex sheet is a mathematical representation of a shear layer as a surface across which the tangential velocity is discontinuous but the normal velocity is continuous [66]. In three dimensions, a vortex sheet is defined by a distribution of vorticity concentrated along a continuous surface. Vortex sheets often arise in high Reynolds number flows in which viscous diffusion is small at the reference velocity.

The numerical simulation of vortex sheets has been an extensive area of research in the field of *vortex methods*. Cottet and Koumoutsakos [1] state: “Vortex methods have reached today a level of maturity, offering an interesting alternative to finite difference and spectral methods for high-resolution numerical solutions of the Navier–Stokes equation.” As compared to grid-based Navier–Stokes and Euler methods, Lagrangian vortex methods do not suffer from the numerical dissipation of vorticity [67] or the need to create an appropriate volume mesh.

In the next section, we present the fundamental equations for vortex sheets in exact form. In Section 2.2 we discuss the relevant literature—general concepts in the field of vortex methods as well as similar works. In Section 2.3 we present our numerical method for vortex sheet evolution, including an adaptive paneling algorithm and a treecode for fast summation. Finally, Section 2.4 concludes with a common example from the vortex methods literature.

2.1 Governing Equations

In three dimensions the constant-density Navier–Stokes equations comprise conservation of momentum

$$\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla \boldsymbol{v} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \boldsymbol{v} \tag{2.1}$$

and conservation of mass

$$\nabla \cdot \boldsymbol{v} = 0, \tag{2.2}$$

where $\boldsymbol{v}$ is velocity, t is time, ρ is density, p is pressure, and ν is kinematic viscosity.

Vorticity is defined as the curl of velocity,

$$\boldsymbol{\omega} := \nabla \times \boldsymbol{v}, \tag{2.3}$$

and by taking the curl of Equation 2.1, an alternate form is obtained in terms of vorticity:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \boldsymbol{v} \cdot \nabla \boldsymbol{\omega} = \boldsymbol{\omega} \cdot \nabla \boldsymbol{v} + \nu \nabla^2 \boldsymbol{v}. \tag{2.4}$$

This equation is often referred to as the *vorticity evolution equation* [1, page 6],[14, page 24].

Helmholtz's second vorticity theorem requires that vortex lines move through a fluid as material lines [68–70], enabling a Lagrangian description of the flow. The position of a Lagrangian marker $\boldsymbol{x}$ starting at $\boldsymbol{x}_0$ is therefore given in differential form as

$$\frac{d\boldsymbol{x}(\boldsymbol{x_0}, t))}{dt} = \boldsymbol{v}(\boldsymbol{x_0}, t),\ \boldsymbol{x_0} := \boldsymbol{x}(\boldsymbol{x_0}, 0), \tag{2.5}$$

which may be solved by integrating the velocity of the fluid as an ordinary differential equation.

2.1.1 Velocity–Vorticity Kinematic Relations

Equation 2.3 provides vorticity in terms of velocity; however, we will need velocity in terms of vorticity, which is obtained from velocity–vorticity kinematics.

The Helmholtz decomposition stipulates that any vector field may be expressed as the sum of a solenoidal vector field and an irrotational vector field [69, page 84],[63, page 4]. Then, for any general velocity field $\boldsymbol{v}$, we may specify

$$\boldsymbol{v} = \boldsymbol{v}_\omega + \boldsymbol{v}_\phi \tag{2.6}$$

with the solenoidal vector field $\boldsymbol{v}_\omega$ satisfying

$$\nabla \times \boldsymbol{v}_\omega = \boldsymbol{\omega}, \quad \nabla \cdot \boldsymbol{v}_\omega = 0, \tag{2.7}$$

and the irrotational vector field $\boldsymbol{v}_\phi$ satisfying

$$\nabla \times \boldsymbol{v}_\phi = \boldsymbol{0}, \quad \nabla \cdot \boldsymbol{v}_\phi = \sigma, \tag{2.8}$$

where σ is the local volumetric rate of expansion and $\boldsymbol{\omega}$ is the local vorticity. The irrotational field can be related to a scalar function ϕ as $\boldsymbol{v}_\phi = \nabla\phi$.

When combined, the equations governing $\boldsymbol{v}_\omega$ produce a Poisson equation

$$\nabla^2 \boldsymbol{v}_\omega = -\nabla \times \boldsymbol{\omega} \tag{2.9}$$

that can be solved using the infinite-medium Green's function [63, page 4],[65, page 307] as

$$\boldsymbol{v}_\omega(\boldsymbol{x}) = \frac{1}{4\pi} \int \frac{(\boldsymbol{x}' - \boldsymbol{x}) \times \boldsymbol{\omega}(\boldsymbol{x}')}{||\boldsymbol{x}' - \boldsymbol{x}||^3} dV(\boldsymbol{x}'), \tag{2.10}$$

which is often called the Biot–Savart law. Equation 2.6 may now be rewritten as

$$\boldsymbol{v}(\boldsymbol{x}) = \frac{1}{4\pi} \int \frac{(\boldsymbol{x}' - \boldsymbol{x}) \times \boldsymbol{\omega}(\boldsymbol{x}')}{||\boldsymbol{x}' - \boldsymbol{x}||^3} dV(\boldsymbol{x}') + \boldsymbol{v}_\phi, \tag{2.11}$$

which provides velocity in terms of vorticity and an irrotational velocity field.

2.1.2 Vortex Sheet Evolution Equations

A vortex sheet is represented by an orientable surface S paired with a vector field $\boldsymbol{\gamma}$ in the local tangent space of S. This vector field is called the *surface vorticity* and determines the strength of the vortex sheet. As described by Stock [67] and Sohn [71], surface vorticity is defined as

$$\boldsymbol{\gamma} := \boldsymbol{n} \times (\boldsymbol{v}^+ - \boldsymbol{v}^-), \tag{2.12}$$

where $\boldsymbol{n}$ is the unit vector normal to S oriented outward, $\boldsymbol{v}^+$ is the velocity immediately above (or outside) S, and $\boldsymbol{v}^-$ is the velocity on the opposite side. The sheet moves with the average of the velocity on either side

$$\boldsymbol{v}_{\text{sheet}} = \tfrac{1}{2}(\boldsymbol{v}^- + \boldsymbol{v}^+). \tag{2.13}$$

This notation is demonstrated in Figure 2.1.

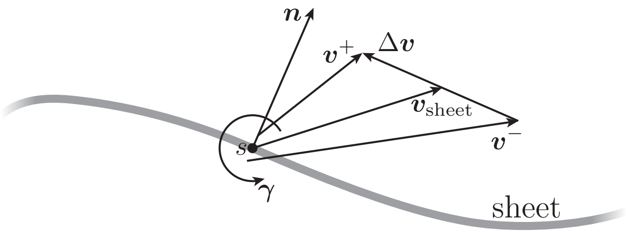


Figure 2.1: Notation for vector quantities associated with a vortex sheet.

Pozrikidis [72] derived the vortex sheet evolution equation by applying the inviscid momentum equation to $\boldsymbol{v}^+$ and $\boldsymbol{v}^-$ and relating the results via Equations 2.12 and 2.13:

$$\frac{\partial \boldsymbol{\gamma}}{\partial t} = \underbrace{-\,\boldsymbol{v}\cdot\nabla\boldsymbol{\gamma}}_{\text{advection}}\underbrace{+\,\boldsymbol{\gamma}\cdot\nabla\boldsymbol{v}}_{\text{stretch}}\underbrace{-\,\boldsymbol{\gamma}(\nabla\cdot\boldsymbol{P}\cdot\boldsymbol{v})}_{\text{dilation}}. \tag{2.14}$$

$\boldsymbol{P}$ is the tangential projection operator, which is a dyadic (second order tensor) defined as

$$\boldsymbol{P} := \boldsymbol{I} - \boldsymbol{n}\boldsymbol{n}, \tag{2.15}$$

where $\boldsymbol{I}$ is the identity dyadic, $\boldsymbol{n}$ is the sheet normal vector, and the juxtaposition of two vectors indicates the dyadic product [73, page 67]. The advection and stretch terms are common to the vorticity evolution equation; the dilation term is unique to vortex sheets and accounts for changes in surface vorticity parallel to the direction of its strain [63].

The surface vorticity is related to vorticity via the Dirac delta function as $\boldsymbol{\omega} = \boldsymbol{\gamma}\delta(n)$, where n is the distance normal to the sheet. This relation allows Equation 2.11 to be restated in terms of surface vorticity as

$$\boldsymbol{v}(\boldsymbol{x}) = \frac{1}{4\pi}\int \frac{(\boldsymbol{x}' - \boldsymbol{x}) \times \boldsymbol{\gamma}(\boldsymbol{x}')}{||\boldsymbol{x}' - \boldsymbol{x}||^3} dS(\boldsymbol{x}') + \boldsymbol{v}_\phi. \tag{2.16}$$

Next, the circulation around a closed path is defined as

$$\Gamma := \oint \boldsymbol{v}\cdot d\boldsymbol{l}. \tag{2.17}$$

To evaluate the circulation along a portion of a vortex sheet, consider a path C_0 along the sheet from point P_1 to point P_2 and a corresponding closed loop C that parallels C_0 and pierces the sheet at points P_1 and P_2, as demonstrated in Figure 2.2. Parameterize path C_0 as $C_0(s)$ with $C_0(s_1) = P_1$ and $C_0(s_2) = P_2$.

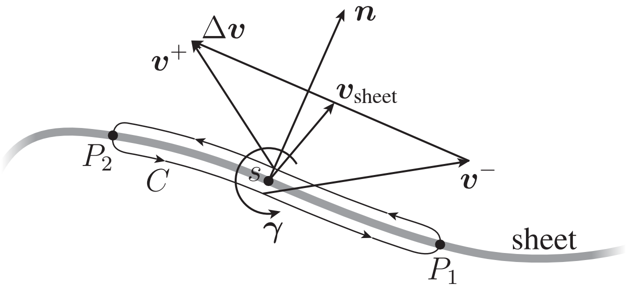


Figure 2.2: The notation for calculating the circulation around closed loop C. This is a side view of the vortex sheet.

Then, using the relation

$$\boldsymbol{v}^+ - \boldsymbol{v}^- = \Delta\boldsymbol{v} = \boldsymbol{\gamma} \times \boldsymbol{n}, \tag{2.18}$$

the circulation around C can be expressed in terms of surface vorticity as

$$\begin{aligned}
\Gamma_C &= \oint_C \boldsymbol{v} \cdot d\boldsymbol{l} \\
&= \int_{C_0} \boldsymbol{v}^+ \cdot d\boldsymbol{l} + \int_{-C_0} \boldsymbol{v}^- \cdot d\boldsymbol{l} \\
&= \int_{s_1}^{s_2} \boldsymbol{v}^+(s) \cdot d\boldsymbol{l}(s) + \int_{s_2}^{s_1} \boldsymbol{v}^-(s) \cdot d\boldsymbol{l}(s) \\
&= \int_{s_1}^{s_2} (\boldsymbol{v}^+(s) - \boldsymbol{v}^-(s)) \cdot d\boldsymbol{l}(s) \\
&= \int_{s_1}^{s_2} \Delta\boldsymbol{v} \cdot d\boldsymbol{l}(s) \\
&= \int_{s_1}^{s_2} (\boldsymbol{\gamma}(s) \times \boldsymbol{n}(s)) \cdot d\boldsymbol{l}(s).
\end{aligned} \tag{2.19}$$

This result leads to a fundamental relation between circulation and surface vorticity. Consider an arbitrary unit vector $\boldsymbol{m}$ in the tangent plane of the surface at an arbitrary point $\boldsymbol{x}$. Then, by differentiation under the integral sign, the rate of change of circulation in the direction of $\boldsymbol{m}$ at $\boldsymbol{x}$ is

$$\frac{d\Gamma_{\boldsymbol{m}}(\boldsymbol{x})}{ds} = \Delta\boldsymbol{v} \cdot \boldsymbol{m} = (\boldsymbol{n} \times \boldsymbol{m}) \cdot \boldsymbol{\gamma}(\boldsymbol{x}). \tag{2.20}$$

Furthermore, if the tangent space at $\boldsymbol{x}$ is spanned by orthonormal basis $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ with respective Cartesian coordinates (u, v), the total differential of circulation is

$$d\Gamma = \boldsymbol{b}_2 \cdot \boldsymbol{\gamma} du - \boldsymbol{b}_1 \cdot \boldsymbol{\gamma} dv. \tag{2.21}$$

2.2 Literature Review for Modeling Vortex Sheet Evolution

We now review the body of literature broadly categorized as *vortex methods*, with a limitation to methods in which discrete Lagrangian elements carry vorticity and represent a thin vortex sheet. The three-dimensional methods with sheet elements share the most similarities with our method.

2.2.1 Vortex Sheets in Two Dimensions

Point Vortex Method

Numerical approximations of a vortex sheet originated with the *point vortex method* (also called the discrete vortex method), which can be traced to the early works of Rosenhead [74] and Westwater [75]. In this method, vortex sheets are represented by discrete point vortices with a specified value of circulation, which is constant with time for inviscid flow. The general procedure for time-stepping point vortices with ordinary differential equations can be traced back at least as far as Rosenhead [74]. As described by Kuwahara and Takami

[76], the equations of motion in Cartesian coordinates (x, y) for the ith particle are

$$\frac{dx_i}{dt} = \frac{-1}{2\pi} \sum_{j \neq i} \Gamma_j \frac{y_i - y_j}{r_{i,j}^2},$$
$$\frac{dy_i}{dt} = \frac{1}{2\pi} \sum_{j \neq i} \Gamma_j \frac{x_i - x_j}{r_{i,j}^2},$$
$$r_{i,j}^2 = (x_i - x_j)^2 + (y_i - y_j)^2.$$

These early calculations were done by hand and limited to a small number of particles. As a fortuitous result, the calculated roll-up patterns appeared realistic and matched the expectations of Prandtl [77]. When Birkhoff and Fisher attempted to replicate the results with a computer and with many more particles, irregularities became apparent, with the point vortices "tending toward randomization of position" [78]. Further improvements to the point vortex method were made, including requirements for convergence, and the method eventually became codified as the Birkhoff–Rott equation [79–85].

Regularization and the Vortex Blob Method

Chorin and Bernard [86] took a different approach to resolving the irregularities associated with point vortices:

> It is fairly obvious that a point vortex approximation to a vortex sheet cannot be taken too literally, since a point vortex induces a velocity field which becomes unbounded, and cannot approximate a bounded field in any reasonable norm . . . However, we conjecture that as soon as the velocity field of the point vortices is smoothed out and made bounded, i.e., the point character of the point vortices is not taken too literally, the approximation becomes reasonable.

In their method, the circumferential velocity v_θ at a radial distance r from a point vortex with strength Γ_0 is modified as

$$v_\theta(r) = \begin{cases} \dfrac{\Gamma_0}{2\pi r_c} & \text{if } r \leq r_c, \\ \dfrac{\Gamma_0}{2\pi r} & \text{if } r > r_c. \end{cases}$$

They suggest this spreading of vorticity is analogous to artificial viscosity, and the method produced improved results.

Around the same time, Kuwahara and Takami [76] took a similar approach. They also applied a viscous core model to "suppress the unessential irregularity by introducing an 'artificial viscosity.'" They modified the velocity induced by a point vortex with an exact solution to the Navier–Stokes equations for a single vortex in a viscous fluid introduced by Oseen [87] and further developed by Rott [88]. In this viscous vortex model, the circumferential velocity at a radial distance r from a point vortex with strength Γ_0 is

$$v_\theta(r,t) = \frac{\Gamma_0}{2\pi r}\left[1 - \exp\left(\frac{-r^2}{4\nu t}\right)\right],$$

and the corresponding vorticity is a Gaussian distribution

$$\omega(r,t) = \frac{\Gamma_0}{4\pi\nu t}\exp\left(\frac{-r^2}{4\nu t}\right).$$

They note that applying this viscous core model to multiple point vortices does not retain exactness because the nonlinearity of the Navier–Stokes equations does not permit superposition of independent solutions. Further implementations of this approach include Bloom and Jen [89] and Leonard [65]. In contrast to the viscous core model used by Chorin and Bernard, the Gaussian core function used by Kuwahara and Takami spreads with time.

The concept of *regularizing* the point vortex method by desingularizing the vorticity is generally referred to the *vortex blob method* and is attributed to Chorin and Bernard [86]. Krasny [90] and Anderson [91] formulated an implementation of this method which includes a smoothing parameter $\delta > 0$ to mollify the induced velocity as

$$v_\theta(r) = \frac{\Gamma_0}{2\pi\sqrt{r^2 + \delta^2}},$$

and the corresponding equations of motion are referred to as the "δ-equations." The smoothing parameter, which was originally introduced by Rosenhead [92], controls the rate at which the vortex sheet rolls up [93].

Smoothing is often insufficient to prevent the eventual formation of irregularities. Fink and Soh [81] reposition the point vortices after each iteration such that they are equally spaced and adjust the strength of each point vortex accordingly. Krasny [84, 90] applied a spectral filter to suppress noise incited by computational machine precision errors.

Also, as the vortex sheet strains, gaps form between the vortex blobs (i.e., the cut-off radii cease to overlap). Indeed, Hald [94, 95] relies on this overlapping in his proof of convergence for the vortex blob method. If these gaps coincide with high curvature, the discrete representation of the vortex sheet may also intersect itself, which should be avoided.

Krasny [96] discusses two possible solutions to avoid these gaps: start with a large enough number of vortices, or adaptively insert new vortices. The most efficient solution, and often the most practical, is the latter. Winckelmans et al. [97] refer to a more general form of this adaptive point insertion as *particle redistribution* which is conducted either locally or globally.

Further detailed descriptions of the vortex blob method may be found in references [1, 7, 65, 97, 98].

Singularity Formation

From the beginning of the development of the vortex blob method, it was soon realized that vortex sheets tend to roll up into singularities [99–102]. As phrased by Krasny [84], "the singularity which appears is an infinite jump discontinuity in the vortex sheet's curvature." While the singularity poses a computational challenge, a more general challenge is the exponential growth of the arc length of the vortex sheet. With the use of adaptive point insertion, this causes a rapid increase in the number of vortices and subsequent computational burden. Singularity formation and rapid growth of surface area is also a characteristic of 3-D vortex sheets [103].

2.2.2 Vortex Sheets in Three Dimensions

In three dimensions, the inviscid vorticity transport equation is

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \boldsymbol{v} \cdot \nabla \boldsymbol{\omega} = \boldsymbol{\omega} \cdot \nabla \boldsymbol{v}. \tag{2.22}$$

The term on the right-hand side is the vortex stretching term, which is absent in the 2-D form of this equation. Cottet and Koumoutsakos [1, page 55] state "this term fundamentally affects the dynamics of the flow; it is [particularly] responsible for vorticity intensification mechanisms that make long-time inviscid calculations very difficult." The solution technique for this term is driven by the degree of connectivity between computational elements, and it is this connectivity which characterizes 3-D vortex methods.

Vortex Particles

One of the first vortex particle methods was proposed by Beale [104, 105], in which a collection of disconnected particles are arranged to approximate a vortex sheet. The vector strength $\boldsymbol{\alpha}_p$ of particle p is initially assigned by the vorticity $\boldsymbol{\omega}_p$ and the volume v_p represented by the particle as

$$\boldsymbol{\alpha}_p = v_p \boldsymbol{\omega}_p. \tag{2.23}$$

Because the flow is incompressible, v_p is constant in time, even though the shape of the particle could deform. Similar to the vortex blob method, each particle serves as a basis function of vorticity as

$$\boldsymbol{\omega}(\boldsymbol{x}, t) = \sum_{p=1}^{N} \boldsymbol{\alpha}_p \delta(\boldsymbol{x} - \boldsymbol{x}_p), \tag{2.24}$$

where $\delta(\boldsymbol{x})$ is the 3-D Dirac delta function [106]. The strength of each particle changes with time according to

$$\frac{d\boldsymbol{\alpha}_p}{dt} = \boldsymbol{\alpha}_p \cdot \nabla \boldsymbol{v}, \tag{2.25}$$

and each particle moves with the local velocity.

Various implementations often differ by their approaches to the calculation of velocity, the calculation of the velocity gradient tensor, the vorticity distributions (regularization functions), and remeshing schemes [65, 106].

Vortex particles are not considered to be connected. This makes local remeshing techniques difficult to implement, but allows advanced techniques for modeling viscous diffusion [63, 106].

Vortex Filaments

In vortex filament methods, a vortex sheet is represented by a collection of filaments in which vorticity is concentrated along one-dimensional curves and the direction of vorticity is locally parallel to the tangent of the curve [107]. Vortex filaments are the natural generalization of the 2-D vortex blob method [65]. The space curve representing a filament may be piecewise linear or have a higher order of continuity, but it is generally defined by a set of Lagrangian markers. Each filament maintains a scalar value of circulation that is constant along its length as well as constant with time.

One difficulty with vortex filaments is that they must be initialized according to the topology of the vortex lines [107, page 70].

Sheet Elements

A vortex sheet can also be represented by sheet elements (i.e., panels). These sheet elements are often triangles, but quadrilaterals have also been used. Agishtein and Migdal [108] represent a vortex sheet with flat triangles connected as a triangulated mesh. In their method, vortex sheet strength is constant along each triangle and velocity is calculated with an exact solution to the Biot–Savart law, which is modified to include a smoothing parameter to provide apparent thickness to the sheet. The motion of the vortex sheet is then described by the motion of the vertices of the triangulated mesh, which move as Lagrangian markers (i.e., with the local fluid velocity). They re-triangulate the sheet at every time-step with a new Delaunay triangulation.

Brady et al. [109] extended the work of Agishtein and Migdal. A scalar circulation function is defined on the vortex sheet that is constant with time, and then the surface vorticity (which they call "sheet vorticity") is determined from spatial derivatives of circulation. They use cubic Bézier triangular patches to provide C^1 continuity for the mesh and circulation and C^0 continuity for the sheet vorticity. They calculate induced velocities by a Gaussian quadrature approximation to the Biot–Savart integral, which is regularized with a smoothing parameter.

Stock [63] devised a method for evolving regularized vortex sheets in which a vortex sheet is represented by a triangular mesh with vortex sheet strength defined as constant along each flat triangle. Each edge carries a scalar circulation value that is used to update vorticity at each time-step to satisfy Equation 2.14. The triangular panels adapt to excessive curvature and strain with schemes for edge splitting and adjacent-node merging. Each triangle serves as a smoothed basis function for vorticity which is interpolated onto a regular Cartesian grid, and then the Poisson equation

$$\nabla^2 \boldsymbol{v} = -\nabla \times \boldsymbol{\omega} \tag{2.26}$$

is solved with a fast Poisson solver to obtain velocity at the triangle vertices (Lagrangian markers).

Feng [64] uses triangular panels each with constant surface vorticity. The velocity induced by a panel is calculated with a quadrature scheme that considers point vortices at each triangle vertex. He implemented a tree-code based on that of Lindsay and Krasny [110], which is based on Cartesian Taylor expansions of the Biot–Savart kernel. Citing stability considerations, Feng includes a smoothing parameter in the Biot–Savart kernel to regularize the vortex sheet. The method was further developed and applied to vortex rings by Feng et al. [111].

Kandil et al. [112] devised an unsteady lifting surface method in which both the wing and wake comprised panels with linearly varying surface vorticity. Though this is actually published in the aeronautics literature, it is worth mentioning here as the only method we are aware of that used wake panels with linearly varying surface vorticity. It was not developed further, probably because the computational requirements imposed an intractable burden; a computation with 25 wing panels and 22 time steps took 10 minutes on a CYBER 175 computer.

Additional methods based on sheet elements include [72, 113–115]. As compared to particle methods, the computational elements are fully connected. This connectivity provides for a simple solution to the stretch term in the vorticity transport equation but obstructs the generalization of sheet methods to rigorously model viscous diffusion [63].

2.3 Numerical Method for Regularized Vortex Sheet Evolution

In this section, we detail the equations and algorithmic approach comprising our method for numerically simulating the evolution of a regularized vortex sheet.

A vortex sheet is represented by connected triangular panels that each carry a linear distribution of surface vorticity. The vorticity transport equation is satisfied by maintaining circulation along the edges of the triangulation as the vertices move with the local flow velocity in a Lagrangian manner, which is similar to Stock [63, 67]. Induced velocities are calculated directly from the Biot–Savart law to avoid computational grids used with Poisson solvers, because that could become cumbersome for wakes that grow far downstream. At each time-step: (1) surface vorticity is updated, (2) the velocity at each vertex is calculated,

(3) the vertices are displaced according to the chosen algorithm for solving ODEs, such as the Runge–Kutta method, and (4) the triangular panels are locally refined to maintain sufficient detail.

2.3.1 Discretization with Triangular Panels

The vortex sheet surface is discretized into flat, triangular, connected panels. The resulting *triangulation* is a union of triangles such that the intersection of any two triangles is either empty, a shared edge, or a shared vertex[1].

Each panel serves as a basis for a linearly varying vector function $\boldsymbol{\gamma}$ representing the surface vorticity, which lies in the plane of the triangle. The surface vorticity is expressed in a local coordinate system constructed from the three edge vectors belonging to a triangle. For a triangle comprising vertices (P_1, P_2, P_3) with corresponding locations $\boldsymbol{p}_1$, $\boldsymbol{p}_2$, and $\boldsymbol{p}_3$, the edge vectors are defined as

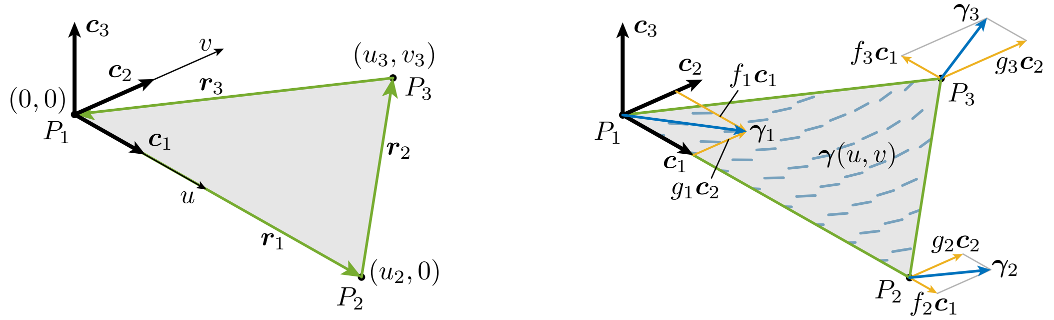
$$\boldsymbol{r}_1 := \boldsymbol{p}_2 - \boldsymbol{p}_1, \quad \boldsymbol{r}_2 := \boldsymbol{p}_3 - \boldsymbol{p}_2, \quad \boldsymbol{r}_3 := \boldsymbol{p}_1 - \boldsymbol{p}_3. \tag{2.27}$$

Each triangle is assigned a basis $(\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3)$, with

$$\boldsymbol{c}_1 := \frac{\boldsymbol{r}_1}{||\boldsymbol{r}_1||}, \quad \boldsymbol{c}_2 := \boldsymbol{c}_3 \times \boldsymbol{c}_1, \quad \boldsymbol{c}_3 := \frac{\boldsymbol{r}_1 \times \boldsymbol{r}_2}{||\boldsymbol{r}_1 \times \boldsymbol{r}_2||}, \tag{2.28}$$

in which case $\boldsymbol{c}_3$ is the triangle normal vector. This notation is illustrated in Figure 2.3a. A local Cartesian coordinate system may then be established on the triangle with coordinate u along $\boldsymbol{c}_1$, coordinate v along $\boldsymbol{c}_2$, and the origin at $\boldsymbol{p}_1$. Next, as illustrated in Figure 2.3b, the surface vorticity $\boldsymbol{\gamma}$ varies linearly along the triangle with two scalar functions

[1] This definition is accompanied by a few more characteristics described in [116, page 31]



(a) Panel vertices, edges, and coordinates

(b) Panel surface vorticity

Figure 2.3: Notation for the triangular panels.

$f, g : \mathbb{R}^2 \to \mathbb{R}$ as

$$\boldsymbol{\gamma}(u,v) = f(u,v)\boldsymbol{c}_1 + g(u,v)\boldsymbol{c}_2, \tag{2.29}$$

$$f(u,v) = a_{00} + a_{10}u + a_{01}v, \tag{2.30}$$

$$g(u,v) = b_{00} + b_{10}u + b_{01}v. \tag{2.31}$$

These coefficients are uniquely determined by the surface vorticity at each of the three vertices $\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \boldsymbol{\gamma}_3$ according to the linear system

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & u_2 & 0 \\ 1 & u_3 & v_3 \end{bmatrix} \begin{bmatrix} a_{00} & b_{00} \\ a_{10} & b_{10} \\ a_{01} & b_{01} \end{bmatrix} = \begin{bmatrix} f_1 & g_1 \\ f_2 & g_2 \\ f_3 & g_3 \end{bmatrix}, \tag{2.32}$$

where these vertex values are given by

$$\begin{aligned} u_2 &= \boldsymbol{r}_1 \cdot \boldsymbol{c}_1 & f_1 &= \boldsymbol{\gamma}_1 \cdot \boldsymbol{c}_1 & g_1 &= \boldsymbol{\gamma}_1 \cdot \boldsymbol{c}_2 \\ u_3 &= -\boldsymbol{r}_3 \cdot \boldsymbol{c}_1 & f_2 &= \boldsymbol{\gamma}_2 \cdot \boldsymbol{c}_1 & g_2 &= \boldsymbol{\gamma}_2 \cdot \boldsymbol{c}_2 \\ v_3 &= -\boldsymbol{r}_3 \cdot \boldsymbol{c}_2 & f_3 &= \boldsymbol{\gamma}_3 \cdot \boldsymbol{c}_1 & g_3 &= \boldsymbol{\gamma}_3 \cdot \boldsymbol{c}_2, \end{aligned}$$

as illustrated in Figures 2.3a and 2.3b. The solution to Equation 2.32 yields

$$\begin{aligned} a_{00} &= f_1 & b_{00} &= g_1 \\ a_{10} &= \frac{f_2 - f_1}{u_2} & b_{10} &= \frac{g_2 - g_1}{u_2} \\ a_{01} &= \frac{u_3 - u_2}{u_2 v_3} f_1 - \frac{u_3}{u_2 v_3} f_2 + \frac{1}{v_3} f_3 & b_{01} &= \frac{u_3 - u_2}{u_2 v_3} g_1 - \frac{u_3}{u_2 v_3} g_2 + \frac{1}{v_3} g_3. \end{aligned} \tag{2.33}$$

2.3.2 Vorticity Update via Half-Edge Circulations

The vortex sheet evolution equation governs the manner by which surface vorticity changes over time—that is, *updates*. Unfortunately, this equation does not welcome a direct solution. An alternate approach to update surface vorticity at each time step is to leverage an advantageous property of circulation in the Euler limit as viscosity vanishes [1, page 9],[63, page 37],[14, page 31]:

$$\frac{D\Gamma}{Dt} = 0. \tag{2.34}$$

Then, by carefully selecting many distinct paths along which circulation must remain constant, we can update surface vorticity at each time step by relating it to circulation. Stock, who used triangular sheet elements each with constant surface vorticity, maintained constant circulation along each edge of the triangulation [63, 67]. However, in our case, three scalar circulation strengths governing a triangle leaves too many degrees of freedom for γ to vary linearly along it. Instead, we consider the circulation along each half of each edge.

We can apply Equation 2.19 to calculate the circulation around each half-edge, as illustrated in Figure 2.4. Without loss of generality, consider the first half of the edge from P_1

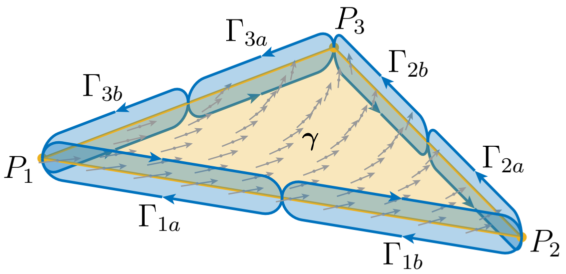


Figure 2.4: Half-edge circulations around a triangle.

to P_2 with edge vector $\boldsymbol{r}_1$. The differential path length is

$$d\boldsymbol{l} = \boldsymbol{r}_1 ds, \tag{2.35}$$

the surface vorticity varies linearly along the edge as

$$\begin{aligned}\boldsymbol{\gamma}(s) &= (1-s)\boldsymbol{\gamma}_1 + s\boldsymbol{\gamma}_2 \\ &= (1-s)(f_1\boldsymbol{c}_1 + g_1\boldsymbol{c}_2) + s(f_2\boldsymbol{c}_1 + g_2\boldsymbol{c}_2),\end{aligned} \tag{2.36}$$

and the normal vector $\boldsymbol{n}$ is the third element of basis $(\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3)$. Then, the circulation along the first half-edge is

$$\begin{aligned}\Gamma_{1a} &= \int_0^{1/2} (\boldsymbol{\gamma}(s) \times \boldsymbol{n}) \cdot \boldsymbol{r}_1 ds \\ &= \int_0^{1/2} [(1-s)(f_1\boldsymbol{c}_1 + g_1\boldsymbol{c}_2) \times \boldsymbol{n} + s(f_2\boldsymbol{c}_1 + g_2\boldsymbol{c}_2) \times \boldsymbol{n}] \cdot \boldsymbol{r}_1 ds \\ &= \int_0^{1/2} [(1-s)(g_1\boldsymbol{r}_1 \cdot \boldsymbol{c}_1 - f_1\boldsymbol{r}_1 \cdot \boldsymbol{c}_2) + s(g_2\boldsymbol{r}_1 \cdot \boldsymbol{c}_1 - f_2\boldsymbol{r}_1 \cdot \boldsymbol{c}_2)] ds \\ &= \tfrac{3}{8}(g_1\boldsymbol{r}_1 \cdot \boldsymbol{c}_1 - f_1\boldsymbol{r}_1 \cdot \boldsymbol{c}_2) + \tfrac{1}{8}(g_2\boldsymbol{r}_1 \cdot \boldsymbol{c}_1 - f_2\boldsymbol{r}_1 \cdot \boldsymbol{c}_2),\end{aligned} \tag{2.37}$$

which is a linear expression in f_1, g_1, f_2, and g_2. All six half-edge circulations for a triangle may be expressed in matrix form as

$$\begin{bmatrix} \Gamma_{1a} \\ \Gamma_{1b} \\ \Gamma_{2a} \\ \Gamma_{2b} \\ \Gamma_{3a} \\ \Gamma_{3b} \end{bmatrix} = \frac{1}{8} \begin{bmatrix} -3\boldsymbol{r}_1 \cdot \boldsymbol{c}_2 & -\boldsymbol{r}_1 \cdot \boldsymbol{c}_2 & 0 & 3\boldsymbol{r}_1 \cdot \boldsymbol{c}_1 & \boldsymbol{r}_1 \cdot \boldsymbol{c}_1 & 0 \\ -\boldsymbol{r}_1 \cdot \boldsymbol{c}_2 & -3\boldsymbol{r}_1 \cdot \boldsymbol{c}_2 & 0 & \boldsymbol{r}_1 \cdot c_1 & 3\boldsymbol{r}_1 \cdot \boldsymbol{c}_1 & 0 \\ 0 & -3\boldsymbol{r}_2 \cdot \boldsymbol{c}_2 & -\boldsymbol{r}_2 \cdot \boldsymbol{c}_2 & 0 & 3\boldsymbol{r}_2 \cdot \boldsymbol{c}_1 & \boldsymbol{r}_2 \cdot \boldsymbol{c}_1 \\ 0 & -\boldsymbol{r}_2 \cdot \boldsymbol{c}_2 & -3\boldsymbol{r}_2 \cdot \boldsymbol{c}_2 & 0 & \boldsymbol{r}_2 \cdot \boldsymbol{c}_1 & 3\boldsymbol{r}_2 \cdot \boldsymbol{c}_1 \\ -\boldsymbol{r}_3 \cdot \boldsymbol{c}_2 & 0 & -3\boldsymbol{r}_3 \cdot \boldsymbol{c}_2 & \boldsymbol{r}_3 \cdot \boldsymbol{c}_1 & 0 & 3\boldsymbol{r}_3 \cdot \boldsymbol{c}_1 \\ -3\boldsymbol{r}_3 \cdot \boldsymbol{c}_2 & 0 & -\boldsymbol{r}_3 \cdot \boldsymbol{c}_2 & 3\boldsymbol{r}_3 \cdot \boldsymbol{c}_1 & 0 & \boldsymbol{r}_3 \cdot \boldsymbol{c}_1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix} \tag{2.38}$$

Therefore, the following sets of six scalars are interchangeable for a given triangle: the half-edge circulations $\Gamma_{1a}, \ldots, \Gamma_{3b}$, the vertex surface vorticity scalars $f_1, \ldots, g_3$, and the multilinear polynomial coefficients $a_{00}, \ldots, b_{01}$.

As the vortex sheet evolves, the surface vorticity for a given triangle at a particular time-step is determined from the half-edge circulations, which are held constant, via Equation 2.38. This vorticity update scheme automatically satisfies the vortex sheet evolution equation [67].

Validation of the Vorticity Update

To validate this approach for updating surface vorticity from the half-edge circulations, we conduct two tests—the same tests introduced and used by Stock [63, page 63]. In these tests, a vortex sheet is immersed in a prescribed and constant velocity field, and the subsequent evolution of this vortex sheet is determined analytically (both position and surface vorticity). The initial conditions also establish a discretized vortex sheet, the half-edge circulations of which are calculated from the surface vorticity prescribed at each vertex with Equation 2.38. The vertices of the discretized vortex sheet move exactly according to the prescribed velocity field, and then the surface vorticity calculated from the half-edge circulations may be compared to the analytic solution.

For both tests, a solenoidal, periodic, and steady velocity field is prescribed as

$$\boldsymbol{v}(x, y, z) = \cos(2\pi y)\cos(2\pi z)\boldsymbol{a}_2 + \sin(2\pi y)\sin(2\pi z)\boldsymbol{a}_3 \tag{2.39}$$

with orthonormal basis $(\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3)$ and corresponding Cartesian coordinates (x, y, z). A planar vortex sheet is initiated at the $z = \frac{1}{2}$ plane, where $\boldsymbol{v}$ is zero along $\boldsymbol{a}_1$ and $\boldsymbol{a}_3$, which implies the vortex sheet remains planar and strains along $\boldsymbol{a}_2$. The solution to Equation 2.39 provides the position y at time t given the initial location y_0 as

$$y(y_0, t) = \frac{1}{2\pi}\tan^{-1}\left(\frac{e^{4\pi s(t+c)} - 1}{2e^{2\pi s(t+c)}}\right) + \frac{1}{2}\text{round}(2y_0), \tag{2.40}$$

where

$$s = \text{sgn}\left(-\cos(2\pi y_0)\right), \tag{2.41}$$

$$c = \frac{s}{2\pi}\ln\left(\tan(2\pi y_0) + \sqrt{\tan^2(2\pi y_0) + 1}\right), \tag{2.42}$$

and round$(\cdot)$ indicates rounding to the nearest integer. The spatial evolution of the discrete vortex sheet is demonstrated in Figure 2.5, which depicts the initial triangulation as well as the triangulation at $t = 0.25$.

The surface vorticity evolution equation is invoked to solve for the exact evolution of surface vorticity with time, and it is restated here for convenience:

$$\frac{\partial \boldsymbol{\gamma}}{\partial t} = -\,\boldsymbol{v} \cdot \nabla\boldsymbol{\gamma} + \boldsymbol{\gamma} \cdot \nabla\boldsymbol{v} - \boldsymbol{\gamma}(\nabla \cdot \boldsymbol{P} \cdot \boldsymbol{v}). \tag{2.43}$$

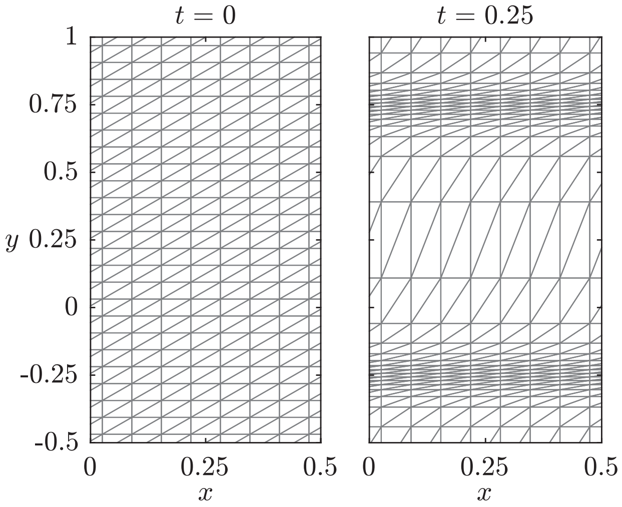


Figure 2.5: Initial and final triangulation of the vortex sheet in the $z = 1/2$ plane.

$\boldsymbol{P}$ is the tangential projection operator, which is a dyadic defined as

$$\boldsymbol{P} := \boldsymbol{I} - \boldsymbol{n}\boldsymbol{n}, \tag{2.44}$$

where $\boldsymbol{I}$ is the identity dyadic, $\boldsymbol{n}$ is the sheet normal vector, and the juxtaposition of two vectors indicates the dyadic product.

The variables are expanded in the working basis to

$$\boldsymbol{\gamma} = \gamma_x \boldsymbol{a}_1 + \gamma_y \boldsymbol{a}_2 + \gamma_z \boldsymbol{a}_3, \tag{2.45}$$

$$\boldsymbol{v} = v_x \boldsymbol{a}_1 + v_y \boldsymbol{a}_2 + v_z \boldsymbol{a}_3, \tag{2.46}$$

$$\nabla = \boldsymbol{a}_1 \frac{\partial}{\partial x} + \boldsymbol{a}_2 \frac{\partial}{\partial y} + \boldsymbol{a}_3 \frac{\partial}{\partial z}, \tag{2.47}$$

and, accordingly, the terms of Equation 2.43 are expanded to

$$\boldsymbol{v} \cdot \nabla \boldsymbol{\gamma} = (\boldsymbol{v} \cdot \nabla \gamma_x)\boldsymbol{a}_1 + (\boldsymbol{v} \cdot \nabla \gamma_y)\boldsymbol{a}_2 + (\boldsymbol{v} \cdot \nabla \gamma_z)\boldsymbol{a}_3, \tag{2.48}$$

$$\boldsymbol{\gamma} \cdot \nabla \boldsymbol{v} = (\boldsymbol{\gamma} \cdot \nabla v_x)\boldsymbol{a}_1 + (\boldsymbol{\gamma} \cdot \nabla v_y)\boldsymbol{a}_2 + (\boldsymbol{\gamma} \cdot \nabla v_z)\boldsymbol{a}_3, \tag{2.49}$$

$$\nabla \cdot \boldsymbol{P} \cdot \boldsymbol{v} = \nabla \cdot \boldsymbol{v} - [(\boldsymbol{n} \cdot \nabla v_x)\boldsymbol{a}_1 + (\boldsymbol{n} \cdot \nabla v_y)\boldsymbol{a}_2 + (\boldsymbol{n} \cdot \nabla v_z)\boldsymbol{a}_3] \cdot \boldsymbol{n} \tag{2.50}$$

Convenient choices of $\boldsymbol{\gamma}$ along with $\boldsymbol{n} = \boldsymbol{a}_3$ will lead to substantial simplification of the evolution equation.

For the first test case, the initial surface vorticity $\boldsymbol{\gamma}_0$ is parallel to the direction of strain:

$$\boldsymbol{\gamma}_0 := \boldsymbol{a}_2. \tag{2.51}$$

This results in a constant surface vorticity:

$$\gamma_x = 0, \quad \gamma_y = 1. \tag{2.52}$$

In this simple case, the numerical scheme correctly matches this solution to machine precision.

For the second test case, the initial surface vorticity is transverse to the direction of strain:

$$\boldsymbol{\gamma}_0 := \boldsymbol{a}_1, \tag{2.53}$$

which simplifies the evolution equation to

$$\frac{\partial \gamma_x}{\partial t} = -v_y \frac{\partial \gamma_x}{\partial y} - \gamma_x \frac{\partial v_y}{\partial y}, \tag{2.54}$$

with the solution

$$\gamma_x(y,t) = \frac{2\left[1+\sin(2\pi y)\right] e^{2\pi t}}{\left[1+\sin(2\pi y)\right]^2 e^{4\pi t} + \cos^2(2\pi y)}. \tag{2.55}$$

Figure 2.6 depicts the surface vorticity predicted by the numerical update scheme as compared to the analytic solution at $t = 0.25$. The error remains small compared to the initial value even for this coarse discretization (16 vertices between peaks).

Therefore, we can conclude that the half-edge circulation method is effective at resolving a vortex sheet in the presence of planar strain in directions parallel or transverse to

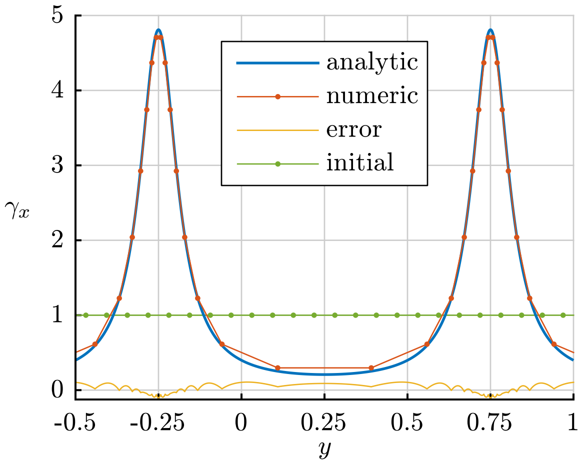


Figure 2.6: Comparison of surface vorticity at $t = 0.25$.

vorticity. As noted by Stock, the vortex blob method requires remeshing of the particles in both of these cases and the vortex filament method requires remeshing when the strain is transverse to vorticity [63, page 67].

2.3.3 Smoothing

As mentioned previously, vortex sheets have a tendency to form singularities in the center of a rolled-up region. Additionally, numerical imprecision tends to provoke instabilities [84]. These features may be limited by regularizing the vortex sheet, which we implement by including a smoothing parameter in the Biot–Savart integral [64, 90]. Smoothing provides the vortex sheet an apparent thickness that more closely resembles a high Reynolds number flow rather than an inviscid one, which is desirable [117, 118].

We employ the smoothing parameter introduced by Anderson [91] and Krasny [90] so as to maintain the same method of calculating the velocity induced by the vortex sheet (the

Biot–Savart law), which is now expressed as

$$\boldsymbol{v}(\boldsymbol{x}) = \frac{1}{4\pi} \int_S \frac{(\boldsymbol{x}' - \boldsymbol{x}) \times \boldsymbol{\gamma}(\boldsymbol{x}')}{\left(||\boldsymbol{x}' - \boldsymbol{x}||^2 + \delta^2\right)^{3/2}} dS(\boldsymbol{x}'). \tag{2.56}$$

The effect of the smoothing parameter is demonstrated in Figure 2.7, which depicts the velocity field induced by a triangular vortex sheet element for two cases: $\delta = 0$ and $\delta = 0.2$. The two induced velocity fields are sampled along the z-axis, which is normal to the triangle and pierces it at the black marker. The $\delta = 0$ case, which lacks smoothing, is discontinuous at this intersection; however, it remains bounded, which would not be the case for a filament or particle. In the smoothed case, $\delta = 0.2$, the induced velocity exhibits a smooth transition across the triangle and asymptotically approaches the $\delta = 0$ case away from the triangle. Note that for this particular surface vorticity field, the induced velocities include a component normal to the triangle.

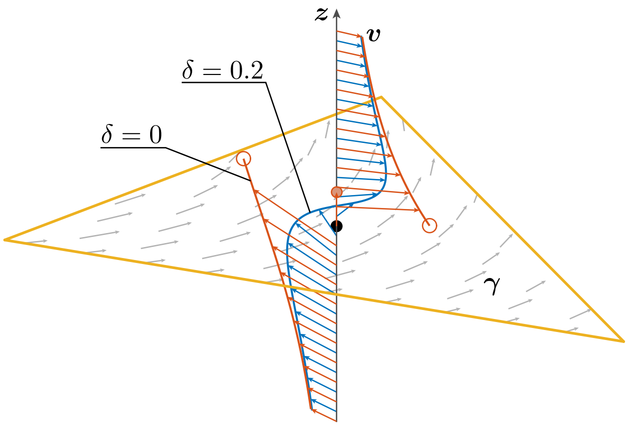


Figure 2.7: Example of the velocity field induced by a vortex sheet with smoothing and without smoothing.

The smoothed vorticity generated by this vortex sheet element can be expressed as the curl of Equation 2.16. Larger values of δ spreads the vorticity outward from the panel, and this limits the peak vorticity reached at the center of a rolled-up vortex core.

2.3.4 Adaptive Paneling

In Section 2.3.2 we demonstrated that maintaining constant half-edge circulation passively satisfies Equation 2.14 (the vortex sheet evolution equation) in the presence of planar strain. However, no such guarantee exists in the presence of large curvature, such as in a rolled-up region, and so the resolution will become insufficient to adequately resolve the vorticity [110]. To avoid this eventuality, it is necessary to adaptively refine the resolution of the triangulation, which we accomplish by splitting long edges, removing short edges, and flipping selected edges to satisfy the Delaunay condition [119].

First, any edge longer than a specified threshold is split into two edges, and a new vertex is placed at the geometric midpoint as demonstrated in Figure 2.8. We considered a spline-based point insertion technique, but we found it to be unstable, as did Kaganovskiy [113].

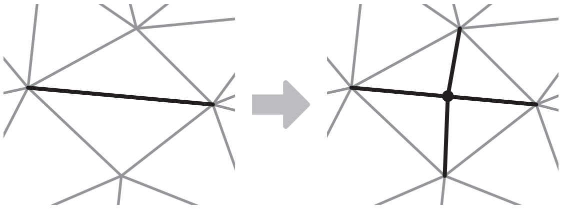


Figure 2.8: Example of triangle splitting.

Second, any edge shorter than a specified threshold is removed, thereby also removing the two triangles that share it. This step serves the opposite purpose of the previous step—to remove detail where it is not required and save computational cost. The two vertices comprising the removed edge are merged and located at their geometric midpoint, as demonstrated in Figure 2.9.

Lastly, we enforce the Delaunay condition on each edge, which stipulates that the pair

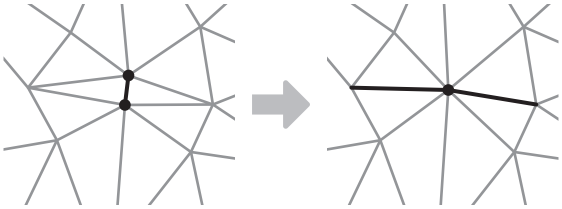


Figure 2.9: Example of edge removal.

of angles opposite an edge should sum to less than or equal to $180°$. Any edge not meeting this condition is flipped as shown in Figure 2.10. This step helps maintain a more regular triangulation where the degree[2] of each vertex is close to six.

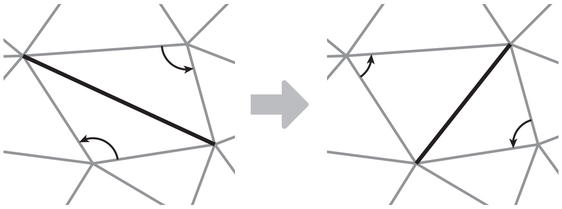


Figure 2.10: Example of Delaunay edge flipping.

The Delaunay edge-flipping procedure we implement is novel, whereas the first two procedures are common to other methods based on triangulations. An edge is prevented from being split, removed, or flipped if the degree of a neighboring vertex would exceed the minimum or maximum limits. Also, edges on the boundary of a sheet are protected from removal to prevent the boundary from degenerating.

To assess this adaptive paneling algorithm, we again leverage a test introduced by Stock [63]. A sphere of radius 0.4 is immersed in a prescribed and constant velocity field, which is constructed by placing three infinite vortex filaments with a smoothed Biot–Savart kernel. The filaments are each parallel to one of the basis vectors $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3$, respectively, and they

[2] The *degree* of a vertex is the number of edges at the vertex, or, equivalently, the number of neighboring vertices [120].

pass through

$$\boldsymbol{p}_1 = 0.2\boldsymbol{a}_2 + 0.2\boldsymbol{a}_3, \tag{2.57}$$

$$\boldsymbol{p}_2 = 0.2\boldsymbol{a}_1 + 0.2\boldsymbol{a}_3, \tag{2.58}$$

$$\boldsymbol{p}_3 = 0.2\boldsymbol{a}_1 + 0.2\boldsymbol{a}_2, \tag{2.59}$$

each with strength $\Gamma = 4\pi$. The resulting velocity at a point $\boldsymbol{p}$ is

$$\boldsymbol{v}(\boldsymbol{p}) = \sum_{i=1}^{3} \frac{\boldsymbol{r}_i \times \boldsymbol{a}_i}{\boldsymbol{r}_i \cdot \boldsymbol{r}_i + \delta^2} \tag{2.60}$$

$$\boldsymbol{r}_i := (\boldsymbol{p}_i - \boldsymbol{p}) - (\boldsymbol{p}_i - \boldsymbol{p}) \cdot \boldsymbol{a}_i \boldsymbol{a}_i \tag{2.61}$$

The test was conducted with $\delta = 0.1$ and $\Delta t = 0.001$, and it was time-stepped with the fourth-order Runge–Kutta method; the triangulation began with 20,480 triangles and ended with 111,586 triangles after 1000 iterations. Figure 2.11 illustrates the shape of the sheet as it evolved.

The metric for assessing the accuracy of the algorithm is the volume enclosed by the triangulation, which should be conserved because the velocity field is solenoidal. The results are presented in Figure 2.12. The histogram of edge lengths suggests that the edge splitting and edge removal schemes successfully maintained the 0.005 and 0.02 threshold values chosen for this test. Similarly, the histogram of vertex degrees indicates that the edge flipping scheme kept the degree of each vertex near six, which corresponds to a triangulation with mostly regular triangles. Finally, the ratio of the volume enclosed by the triangulation to the exact value remained very close to unity.

To visualize the complex geometric forms attained by a vortex sheet, we render it as

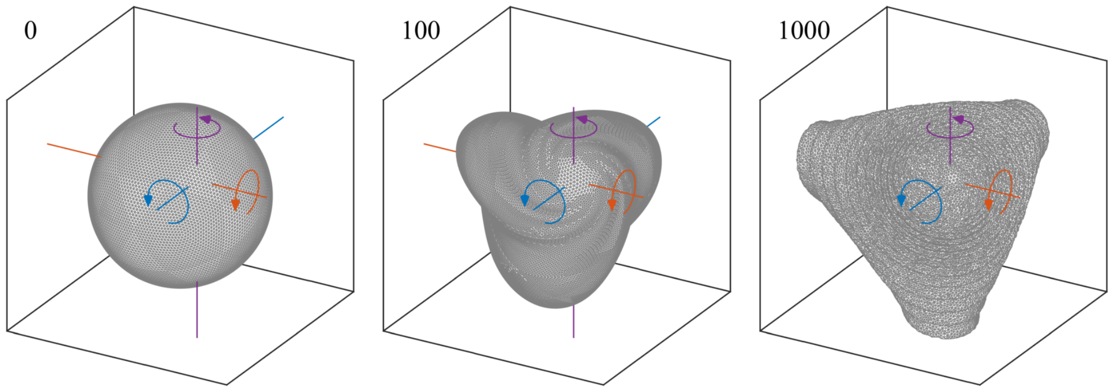


Figure 2.11: The triangulated sheet as it evolves at iterations 0, 100, and 1000. Also indicated are the infinite vortex filaments.

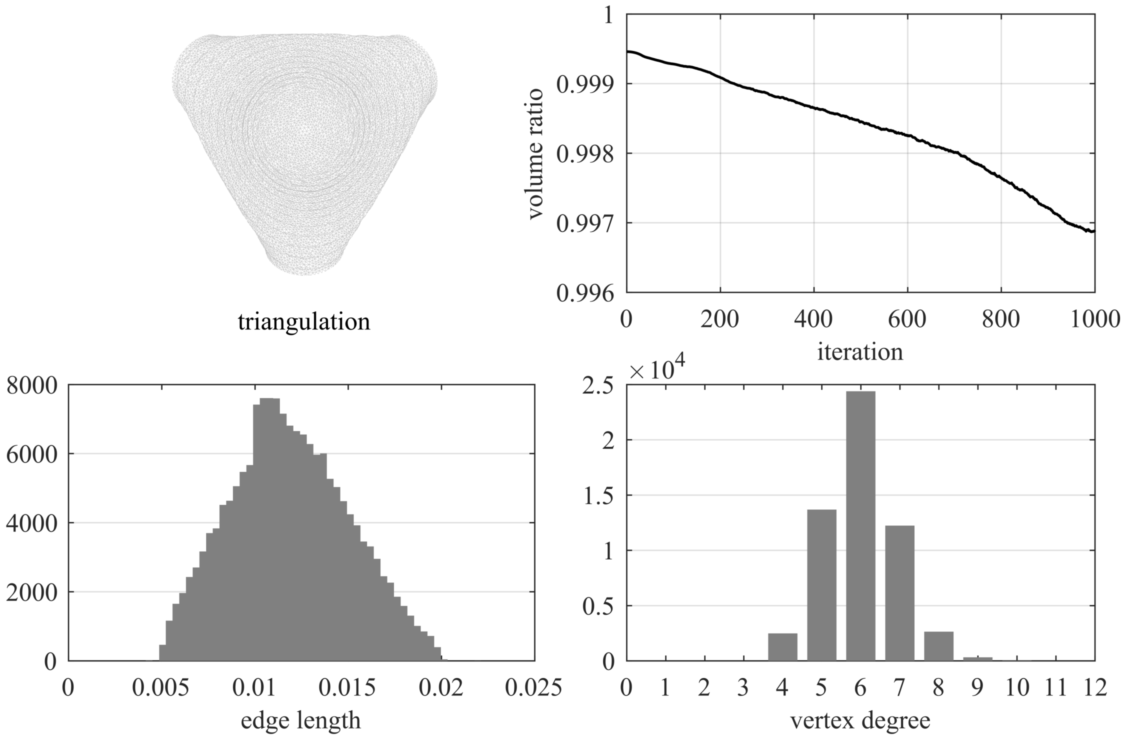


Figure 2.12: The results of the adaptive paneling algorithm test, clockwise from top left: the final triangulation, the computed volume at each iteration normalized by the exact volume of a sphere, a histogram of the edge lengths, and a histogram of the vertex degrees.

a shadowgraph, as originally inspired by the captivating figures presented by Stock [63]. Borrowing an analogy from Schlieren photography, a vortex sheet can be imagined to affect light in a similar manner as a shockwave, which is approximately a thin surface. Each panel is treated as semitransparent plate that obstructs light in proportion to the distance it travels through the plate. Therefore, the image appears darker in regions where the sheet is perpendicular to the viewing direction. The shadowgraph of the final configuration of the surface in this adaptive paneling test is illustrated in Figure 2.13. In this example, the sheet did not carry vorticity and was thus not a vortex sheet; however, this Schlieren-like technique will be used throughout this thesis to visualize vortex sheets.

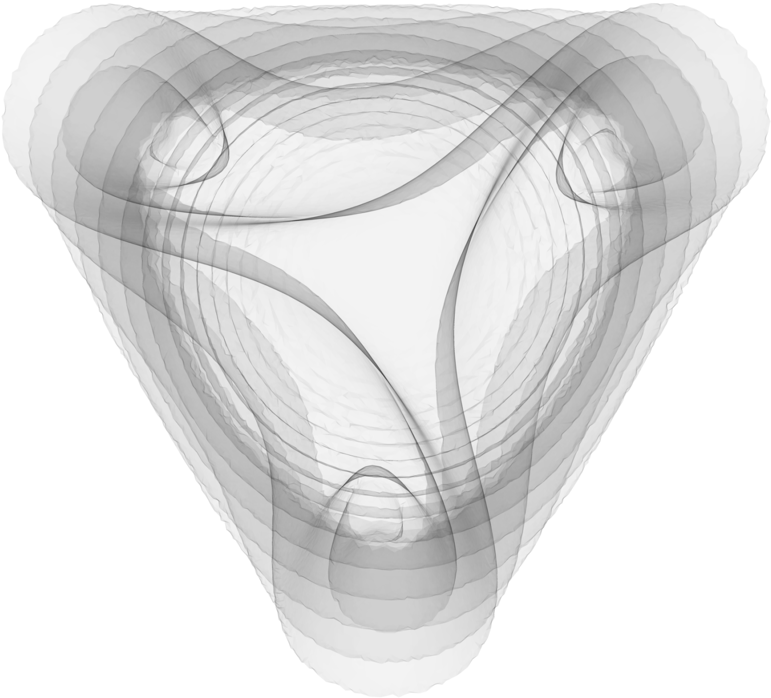


Figure 2.13: Shadowgraph of the deformed sheet at the end of the adaptive paneling test.

2.3.5 Fast Summation

The calculation of the induced velocities comprises nearly all of the computational effort spent while time-stepping a vortex sheet. For a triangulation with M triangles and N vertices, the number of computations scales as $O(MN)$, and a typical triangulation will have approximately twice as many triangles as vertices. Fast calculations will therefore be essential for complex multi-body problems with interacting wakes

The computational efficiency is improved in three ways. First, a triangle is approximated as a point vortex of equal strength when the triangle and point of velocity computation are sufficiently distant. The contribution of a triangle T to the induced velocity at $\boldsymbol{x}'$ is described by the regularized Biot–Savart law, which is then approximated as

$$\begin{aligned}\boldsymbol{v}(\boldsymbol{x}) &= \frac{1}{4\pi}\int_T \frac{(\boldsymbol{x}'-\boldsymbol{x})\times\boldsymbol{\gamma}(\boldsymbol{x}')}{(||\boldsymbol{x}'-\boldsymbol{x}||^2+\delta^2)^{3/2}}dS(\boldsymbol{x}') \\ &\approx \frac{1}{4\pi}\frac{\boldsymbol{r}}{(\boldsymbol{r}\cdot\boldsymbol{r}+\delta^2)^{3/2}}\times\int_T \boldsymbol{\gamma}(\boldsymbol{x}')dS(\boldsymbol{x}'),\end{aligned} \tag{2.62}$$

where $\boldsymbol{r}=\boldsymbol{p}_C-\boldsymbol{x}$, and $\boldsymbol{p}_C$ is the centroid of the triangle. This simplified integral provides the strength of the particle approximating the surface vorticity carried by the triangle:

$$\boldsymbol{\alpha} := \int_T \boldsymbol{\gamma}(\boldsymbol{x}')dS(\boldsymbol{x}'), \tag{2.63}$$

which is solved in terms of the polynomial coefficients presented in Section 2.3.1 as

$$\begin{aligned}\boldsymbol{\alpha} = &\left[\tfrac{1}{2}u_2v_3a_{00}+\tfrac{1}{6}(u_2+u_3)u_2v_3a_{10}+\tfrac{1}{6}u_2v_3^3a_{01}\right]\boldsymbol{c}_1 \\ &+\left[\tfrac{1}{2}u_2v_3b_{00}+\tfrac{1}{6}(u_2+u_3)u_2v_3b_{10}+\tfrac{1}{6}u_2v_3^3b_{01}\right]\boldsymbol{c}_2.\end{aligned} \tag{2.64}$$

Then, for a set of M vortex particles—approximating triangular panels—the induced velocity at the ith vertex is

$$\boldsymbol{v}_i = \frac{1}{4\pi}\sum_{j=1}^{M}\frac{\boldsymbol{r}_{ij}\times\boldsymbol{\alpha}_j}{(\boldsymbol{r}_{ij}\cdot\boldsymbol{r}_{ij}+\delta^2)^{3/2}}, \quad \boldsymbol{r}_{ij}=\boldsymbol{p}_{C,j}-\boldsymbol{p}_i. \tag{2.65}$$

Equation 2.65 requires many fewer floating point operations than the exact solution for a triangular panel provided in Appendix A. However, the computational expense of this direct summation will still scale as $O(N^2)$ and remains prohibitively expensive for large problems. Therefore, to further improve computation efficiency we invoke a vortex particle treecode. A *treecode* provides a summation that scales as $O(N \log N)$ by representing the influence of a cluster of particles on a far away point with an approximation to a specified tolerance. Along with the related fast multipole method, treecodes are common in N-body problems seen electrostatics, celestial mechanics, and vortex methods [97, 110, 121–129].

The celestial mechanics problem [121], in which a set of point masses experience a mutual gravitational pull described by Newton's law of gravitation, provides a convenient and intuitive example: the influence of a dense cluster of point masses on a distant point can be approximated by replacing the cluster by a single point mass with the same total mass placed at the mass center of the cluster. Better approximations are produced by including higher-order mass moment of inertia terms. The fast multipole method goes further by including approximations for cluster–cluster interactions.

In this work, we construct the tree in the same manner as Barnes and Hut [122], with the exception that we allow multiple particles per cell instead of only one. We leverage the work of Lindsay [128] to compute the cell moments as a Cartesian Taylor series expansion of Equation 2.65. The details of the implementation and testing of the treecode are presented in Appendix B.

Lastly, to improve computational efficiency even further, we implemented the treecode in parallel on a graphics processing unit (GPU) in the CUDA programing language, which is a recent advance in the vortex methods literature [46, 67, 125, 130, 131]. The large number of processors on a GPU, while individually slower than a CPU, operate in parallel, making a GPU well suited to compute these N-body interactions.

To demonstrate these efficiency improvements, the total computational time was measured for several summing approaches for a varying number of panels for a structured toroidal mesh. The results depicted in Figure 2.14 compare CPU vs GPU, direct summation vs treecode, and single precision vs double precision. The gray lines provide a reference to the canonical growth rates of $O(N^2)$ and $O(N \log N)$, and the data reveal that each summation algorithm asymptotically approaches the expected growth rate.

This example, as well as all of the numerical simulations in this thesis, was conducted on a 3.5 GHz Intel Core i7-3770K and a GeForce GTX TITAN Z with 5760 CUDA cores at 705 MHz clock speed, and the algorithms were written in C/C++ and CUDA.

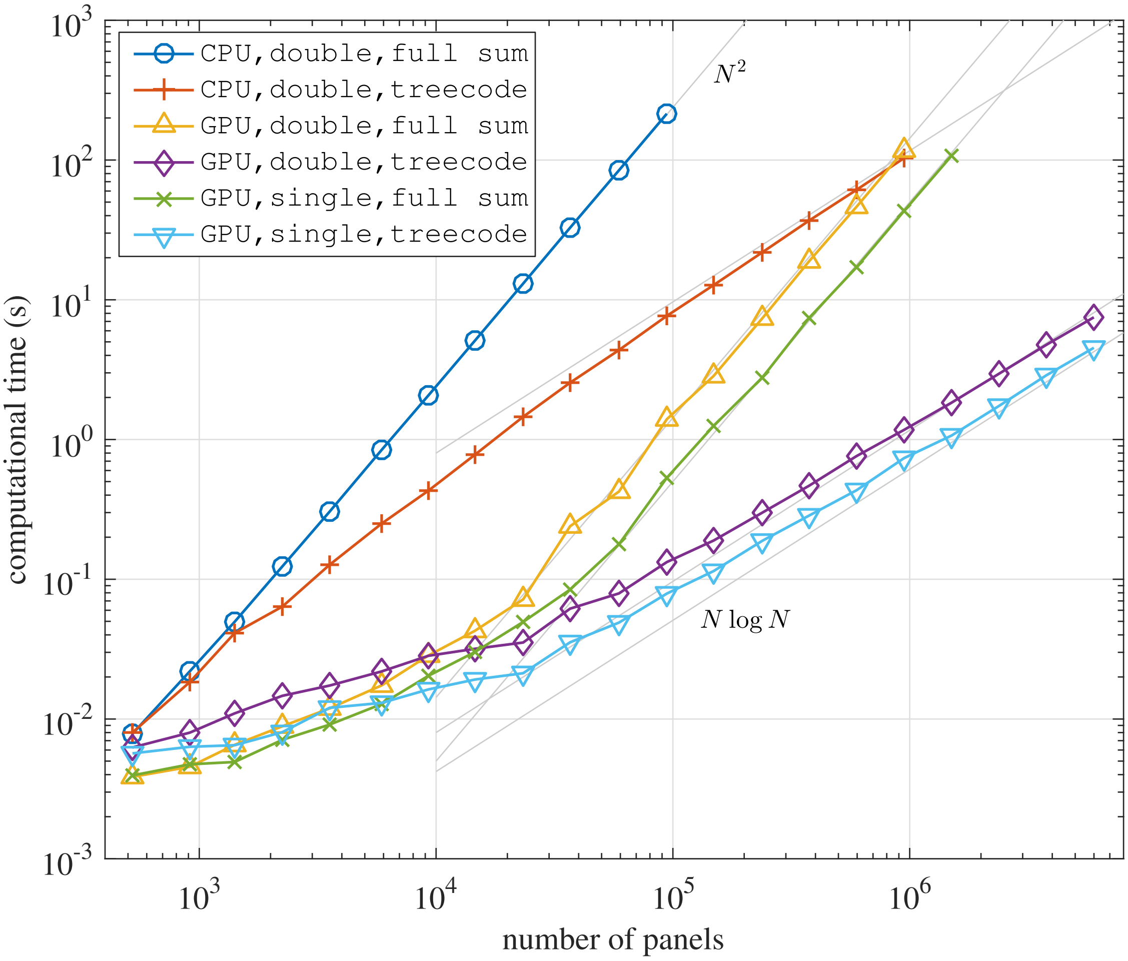


Figure 2.14: Comparison of computational time for several approaches to evaluating Equation 2.65.

2.4 Evolution of a Sphere in a Uniform Freestream

To demonstrate the efficacy of the method presented in this chapter for simulating the evolution of a free vortex sheet, we conduct the following typical example [63, 72, 103, 127, 132]. Consider a spherical vortex sheet of unit radius immersed in an otherwise incompressible uniform freestream of unit velocity. The surface vorticity initially establishes potential flow around the sphere and then evolves according to its own influence starting at time $t = 0$. This simulation began 20,480 triangles and ended with 408,590 triangles after 1000 time-steps with $\Delta t = 0.01$, $\delta = 0.1$, and using the fourth-order Runge–Kutta method.

The simulated evolution of the spherical vortex sheet is presented in Figure 2.15. The left column contains shadowgraphs of the vortex sheet (the type of imaging introduced via Figure 2.13). These images demonstrate the sheet's continuous growth in surface area and complexity with time. The majority of the sheet rolls into a ring but some escapes downstream. The right column contains *radiographs*, which depict the vorticity carried by the vortex sheet. This imaging technique was also inspired by Stock [46], and is constructed by computing the vorticity $\boldsymbol{\omega}$ on a volume grid throughout the flow and then correlating $||\boldsymbol{\omega}||$ to a grayscale, similar to the way in which dense bone obstructs radiation in a medical radiograph. Recall that the smoothing parameter δ has the effect of spreading the vorticity away from the vortex sheet. Most of the vorticity can be seen to concentrate in a toroidal ring, with only a small amount escaping downstream. Both sets of images reveal the instabilities natural to vortex sheets; these particular instabilities can be attributed to small perturbations arising from numerical roundoff error, as discussed by Krasny [84].

The accuracy of this simulation may be assessed by tracking the invariant (conserved) properties of the flow [1, 109]. In this case we examine volume and circulation. The volume is calculated directly from the triangulation, as was done in Section 2.3.4. The circulation is calculated with Equation 2.17 by integrating velocity along the closed path illustrated in Figure 2.16, which extends to $\pm\infty$ along the x-axis (the freestream direction) and then connects via a circular arc. In the limit as this path extends to infinity, the velocity along the circular arc becomes $\boldsymbol{v}_\infty$, and so only the velocity along the straight segment contributes to the integral. The exact solutions are $V^* = 4\pi/3$ and $\Gamma^* = 3$ for volume and circulation, respectively. These metrics are displayed in Figure 2.17a; there is a slight initial error in volume which then grows but remains within about 2%, while circulation error remains within about 3%. The short period noise observed in the circulation history is due to the treecode, which can produce small but abrupt jumps in velocity as slight changes in vertex location cause topological changes in the tree. Figure 2.17b demonstrates the growth in the number of triangles over time, which is approximately quadratic.

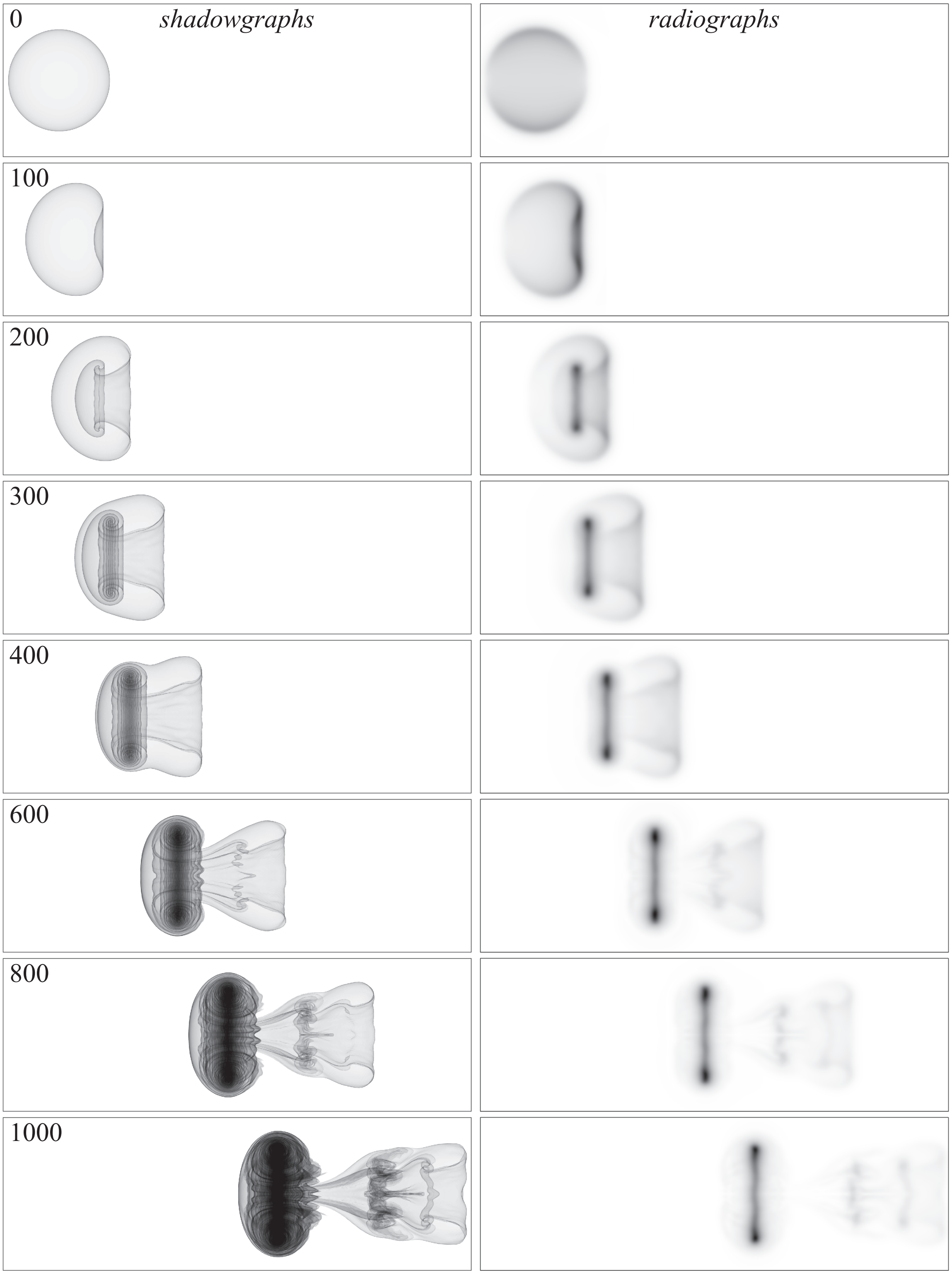


Figure 2.15: The evolution of a spherical vortex sheet over 1000 iterations. The shadowgraphs reveal the shape of the sheet while the radiographs reveal the vorticity carried by the sheet.

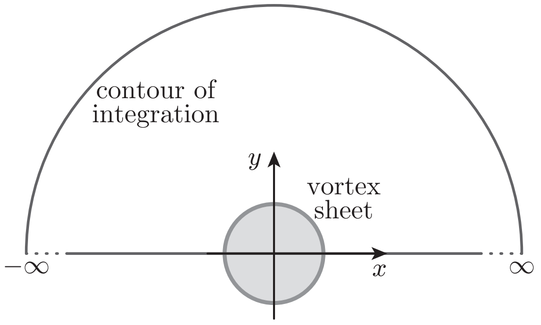
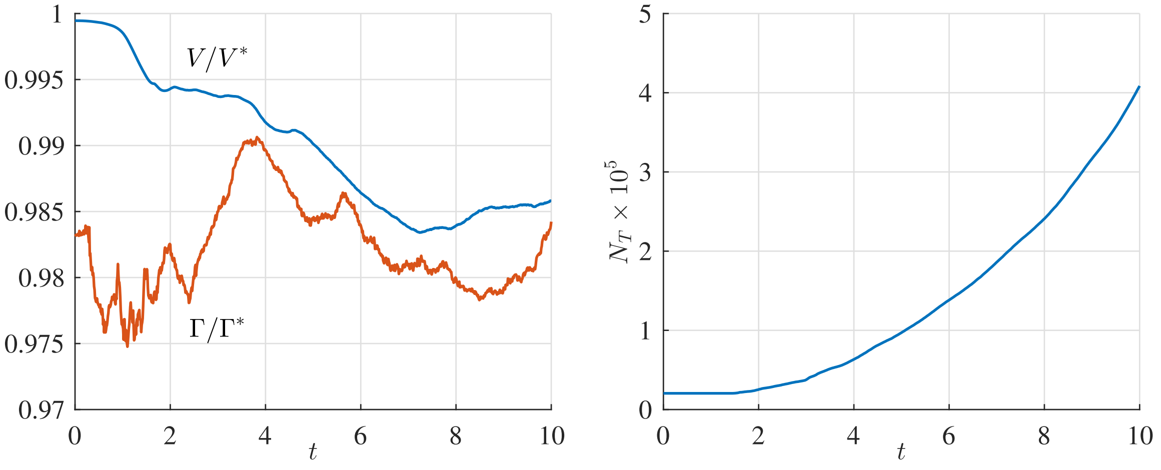


Figure 2.16: Contour of integration for calculating the circulation.



(a) Volume and circulation error ratios.

(b) Number of panels in the sheet triangulation.

Figure 2.17: The time history of the simulated evolution of a spherical vortex sheet.

CHAPTER 3

BOUND VORTEX SHEETS

The numerical calculation of potential flow around three-dimensional arbitrary bodies originated with the work of Smith, Hess, and others at the Douglas Aircraft Company [133–139]. They conceived the *panel method* in which the body is represented with a source singularity distribution on the surface, leveraging classical potential theory [136, 140]. Lifting bodies are modeled by including doublet distributions both on the body surface and in a shed wake. The singularity distributions are discretized with elementary basis functions defined on panels, thereby enabling the scalar potential function to be calculated. The resulting set of linear algebraic equations is solved numerically to yield the source and doublet distribution strengths [14, page 241]. Variations of the panel method have been developed into several successful computer programs including VSAERO [141], PANAIR [142, 143], and PMARC [13].

An alternative approach to a source–doublet formulation is to represent a body with a surface vortex sheet and to solve directly for the surface vorticity. Martensen [144] developed this approach into the *surface vorticity method* for two-dimensional bodies, also leveraging classical potential theory [140] as well as analytical work by Prager [145]. The surface vorticity method was further developed and applied to airfoil cascades and annular airfoils for use in turbomachinary flow problems [146–150].

Lewis and Ryan [149] introduced a general analytical form of the surface vorticity method in three dimensions. They took special note of the restriction that surface vorticity must have zero divergence and called this restriction a “continuity equation for the vortex sheet.” This constraint poses a unique challenge to the surface vorticity method in three dimensions that is not present in the source panel method [151]. For general body shapes, the most advanced implementations of surface vorticity methods have been the ring vortex and horseshoe vortex methods [150], variations of which are presented by Katz and Plotkin [14]. Unfortunately, these low order approaches prevent the surface vorticity method from competing with higher-order panel methods for three-dimensional flow in terms of accuracy. A high order surface vorticity method for three-dimensional flow has remained an elusive challenge in part because of the divergence obstacle.

In this chapter, we develop a new higher-order surface vorticity panel method for three-dimensional bodies. The approach is to determine the strength of the vortex sheet on the body surface such that the surface velocity is tangential and the surface vorticity is divergence-free. The only modifications for lifting flows are the shedding of a wake, which is also modeled as a vortex sheet, and the enforcement of the unsteady Kutta condition. The equations are discretized via a surface triangulation with linear variation of surface vorticity along each triangle. This discretization provides a fidelity analogous to second order source–doublet panel methods. The discretized problem formulation produces an overdetermined linear system which is solved approximately by minimizing the sum of the squares of the residuals. We assess the convergence of the method by comparing it to the analytical (exact) solution to the potential flow over an ellipsoid. Some of the equations in this section were also presented in Chapter 2, but they are repeated to provide an uninterrupted derivation.

3.1 Governing Equations for the Surface Vorticity Model

As described by Batchelor [69, page 147],[14, page 13], the motion of an incompressible, inviscid fluid is governed by the Euler momentum equation

$$\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla \boldsymbol{v} = -\frac{1}{\rho} \nabla p \tag{3.1}$$

and the continuity equation

$$\nabla \cdot \boldsymbol{v} = 0. \tag{3.2}$$

The Helmholtz decomposition stipulates that any vector field may be expressed as the sum of a solenoidal vector field and an irrotational vector field [69, page 84],[63, page 4]. Then, for any general velocity field $\boldsymbol{v}$, we may specify

$$\boldsymbol{v} = \boldsymbol{v}_\omega + \boldsymbol{v}_\phi \tag{3.3}$$

with the solenoidal vector field $\boldsymbol{v}_\omega$ satisfying

$$\nabla \times \boldsymbol{v}_\omega = \boldsymbol{\omega}, \quad \nabla \cdot \boldsymbol{v}_\omega = 0, \tag{3.4}$$

and the irrotational vector field $\boldsymbol{v}_\phi$ satisfying

$$\nabla \times \boldsymbol{v}_\phi = \boldsymbol{0}, \quad \nabla \cdot \boldsymbol{v}_\phi = \sigma, \tag{3.5}$$

where σ is the local volumetric rate of expansion and $\boldsymbol{\omega}$ is the local vorticity. The irrotational field can be obtained from a scalar function ϕ as $\boldsymbol{v}_\phi = \nabla \phi$.

When combined, the equations governing $\boldsymbol{v}_\omega$ produce a Poisson equation

$$\nabla^2 \boldsymbol{v}_\omega = -\nabla \times \boldsymbol{\omega} \tag{3.6}$$

which can be solved using the infinite-medium Green's function [63, page 4],[65, page 307] as

$$\boldsymbol{v}_\omega(\boldsymbol{x}) = \frac{1}{4\pi}\int \frac{(\boldsymbol{x}'-\boldsymbol{x})\times\boldsymbol{\omega}(\boldsymbol{x}')}{||\boldsymbol{x}'-\boldsymbol{x}||^3}dV(\boldsymbol{x}'), \tag{3.7}$$

which is often called the Biot–Savart law. Equation 3.3 may now be rewritten as

$$\boldsymbol{v}(\boldsymbol{x}) = \frac{1}{4\pi}\int \frac{(\boldsymbol{x}'-\boldsymbol{x})\times\boldsymbol{\omega}(\boldsymbol{x}')}{||\boldsymbol{x}'-\boldsymbol{x}||^3}dV(\boldsymbol{x}') + \boldsymbol{v}_\phi, \tag{3.8}$$

which is guaranteed to satisfy Equation 3.2 for any vorticity field

$$\boldsymbol{\omega} = \nabla\times\boldsymbol{v}. \tag{3.9}$$

3.1.1 Bound Vortex Sheet Definition and Continuous Problem Formulation

A vortex sheet is represented by an orientable surface S paired with a vector field $\boldsymbol{\gamma}$ representing the vortex sheet strength, which we call *surface vorticity*, existing in the local tangent space of S. As described by Stock [67] and Sohn [71], the surface vorticity is defined as

$$\boldsymbol{\gamma} := \boldsymbol{n}\times(\boldsymbol{v}^+ - \boldsymbol{v}^-), \tag{3.10}$$

where $\boldsymbol{n}$ is the unit vector normal to S oriented outward, $\boldsymbol{v}^+$ is the velocity immediately above (or outside) S, and $\boldsymbol{v}^-$ is the velocity on the opposite side. The surface vorticity is related to vorticity via the Dirac delta function as $\boldsymbol{\omega} = \boldsymbol{\gamma}\delta(n)$, where n is the distance normal to the sheet. Equation 3.8 may then be restated in terms of surface vorticity as

$$\boldsymbol{v}(\boldsymbol{x}) = \frac{1}{4\pi}\int \frac{(\boldsymbol{x}'-\boldsymbol{x})\times\boldsymbol{\gamma}(\boldsymbol{x}')}{||\boldsymbol{x}'-\boldsymbol{x}||^3}dS(\boldsymbol{x}') + \boldsymbol{v}_\phi. \tag{3.11}$$

Consider a solid body with boundary surface S in a high Reynolds number flow. In this case, vorticity is confined to a thin region around S referred to as the boundary layer [69, 152]. In the limit as Reynolds number tends to infinity (the Euler limit), the boundary layer collapses to an infinitely thin shear layer along S, which is equivalent to a vortex sheet [153, page 194] with strength $\boldsymbol{\gamma}(\boldsymbol{x}),\ \boldsymbol{x} \in S$. The velocity jumps from $\boldsymbol{v}^-$ at the surface to $\boldsymbol{v}^+$ immediately outside, and it follows directly from the main assumption that $\boldsymbol{v}^+$ is irrotational. Thus, our goal is to devise a set of equations in terms of $\boldsymbol{\gamma}$ such that the solution provides a valid vortex sheet bound to the surface S.

For non-lifting flow, there are two requirements for a valid bound vortex sheet. The first requirement comes from the no-slip condition, $\boldsymbol{v}^- = \boldsymbol{0}$ (relative to the body), which persists in the Euler limit. If S is closed and simply connected, it will suffice to enforce this only in the normal direction. Then, we require $\boldsymbol{n} \cdot \boldsymbol{v} = 0$, where the "sheet" designation is not needed because $\boldsymbol{n} \cdot \boldsymbol{v}^- = \boldsymbol{n} \cdot \boldsymbol{v}_{\text{sheet}} = \boldsymbol{n} \cdot \boldsymbol{v}^+$.

The second requirement for a valid bound vortex sheet is that the velocity resulting from the surface vorticity is irrotational. From a vector calculus identity, on S we have

$$\begin{aligned}
\nabla \cdot \boldsymbol{\gamma} &= \nabla \cdot (\boldsymbol{n} \times \boldsymbol{v}^+) \\
&= \boldsymbol{v}^+ \cdot (\nabla \times \boldsymbol{n}) - \boldsymbol{n} \cdot (\nabla \times \boldsymbol{v}^+) \\
&= \boldsymbol{v}^+ \cdot \boldsymbol{0} - \boldsymbol{n} \cdot \boldsymbol{0} \\
&= \boldsymbol{0}.
\end{aligned}$$

Thus, for velocity to be irrotational, surface vorticity must be solenoidal.

The problem formulation to determine the incompressible, infinite Reynolds number flow around a solid body with closed boundary surface S moving with velocity $\boldsymbol{v}_S$ and immersed in a uniform freestream $\boldsymbol{v}_\infty$ can now be stated:

$$\text{Determine } \boldsymbol{\gamma} \text{ on } S \text{ such that} \tag{3.12}$$

$$\boldsymbol{v}(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) = \boldsymbol{v}_S(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}),\ \boldsymbol{x} \in S \tag{3.13}$$

$$\nabla \cdot \boldsymbol{\gamma}(\boldsymbol{x}) = 0,\ \boldsymbol{x} \in S \tag{3.14}$$

$$\boldsymbol{v}(\boldsymbol{x}) = \boldsymbol{v}_\infty + \frac{1}{4\pi} \int_S \frac{(\boldsymbol{x}' - \boldsymbol{x}) \times \boldsymbol{\gamma}(\boldsymbol{x}')}{||\boldsymbol{x}' - \boldsymbol{x}||^3} dS(\boldsymbol{x}') \tag{3.15}$$

This is the continuous *boundary integral form* of the problem formulation, and the corresponding discrete form, which is the actual numerical problem that is solved, is presented subsequently.

3.1.2 Unsteady Pressures

The problem formulation does not invoke Equation 3.1, the Euler momentum equation, which is consistent with traditional potential flow problem formulations based on Laplace's equation [136, page 6],[14, page 33]. Instead, we use Equation 3.1 in the form of the Bernoulli equation to determine pressure from velocity. The following derivation of the Bernoulli equation for unsteady flow is summarized from the presentation by Katz and Plotkin [14, page 28]. Equation 3.1 is reformulated to include vorticity as

$$\frac{\partial \boldsymbol{v}}{\partial t} - \boldsymbol{v} \times \boldsymbol{\omega} + \tfrac{1}{2}\nabla(\boldsymbol{v} \cdot \boldsymbol{v}) = -\frac{1}{\rho}\nabla p. \tag{3.16}$$

This equation will be applied on the body immediately outside the boundary layer, where $\boldsymbol{v}$ is assumed to be irrotational, which implies $\boldsymbol{v} \times \boldsymbol{\omega} = \boldsymbol{0}$; the $\boldsymbol{v}^+$ notation is dropped for convenience. The time derivative of velocity is converted to a time derivative of a scalar potential function Φ defined by

$$\Phi := \int_{P_0}^{P} \boldsymbol{v} \cdot d\boldsymbol{l} \tag{3.17}$$

with an arbitrary reference point P_0. This line integral is path independent because $\boldsymbol{v}$ is a conservative vector field. Substituting

$$\frac{\partial \boldsymbol{v}}{\partial t} = \nabla \frac{\partial \Phi}{\partial t} \tag{3.18}$$

and factoring out the gradient operator from Equation 3.16 produces

$$\nabla \left(\frac{p}{\rho} + \tfrac{1}{2} \boldsymbol{v} \cdot \boldsymbol{v} + \frac{\partial \Phi}{\partial t} \right) = \boldsymbol{0}. \tag{3.19}$$

This implies the scalar function inside the parentheses varies only with time, giving

$$\frac{p}{\rho} + \tfrac{1}{2} \boldsymbol{v} \cdot \boldsymbol{v} + \frac{\partial \Phi}{\partial t} = C(t). \tag{3.20}$$

Finally, the pressure coefficient may be formulated in reference to freestream pressure and velocity as

$$C_p := \frac{p - p_\infty}{\frac{1}{2} \rho v_\infty^2} = 1 - \frac{\boldsymbol{v} \cdot \boldsymbol{v}}{v_\infty^2} - \frac{2}{v_\infty^2} \frac{\partial \Phi}{\partial t}. \tag{3.21}$$

While the solution to the problem formulation provides surface vorticity, velocity can be recovered from surface vorticity by the vector triple product

$$\begin{aligned} \boldsymbol{\gamma} \times \boldsymbol{n} &= (\boldsymbol{n} \times \boldsymbol{v}) \times \boldsymbol{n} \\ &= -(\boldsymbol{n} \cdot \boldsymbol{v}) \boldsymbol{n} + (\boldsymbol{n} \cdot \boldsymbol{n}) \boldsymbol{v} \\ &= \boldsymbol{v}. \end{aligned}$$

This result demonstrates one of the incentives for formulating the flow solution in terms of surface vorticity: velocity on the surface is immediately obtained from the solution and to the same order of accuracy.

3.1.3 Lifting Flow and the Unsteady Kutta Condition

Lifting flow is characterized by the inclusion of a wake emitted from a separation path on the body surface. Following the logic presented previously in Section 3.1.1, the wake is concentrated into an infinitely thin shear layer and is also represented by a vortex sheet. The body surface, renamed as S_b, is topologically split along the separation path (e.g., the trailing edge) so as to admit two distinct values of surface vorticity, $\boldsymbol{\gamma}_u$ and $\boldsymbol{\gamma}_l$, at any point along the path.

This topological description is illustrated in Figure 3.1, where we note that the wing is not physically open at the trailing edge. In the geometric description the black line represents the separation path, and in the topological description the black lines represent the shared boundaries of S_b and S_w.

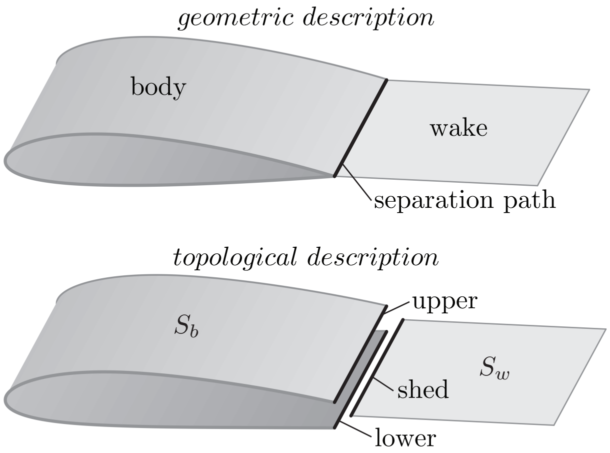


Figure 3.1: Cross section of a wing illustrating the interpretation of the trailing edge.

The introduction of a separation path produces a boundary for the bound vortex sheet, which requires the inclusion of a boundary condition in the problem formulation. The *Kutta* *condition* is imposed when the separation path is a sharp (i.e., wedge-shaped) trailing edge. The Kutta condition stipulates that the flow leaves the trailing edge smoothly with a finite velocity [154],[69, page 436],[14, page 103].

Hess describes three distinct applications of the Kutta condition in a 3-D boundary element method, two of which are relevant to our surface vorticity method [155, page 300]. Application (a): "a stream surface of the flow leaves the trailing edge with a direction that is known, or at least can be approximated." This requires a no-normal-flow constraint at a specified point aft of the trailing edge, typically along the trailing-edge bisector. The selection of this distance aft requires special attention. Application (b): "as the trailing edge is approached, the surface pressures (velocity magnitudes) on the upper and lower surfaces have a common limit." This requirement on surface pressure is applied at the control points nearest the trailing edge and is thus subject to the paneling resolution near the trailing edge. Hess concludes from a numerical experiment that the pressure equality condition (b) is more accurate and robust than the flow tangency condition (a). However, condition (a) produces linear equations while condition (b) produces quadratic equations.

In this thesis, we implement the pressure equality condition because it will not add significant computational expense. At a point P on the trailing edge, consider point P_u assigned to the upper surface and point P_l assigned to the lower surface, both collocated with P. The unsteady pressure difference between P_u and P_l is

$$\begin{aligned} C_p(P_u) - C_p(P_l) &= \left[1 - \frac{\boldsymbol{v}(P_u)\cdot\boldsymbol{v}(P_u)}{v_\infty^2} - \frac{2}{v_\infty^2}\frac{\partial\Phi(P_u)}{\partial t}\right] \\ &\quad - \left[1 - \frac{\boldsymbol{v}(P_l)\cdot\boldsymbol{v}(P_l)}{v_\infty^2} - \frac{2}{v_\infty^2}\frac{\partial\Phi(P_l)}{\partial t}\right] \\ &= \frac{\boldsymbol{v}(P_l)\cdot\boldsymbol{v}(P_l) - \boldsymbol{v}(P_u)\cdot\boldsymbol{v}(P_u)}{v_\infty^2} \\ &\quad - \frac{2}{v_\infty^2}\frac{\partial}{\partial t}\left[\Phi(P_u) - \Phi(P_l)\right]. \end{aligned} \tag{3.22}$$

The difference of potentials can be reformulated as the circulation around the wing. Construct paths C_u and C_l from reference point P_0 to P_u and P_l, respectively, as illustrated in Figure 3.2. The path integrals for potential combine to give the closed path integral for circulation as

$$\begin{aligned} \Phi(P_u) - \Phi(P_l) &= \int_{C_u} \boldsymbol{v}\cdot d\boldsymbol{l} - \int_{C_l} \boldsymbol{v}\cdot d\boldsymbol{l} \\ &= \oint_{C_{l\to u}} \boldsymbol{v}\cdot d\boldsymbol{l} \\ &= \Gamma(P). \end{aligned} \tag{3.23}$$

In this example, paths C_u and C_l were purposefully constructed to combine to produce

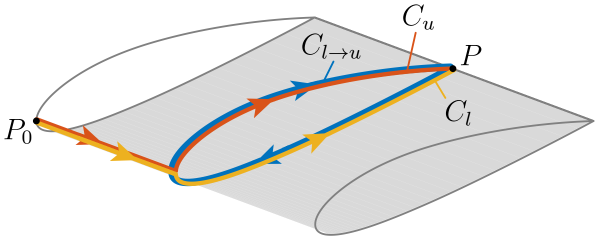


Figure 3.2: Cross section of a wing illustrating paths along the surface to a point on the trailing edge.

path $C_{l\to u}$ for which Γ is intuitive, but any arbitrary paths would suffice because $\boldsymbol{v}$ is a conservative vector field on the body surface.

Therefore, the unsteady Kutta condition requiring pressure equality at the trailing edge simplifies to

$$0 = C_p(P_u) - C_p(P_l) = \frac{\boldsymbol{v}(P_l)\cdot\boldsymbol{v}(P_l) - \boldsymbol{v}(P_u)\cdot\boldsymbol{v}(P_u)}{v_\infty^2} - \frac{2}{v_\infty^2}\frac{\partial}{\partial t}\Gamma(P). \tag{3.24}$$

The application of the unsteady Kutta condition to the problem formulation is discussed in Chapter 4, in which the bound vortex sheets and free vortex sheets are combined.

3.2 Triangular Panel Representation

To discretize the continuous problem formulation into a numerical problem formulation that can be solved by a computer, the solid body surface S_b is triangulated into surface $S_{b\triangle}$ composed of flat triangular panels. These panels are then used for the subsequent calculations of influence velocities, circulation, and surface pressure. For a triangle comprising vertices (P_1, P_2, P_3) with vertex locations $\boldsymbol{p}_1$, $\boldsymbol{p}_2$, and $\boldsymbol{p}_3$, the edge vectors are defined as

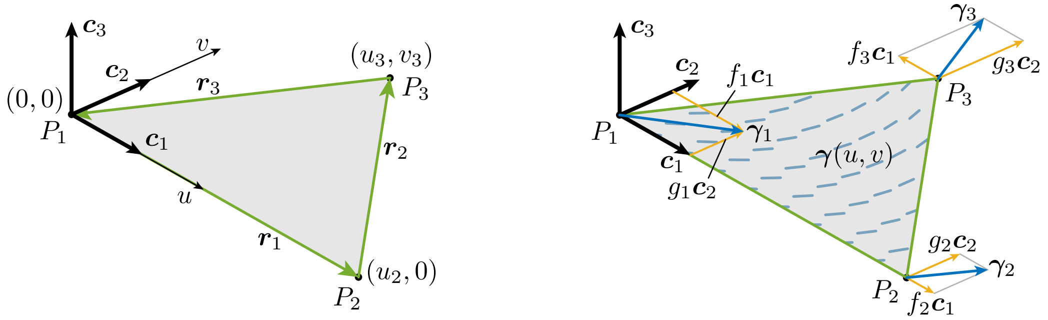
$$\boldsymbol{r}_1 := \boldsymbol{p}_2 - \boldsymbol{p}_1, \quad \boldsymbol{r}_2 := \boldsymbol{p}_3 - \boldsymbol{p}_2, \quad \boldsymbol{r}_3 := \boldsymbol{p}_1 - \boldsymbol{p}_3. \tag{3.25}$$

Each triangle is assigned a basis $(\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3)$, with

$$\boldsymbol{c}_1 := \frac{\boldsymbol{r}_1}{||\boldsymbol{r}_1||}, \quad \boldsymbol{c}_2 := \boldsymbol{c}_3 \times \boldsymbol{c}_1, \quad \boldsymbol{c}_3 := \frac{\boldsymbol{r}_1 \times \boldsymbol{r}_2}{||\boldsymbol{r}_1 \times \boldsymbol{r}_2||}, \tag{3.26}$$

in which case $\boldsymbol{c}_3$ is the triangle normal vector. This notation is illustrated in Figure 3.3a.

A local coordinate system may then be established on the triangle with coordinate u along $\boldsymbol{c}_1$, coordinate v along $\boldsymbol{c}_2$, and the origin at $\boldsymbol{p}_1$. Next, as illustrated in Figure 3.3b, the surface vorticity $\boldsymbol{\gamma}$ is confined to vary linearly along the triangle with two scalar functions



(a) Panel edges and coordinate system

(b) Panel surface vorticity

Figure 3.3: Notation for the triangular panels.

$f, g : \mathbb{R}^2 \to \mathbb{R}$ as

$$\boldsymbol{\gamma}(u, v) = f(u, v)\boldsymbol{c}_1 + g(u, v)\boldsymbol{c}_2, \tag{3.27}$$

$$f(u, v) = a_{00} + a_{10}u + a_{01}v, \tag{3.28}$$

$$g(u, v) = b_{00} + b_{10}u + b_{01}v. \tag{3.29}$$

These coefficients are uniquely determined by the surface vorticity at each of the three vertices $\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \boldsymbol{\gamma}_3$ according to the linear system

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & u_2 & 0 \\ 1 & u_3 & v_3 \end{bmatrix} \begin{bmatrix} a_{00} & b_{00} \\ a_{10} & b_{10} \\ a_{01} & b_{01} \end{bmatrix} = \begin{bmatrix} f_1 & g_1 \\ f_2 & g_2 \\ f_3 & g_3 \end{bmatrix}, \tag{3.30}$$

where these terms are given by

$$\begin{aligned} u_2 &= \boldsymbol{r}_1 \cdot \boldsymbol{c}_1 & f_1 &= \boldsymbol{\gamma}_1 \cdot \boldsymbol{c}_1 & g_1 &= \boldsymbol{\gamma}_1 \cdot \boldsymbol{c}_2 \\ u_3 &= -\boldsymbol{r}_3 \cdot \boldsymbol{c}_1 & f_2 &= \boldsymbol{\gamma}_2 \cdot \boldsymbol{c}_1 & g_2 &= \boldsymbol{\gamma}_2 \cdot \boldsymbol{c}_2 \\ v_3 &= -\boldsymbol{r}_3 \cdot \boldsymbol{c}_2 & f_3 &= \boldsymbol{\gamma}_3 \cdot \boldsymbol{c}_1 & g_3 &= \boldsymbol{\gamma}_3 \cdot \boldsymbol{c}_2, \end{aligned} \tag{3.31}$$

as illustrated in figures 3.3a and 3.3b. The solution to Equation 3.30 yields

$$\begin{aligned} a_{00} &= f_1 & b_{00} &= g_1 \\ a_{10} &= \frac{f_2 - f_1}{u_2} & b_{10} &= \frac{g_2 - g_1}{u_2} \\ a_{01} &= \frac{u_3 - u_2}{u_2 v_3} f_1 - \frac{u_3}{u_2 v_3} f_2 + \frac{1}{v_3} f_3 & b_{01} &= \frac{u_3 - u_2}{u_2 v_3} g_1 - \frac{u_3}{u_2 v_3} g_2 + \frac{1}{v_3} g_3. \end{aligned} \tag{3.32}$$

Next, each vertex is assigned a normal vector $\boldsymbol{d}_3$, which may be calculated as the average of the normal vectors from neighboring triangles, or supplied as exact from analytical calculations on S_b. A basis $(\boldsymbol{d}_1, \boldsymbol{d}_2, \boldsymbol{d}_3)$ may then be formed. As stated previously, the surface vorticity on a triangle is determined by the strength at its vertices $\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \boldsymbol{\gamma}_3$. However, when a vertex normal vector $\boldsymbol{d}_3$ is not parallel to normal vector $\boldsymbol{c}_3$ of a neighboring triangle, these strengths will likely not be in the plane of the triangle, as demonstrated in Figure 3.4. Therefore, we must decide on a transformation to convert vector $\boldsymbol{\gamma}_d$ in the $(\boldsymbol{d}_1, \boldsymbol{d}_2)$ plane to a vector $\boldsymbol{\gamma}_c$ in the $(\boldsymbol{c}_1, \boldsymbol{c}_2)$ plane. The vector projection

$$\boldsymbol{\gamma}_c = \boldsymbol{\gamma}_d - (\boldsymbol{\gamma}_d \cdot \boldsymbol{c}_3)\boldsymbol{c}_3 \tag{3.33}$$

is an obvious choice, which, unfortunately, leads to inaccurate calculations because the projection does not preserve the magnitude of $\boldsymbol{\gamma}_d$. Alternatively, a vector rotation does preserve the magnitude of γ_d; rotating a vector around an edge is analogous to the parallel transport of a vector along a smooth curve.

To rotate a vector $\boldsymbol{\gamma}_d$ in the $(\boldsymbol{d}_1, \boldsymbol{d}_2)$ plane onto the $(\boldsymbol{c}_1, \boldsymbol{c}_2)$ plane, construct $\boldsymbol{q} := (\boldsymbol{d}_3 \times \boldsymbol{c}_3)/||\boldsymbol{d}_3 \times \boldsymbol{c}_3||$ and $\theta := \sin^{-1} ||\boldsymbol{d}_3 \times \boldsymbol{c}_3||$, as depicted in Figure 3.4. Then, $\boldsymbol{\gamma}_c$ is obtained

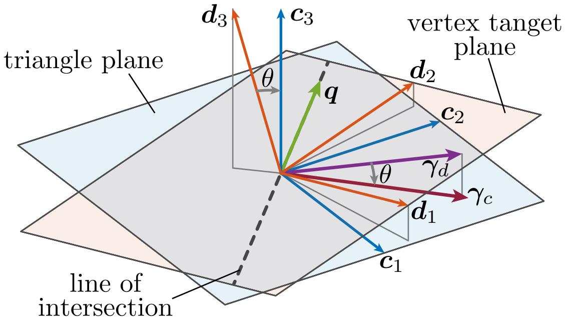


Figure 3.4: The vector γ_d in the vertex plane with normal vector $\boldsymbol{d}_3$ must be related to some vector γ_c in the triangle plane with normal vector $\boldsymbol{c}_3$. All vectors outside of the $(\boldsymbol{c}_1, \boldsymbol{c}_2)$ plane are accompanied by thin gray lines to indicate their height (except $\boldsymbol{c}_3$).

by rotating $\boldsymbol{\gamma}_d$ about $\boldsymbol{q}$ through the angle θ via Rodrigues' rotation formula

$$\boldsymbol{\gamma}_c = \boldsymbol{\gamma}_d \cos\theta + (\boldsymbol{q} \times \boldsymbol{\gamma}_d)\sin\theta + \boldsymbol{q}(\boldsymbol{q} \cdot \boldsymbol{\gamma}_d)(1 - \cos\theta), \tag{3.34}$$

which is linear in $\boldsymbol{\gamma}_d$.

3.2.1 Panel Perimeter Circulation Constraint

The linear representation of surface vorticity expressed in Equation 3.27 provides a straightforward expression to discretize the zero-divergence constraint

$$\begin{aligned}\nabla \cdot \boldsymbol{\gamma} &= \frac{\partial f(u,v)}{\partial u} + \frac{\partial g(u,v)}{\partial v} \\ &= a_{10} + b_{01} \end{aligned} \tag{3.35}$$

However, we found that better solutions are produced by directly requiring the surface velocity to be irrotational, as discussed in Section 3.1.1. Because $\boldsymbol{v}$ is irrotational, it must be conservative on any simply connected subset of the domain, i.e. the circulation must be zero:

$$\Gamma = \oint \boldsymbol{v} \cdot d\boldsymbol{l} = 0. \tag{3.36}$$

The perimeter of each panel provides a convenient set of loops to implement this constraint, which we call the panel perimeter circulation constraint.

3.3 Numerical Problem Formulation

The surface triangulation $S_{b\triangle}$ comprises N_V vertices and N_T triangles; a typical triangulation will have $N_T \approx 2N_V$. Each vertex possesses two degrees of freedom defining the local surface vorticity, and the resulting total number of degrees of freedom is $N_{\text{dof}} = 2N_V$. The surface vorticity at the ith vertex is expressed via the two scalars f_i and g_i and the local basis vectors $\boldsymbol{d}_{1,i}$ and $\boldsymbol{d}_{2,i}$ as $\boldsymbol{\gamma}_i = f_i\boldsymbol{d}_{1,i} + g_i\boldsymbol{d}_{2,i}$. The scalars f_i and g_i, $i = 1, \ldots, N_V$ are arranged into column matrices $\vec{f}$ and $\vec{g}$ and are then concatenated as

$$\vec{x} := \begin{bmatrix} \vec{f} \\ \\ \vec{g} \end{bmatrix} \tag{3.37}$$

to form one complete list of unknowns. The notation $\vec{\cdot}$ indicates a column matrix (or column vector) and is not to be confused with physical vectors in $\mathbb{R}^3$ indicated by bold characters.

Both divergence and induced velocity are calculated on a triangular panel in terms of the surface vorticity at its three vertices, which are given by $f_1, f_2, f_3, g_1, g_2, g_3$, as indicated in Figure 3.3b. Therefore, we need access to $\vec{f_1}, \vec{f_2}$, etc., which requires the introduction of matrices $A_{f_1 \leftarrow x}, A_{f_2 \leftarrow x}$, etc., the elements of which are populated by applying Equation 3.34 to the basis vectors of every pair of associated vertices and triangles. For example,

$$\vec{f_1} = A_{f_1 \leftarrow x}\vec{x}, \tag{3.38}$$

which means $A_{f_1 \leftarrow x}$ is a sparse matrix of size $N_T \times N_{\text{dof}}$; together, $\vec{f_1}$ and $\vec{g}_1$ are the components of the surface vorticity at the first vertex of every triangle.

3.3.1 Normal Flow Constraint in Matrix Form

The normal flow constraint (Equation 3.13) is applied at the centroid of each triangle $\boldsymbol{p}_c = (\boldsymbol{p}_1 + \boldsymbol{p}_2 + \boldsymbol{p}_3)/3$, and the velocity normal to the surface at the centroid of the ith triangle is expressed as $v_{c,i} = \boldsymbol{v}_{c,i} \cdot \boldsymbol{c}_{3,i}$. The change in induced velocity at $\boldsymbol{p}_c$ on the ith triangle due to a change in f_1 on the jth triangle is expressed as $\partial \boldsymbol{v}_{c,i} / \partial f_{1,j}$ and is calculated from the solution to Equation 3.15 presented in Appendix A. These partial derivatives are arranged in a Jacobian matrix

$$\frac{\partial \vec{v}_c}{\partial \vec{f_1}} = \begin{bmatrix} \dfrac{\partial \boldsymbol{v}_{c,1}}{\partial f_{1,1}} \cdot \boldsymbol{c}_{3,1} & \cdots & \dfrac{\partial \boldsymbol{v}_{c,1}}{\partial f_{1,N_T}} \cdot \boldsymbol{c}_{3,1} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial \boldsymbol{v}_{c,N_T}}{\partial f_{1,1}} \cdot \boldsymbol{c}_{3,N_T} & \cdots & \dfrac{\partial \boldsymbol{v}_{c,N_T}}{\partial f_{1,N_T}} \cdot \boldsymbol{c}_{3,N_T} \end{bmatrix}, \tag{3.39}$$

and likewise for $\partial \vec{v}_c / \partial \vec{f_2}$, $\partial \vec{v}_c / \partial \vec{f_3}$, $\partial \vec{v}_c / \partial \vec{g}_1$, etc. Then, the matrix A_{nf} is computed as

$$A_{\text{nf}} = \frac{\partial \vec{v}_c}{\partial \vec{f_1}} A_{f_1 \leftarrow x} + \frac{\partial \vec{v}_c}{\partial \vec{f_2}} A_{f_2 \leftarrow x} + \cdots + \frac{\partial \vec{v}_c}{\partial \vec{g}_3} A_{g_3 \leftarrow x} \tag{3.40}$$

and the corresponding vector $\vec{b}_{\text{nf}}$ is

$$\vec{b}_{\text{nf}} = \begin{bmatrix} (\boldsymbol{v}_{S,1} - \boldsymbol{v}_\infty) \cdot \boldsymbol{c}_{3,1} \\ \vdots \\ (\boldsymbol{v}_{S,N_T} - \boldsymbol{v}_\infty) \cdot \boldsymbol{c}_{3,N_T} \end{bmatrix}, \tag{3.41}$$

where $\boldsymbol{v}_{S,i}$ is the velocity of the body at the centroid of the ith triangle.

3.3.2 Panel Perimeter Circulation Constraint in Matrix Form

The constraint for zero panel perimeter circulation has the form

$$A_{\text{ppc}}\vec{x} = \vec{b}_{\text{ppc}} = \vec{0}. \tag{3.42}$$

To construct matrix A_{ppc}, consider a set of directed edges comprising the triangulation. Each triangle is an ordered assembly of three edges, either with matching or reversed orientation. This allows us to construct the triangle–edge adjacency matrix A_{TE}, where element i, j is unity if triangle i comprises edge j, negative unity in the case of reverse orientation, and zero otherwise.

Next, construct a matrix of edge velocity integrals A_{evi} in which element i, j is the contribution from the jth degree of freedom to the velocity integral along the ith edge. Then, the perimeter panel circulation matrix is constructed as

$$A_{\text{ppc}} = A_{TE}A_{\text{evi}}. \tag{3.43}$$

3.3.3 Numerical Non-Lifting Problem Formulation

With the degrees of freedom arranged in column vector $\vec{x}$ as defined in Equation 3.37, the numerical *boundary element* form of the problem formulation presented in Section 3.1.1 is

$$\text{Determine } \vec{x} \text{ such that} \tag{3.44}$$

$$A_{\text{nf}}\vec{x} = \vec{b}_{\text{nf}} \tag{3.45}$$

$$A_{\text{ppc}}\vec{x} = \vec{b}_{\text{ppc}}. \tag{3.46}$$

These two constraints are concatenated as

$$A := \begin{bmatrix} A_{\text{nf}} \\ A_{\text{ppc}} \end{bmatrix}, \quad \vec{b} := \begin{bmatrix} \vec{b}_{\text{nf}} \\ \vec{b}_{\text{ppc}} \end{bmatrix}. \tag{3.47}$$

The constraint matrices A_{nf} and A_{ppc} both have have size $N_T \times N_{\text{dof}}$, and so A has approximate size $2N_{\text{dof}} \times N_{\text{dof}}$. This produces an overdetermined system that cannot be solved exactly. Our solution strategy is to minimize the sum of the squared residuals

$$\begin{aligned} \min_{\vec{x}} \quad & (A\vec{x} - \vec{b})^T W (A\vec{x} - \vec{b}) \qquad (3.48) \\ \text{subject to} \quad & A_{\text{eq}}\vec{x} = \vec{b}_{\text{eq}} \end{aligned}$$

where W is a weighting matrix and A_{eq} and $\vec{b}_{\text{eq}}$ are optional equality constraints that will be introduced subsequently for lifting flow. A perfectly determined system is often preferable to an overdetermined one, and a square matrix could be obtained by omitting certain residuals. But there is no guarantee that the remaining residuals will be linearly independent or that the resulting system will be well-conditioned.

Because the panel perimeter circulation is an integral expression, the residuals in Equation 3.46 are intrinsically scaled by the panel areas, which are typically small. To balance the scale of these residuals with the normal flow residuals, Equation 3.46 is scaled by the reciprocal of the mean panel area.

Equation 3.48 is a quadratic program with linear equality constraints, which may be solved with the following linear system [156]

$$\begin{bmatrix} G & A_{\text{eq}}^T \\ A_{\text{eq}} & 0 \end{bmatrix} \begin{bmatrix} \vec{x}^* \\ -\vec{\lambda}^* \end{bmatrix} = \begin{bmatrix} -\vec{d} \\ \vec{b}_{\text{eq}} \end{bmatrix}, \tag{3.49}$$

where

$$G := 2(A^T W A), \quad \vec{d} := -2A^T(W\vec{b}). \tag{3.50}$$

The matrix on the left-hand side of Equation 3.49 is Hermitian and dense.

To further control the relative importance of the sources of residuals, the weighting matrix W in Equation 3.48 is expressed simply as scalar weighting factor $w \in (0, 1)$. The quadratic and linear terms of the objective function are computed as

$$G = 2[wA_{\text{nf}}^T A_{\text{nf}} + (1-w)A_{\text{ppc}}^T A_{\text{ppc}}] \tag{3.51}$$

$$\vec{d} = -2[wA_{\text{nf}}^T \vec{b}_{\text{nf}} + (1-w)A_{\text{ppc}}^T \vec{b}_{\text{ppc}}]. \tag{3.52}$$

The relative weighting factor, w is further explored in the next section.

3.4 Non-Lifting Comparison to Analytical Solution for Ellipsoid

To assess the errors associated with the least-squares formulation, consider a tri-axial ellipsoid with semi-major axis lengths $a = 3$, $b = 2$, $c = 1$, corresponding to the body-fixed basis $(\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3)$ with Cartesian coordinates (x, y, z). The triangulations are formed from a regular convex icosahedron by recursively dividing each triangle into four new triangles and projecting the new vertices on the circumscribed sphere. This results in a triangulated sphere which is then stretched along $\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3$ by a, b, c. The number of triangles resulting from N_4 recursive divisions is

$$N_T = 20 \cdot 4^{N_4}. \tag{3.53}$$

Two metrics are considered to compare the numerical approximation to the analytical solution (i.e., exact solution) for the potential flow around an ellipsoid immersed in a uniform freestream that is presented in Appendix C.

The first metric is the relative error of the aerodynamic pitching moment. For a freestream velocity of $\boldsymbol{v}_\infty = \boldsymbol{b}_2 + \frac{1}{8}\boldsymbol{b}_3$, the resulting net force is zero but the nonzero angle of attack creates a pitching moment about $\boldsymbol{b}_1$ as

$$\boldsymbol{b}_1 \cdot \int_{S_{b\triangle}} -\boldsymbol{x} \times \boldsymbol{n}(\boldsymbol{x}) C_p(\boldsymbol{x}) dA. \tag{3.54}$$

The analytic solution is sampled at the vertices of $S_{b\triangle}$ and then used in the same manner as the numerical solution to determine C_p; that is, it defines the linear variation of surface vorticity along each panel.

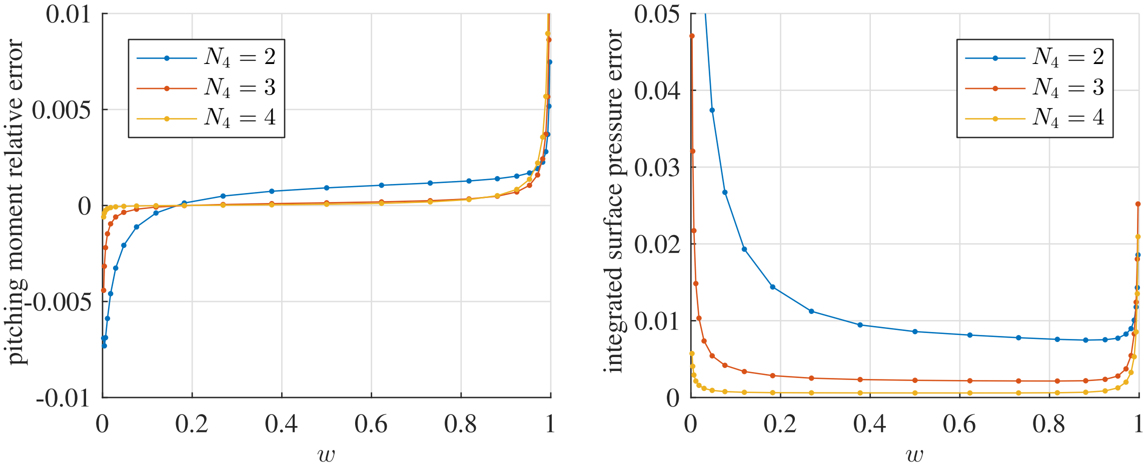
The second metric focuses on the local difference in surface pressure by integrating the error over the surface of the ellipsoid:

$$\int_{S_{b\triangle}} \big|C_p(\boldsymbol{x}) - C_{p,0}(\boldsymbol{x})\big| dA. \tag{3.55}$$

This metric captures noise in the solution that could be hidden in an otherwise accurate pitching moment prediction.

The errors for a full range of values for the relative weighting factor w and triangulation resolution N_4 are depicted in Figures 3.5a and 3.5b. Note that resolutions of $N_4 = 2, 3, 4$ correspond to triangulations with $N_T = 320, 1280, 5120$. These figures reveal two promising results: the errors converge to zero as the resolution increases, and they are insensitive to a broad range of w—from $w = 0.2$ to $w = 0.9$.

To further study the convergence behavior of the numerical method, Figure 3.6 presents several error metrics as the triangulation resolution is refined: first (blue), the relative error of the pitching moment calculated from sampling the analytic solution at the vertices as



(a) Pitching moment relative error.

(b) Integrated surface pressure error.

Figure 3.5: Aggregate metrics indicating the error associated with the relative weighting w for ellipsoids of varying resolution.

compared to the exact pitching moment; second (red), the relative error of the pitching moment calculated from the numerical prediction; third (yellow), the relative error of the volume of the triangulated surface compared to the exact ellipsoid; lastly (purple), the integrated error expressed in Equation 3.55. The horizontal axis variable in Figure 3.6 is the mean edge length of each triangulation, and, therefore, represents the length scale of the surface mesh. A gray line is included to indicate a second order rate of convergence, which suggests we can conclude this boundary element method is second-order accurate.

As a demonstration of this numerical method, the flow solution for an ellipsoid with 5120 triangles is illustrated in Figure 3.7. The pathlines of surface vorticity form closed loops, whereas the pathlines of velocity all travel from the forward stagnation point to the aft stagnation point. As expected, the surface vorticity and surface velocity pathlines are mutually perpendicular. To inspect how closely the normal flow and panel perimeter circulation constraints were satisfied for this solution, histograms of the residuals from Equations 3.45 and 3.46 are presented in Figures 3.8a and 3.8b, respectively. These residuals

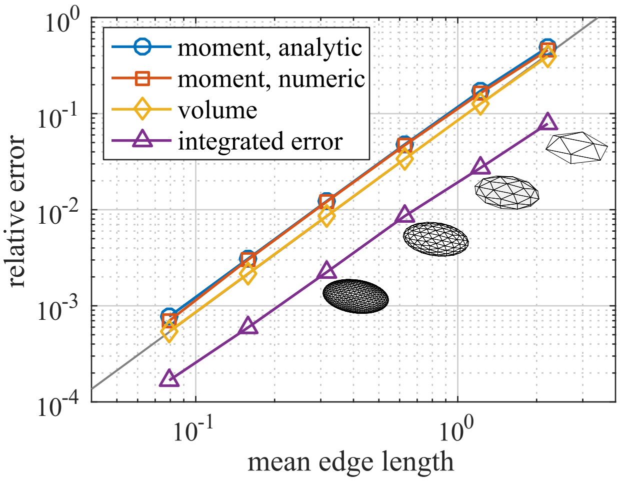


Figure 3.6: Convergence of different calculations with the proposed numerical method.

closely match those of the analytic solution sampled at the vertices.

Also, as a further demonstration with this predicted flow solution, Equation 3.11 is applied to calculate streamsurfaces in the external flow around the ellipsoid, as depicted in Figure 3.9. The blue curves and the yellow curves are streamlines, and the red curves and purple curves indicate fluid particle positions at constant time intervals. The blue/red streamsurface remains in the vertical plane of symmetry and demonstrates that the streamlines passing near the stagnation points lag behind their neighboring streamlines that do not. In contrast, the yellow/purple streamsurface travels around the tip without slowing down, but it is sheared such that the upper portion is displaced inward.

The accurate prediction of streamlines near the surface is the main reason for using panels with linearly varying surface vorticity. Our method is well-suited as an inviscid solver to couple with an interactive boundary layer model to better simulate high Reynolds number flow.

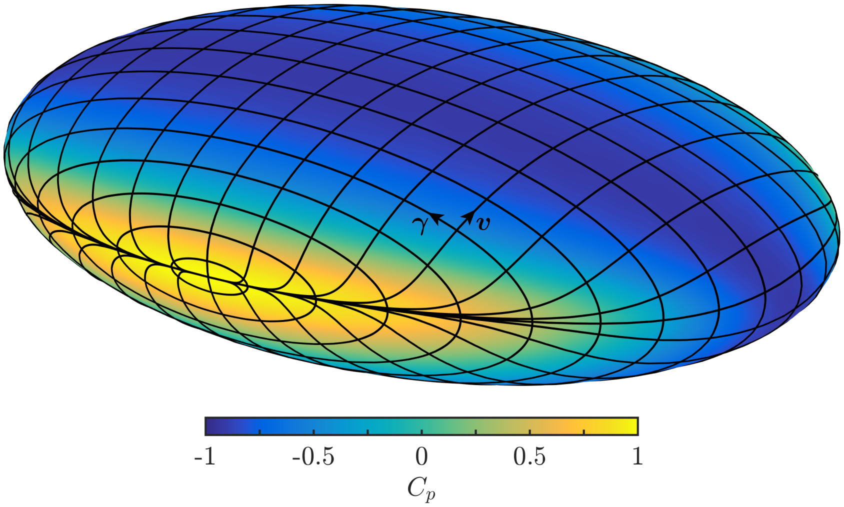
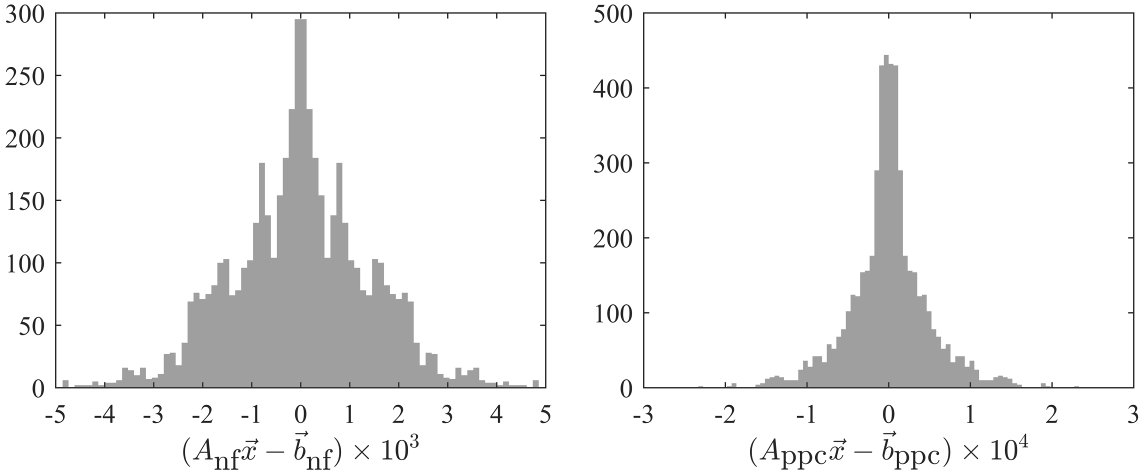


Figure 3.7: Pathlines of surface vorticity, pathlines of surface velocity, and surface pressure coefficient for the numerical flow solution.



(a) Normal flow residuals.

(b) Panel perimeter circulation residuals.

Figure 3.8: Histograms of the residuals for the numerical calculation of surface vorticity on the ellipsoid with 5120 triangles.

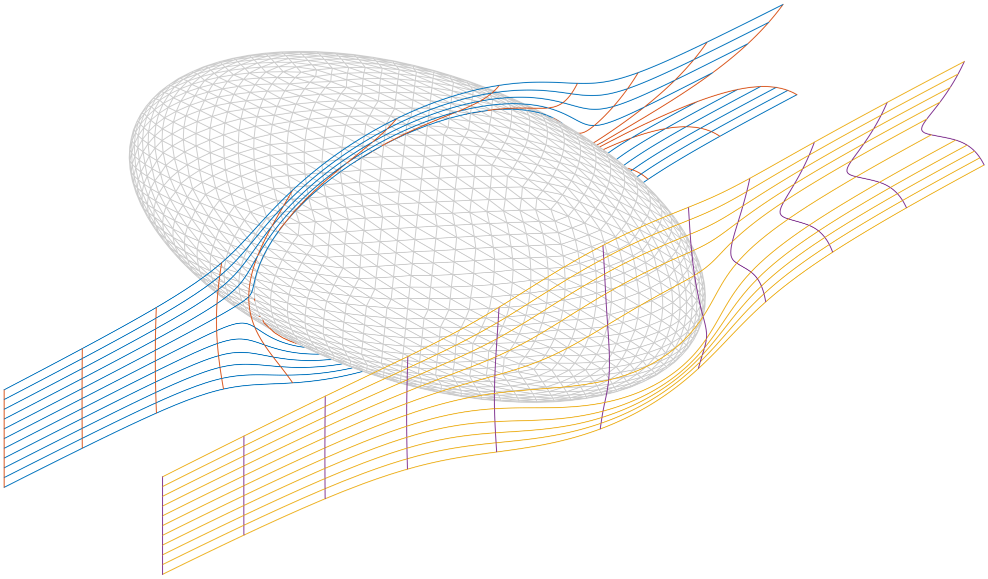


Figure 3.9: Streamsurfaces around an ellipsoid calculated from the numerical solution.

CHAPTER 4

COMBINED BOUND AND FREE VORTEX SHEETS

The previous two chapters detailed the equations and algorithms for modeling free vortex sheets and bound vortex sheets independently. Of course, bound and free vortex sheets abide by the same physical laws—their separate Eulerian and Lagrangian specifications are for convenience.

As a demonstration, consider Figure 4.1, which depicts a sheet of fluid adjacent to a body before and after it is set in motion. In the absence of viscous mixing, this sheet continues to encase the body as it stretches downstream, away from the sharp trailing edge. Thus, the bound and free vortex sheets we employ to describe this flow are actually two parts of the same sheet. Furthermore, because the wake is assumed to be infinitely thin, the two vortex sheets bounding the wake are actually collocated, and so they can be replaced by a single vortex sheet. In Figure 4.1 this corresponds to replacing the two adjacent red curves in the wake with the single yellow curve emanating out of the trailing edge.

In this manner, the free vortex sheet is imagined to be created at the trailing edge, but it is simply an accounting technique to switch from an Eulerian specification to a Lagrangian specification. This chapter is devoted to combining bound and free vortex sheets to produce a comprehensive method capable of modeling complex aerodynamic flows.

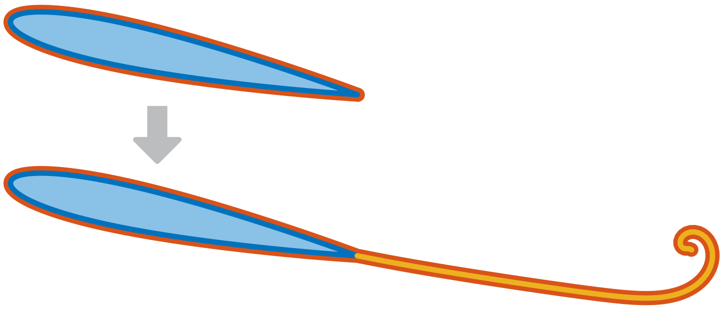


Figure 4.1: A sheet of fluid encasing a body before and after motion.

4.1 Problem Formulation for Unsteady Lifting Flows

In Section 3.1.3, we described the unsteady Kutta condition in continuous form, and we now resume this discussion by presenting the final version of the numerical problem formulation for lifting flow. Solving for the strength of the bound vortex sheet will fortuitously provide the amount of circulation assigned to the free vortex sheet as it is created at the trailing edge.

As previously mentioned, the body surface S_b is topologically split along a prescribed wake separation path, such as a trailing edge. This splitting is implemented in the numerical problem formulation by creating new vertices and edges along the separation path to topologically divide the upper and lower surfaces. The two endpoint vertices of the separation path are not duplicated; this modification to the surface triangulation is illustrated in Figure 4.2. The circulation corresponding to each trailing edge vertex pair is calculated by integrating the velocity along the loops indicated by the purple paths. This example depicts a *structured* triangulation for which the paths comprising these loops can be determined easily. Fortunately, appropriate paths can also be found on an *unstructured* triangulation by applying shortest path algorithms from graph theory [120].

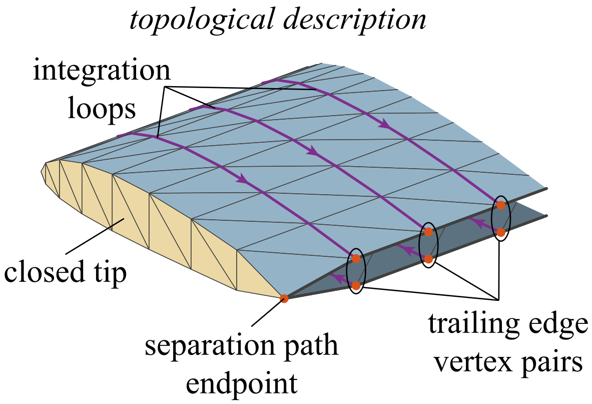


Figure 4.2: The tip section of a sample triangulated wing demonstrating the topological split along the trailing edge. Note that the trailing edge vertex pairs are actually collocated.

Similar to calculating the circulation around each panel perimeter, the circulation around these loops can be calculated by multiplying a sparse matrix A_{loop}, which assigns oriented edges to each integration loop, and the matrix of edge velocity integrals previously introduced in Section 3.3.2, A_{evi}. These circulations are arranged in a column vector and expressed as

$$\vec{\Gamma} = A_\Gamma \vec{x}, \quad A_\Gamma := A_{\text{loop}} A_{\text{evi}}. \tag{4.1}$$

Unfortunately, matching the pressures at the trailing edge vertex pairs requires the solution to Equation 3.24,

$$0 = C_p(P_u) - C_p(P_l) = \frac{\boldsymbol{v}(P_l) \cdot \boldsymbol{v}(P_l) - \boldsymbol{v}(P_u) \cdot \boldsymbol{v}(P_u)}{v_\infty^2} - \frac{2}{v_\infty^2} \frac{\partial}{\partial t} \Gamma(P),$$

which is nonlinear and thus cannot be included in the linear solution to the quadratic program, Equation 3.49. Instead of solving Equation 3.24 directly, we modify the problem formulation and implement a nonlinear iterative optimization procedure that leverages the linear solution to the quadratic program.

In this nonlinear optimization procedure, the degrees of freedom are the scalar components of surface vorticity at each of the trailing edge vertices, which are arranged in column vectors as $\vec{f}_u, \vec{g}_u, \vec{f}_l, \vec{g}_l$ and then concatenated into a single column vector

$$\vec{z} := \begin{bmatrix} \vec{f}_u \\ \vec{g}_u \\ \vec{f}_l \\ \vec{g}_l \end{bmatrix}. \tag{4.2}$$

The number of degrees of freedom in $\vec{z}$ is quadruple the number of trailing edge vertex pairs in the triangulation, and they are enforced in the quadratic program via an additional set of linear equality constraints

$$A_{\text{eq,opt}}\vec{x} = \vec{z}. \tag{4.3}$$

The matrix $A_{\text{eq,opt}}$ is a sparse matrix with one element per row in the appropriate column to provide a surjective mapping from a subset of $\vec{x}$ onto $\vec{z}$.

These new constraints are appended to the old linear equality constraints, which are renamed $A_{\text{eq,0}}$:

$$A_{\text{eq}} = \begin{bmatrix} A_{\text{eq,0}} \\ A_{\text{eq,opt}} \end{bmatrix}, \tag{4.4}$$

thereby allowing $\vec{x}$ and $\vec{z}$ to be related via the solution to the modified quadratic program:

$$\begin{bmatrix} G & A_{\text{eq}}^T \\ A_{\text{eq}} & 0 \end{bmatrix} \begin{bmatrix} \vec{x} \\ -\vec{\lambda} \end{bmatrix} = \begin{bmatrix} -\vec{d} \\ \vec{b}_{\text{eq,0}} \\ \vec{z} \end{bmatrix} = \begin{bmatrix} -\vec{d} \\ \vec{b}_{\text{eq,0}} \\ \vec{0} \end{bmatrix} + A_z\vec{z}, \tag{4.5}$$

where A_z is a sparse matrix that places $\vec{z}$ at the bottom of the right-hand side vector. Then,

$\vec{x}$ can be determined from $\vec{z}$ as

$$\vec{x} = A_x M^{-1}(\vec{y}_0 + A_z\vec{z}), \tag{4.6}$$

where

$$M := \begin{bmatrix} G & A_{\text{eq}}^T \\ A_{\text{eq}} & 0 \end{bmatrix}, \quad \vec{y}_0 := \begin{bmatrix} -\vec{d} \\ \vec{b}_{\text{eq},0} \\ \vec{0} \end{bmatrix}, \tag{4.7}$$

and A_x is a sparse matrix that removes the Lagrange multipliers $\vec{\lambda}$ present after applying M^{-1}.

Therefore, the optimization problem statement to minimize the sum of the squared residuals while enforcing the unsteady pressure matching Kutta condition is:

$$\begin{aligned} \min_{\vec{z}} \quad & F(\vec{x}) \\ \text{subject to} \quad & \Delta\vec{C}_p(\vec{x},\vec{z}) = \vec{0} \\ & \vec{x} = A_x M^{-1}\vec{(}\vec{y}_0 + A_z\vec{z}) \end{aligned} \tag{4.8}$$

with

$$F(\vec{z}) = \tfrac{1}{2}\vec{x}^T G\vec{x} + \vec{x}^T\vec{d} + \vec{b}^T\vec{b} \tag{4.9}$$

$$\Delta\vec{C}_p(\vec{x},\vec{z}) = (\vec{f}_u)^2 + (\vec{g}_u)^2 - (\vec{f}_l)^2 - (\vec{g}_l)^2 + \frac{2}{\Delta t}(\vec{\Gamma}_{\text{new}} - \vec{\Gamma}_{\text{old}}) \tag{4.10}$$

$$\vec{\Gamma}_{\text{new}} = A_\Gamma\vec{x}. \tag{4.11}$$

The vector square notation in Equation 4.10 refers to the Hadamard product, e.g., the ith element of vector $(\vec{f}_u)^2$ is $f_{u,i}^2$.

This modified optimization problem formulation has a quadratic objective function with a set of quadratic equality constraints, and, therefore, it can be solved very efficiently by an optimizer that leverages the first and second derivatives of the objective function and constraints:

$$\nabla F(\vec{z}) = (A_z^T M^{-1T} A_x^T G A_x M^{-1} A_z)\vec{z} + A_z^T M^{-1T} A_x^T(\vec{d} + G A_x M^{-1}\vec{y}_0) \tag{4.12}$$

$$\nabla^2 F(\vec{z}) = A_z^T M^{-1T} A_x^T G A_x M^{-1} A_z \tag{4.13}$$

$$\nabla_{\vec{z}}(\Delta\vec{C}_p(\vec{x},\vec{z})_i) = \begin{bmatrix} 2\text{diag}(\vec{f}_u) \\ 2\text{diag}(\vec{g}_u) \\ -2\text{diag}(\vec{f}_l) \\ -2\text{diag}(\vec{g}_l) \end{bmatrix} + \frac{2}{\Delta t} A_z^T M^{-1T} A_x^T A_\Gamma^T \tag{4.14}$$

$$\sum_i \lambda_i \nabla^2_{\vec{z}\vec{z}}(\Delta\vec{C}_p(\vec{x},\vec{z})) = \text{diag}\left(\begin{bmatrix} 2\vec{\lambda} \\ 2\vec{\lambda} \\ -2\vec{\lambda} \\ -2\vec{\lambda} \end{bmatrix}\right) \tag{4.15}$$

4.2 Wake Shedding Procedure

A new set of wake elements is shed at the wing trailing edge at the end of each time-step, as will be discussed in the following section. The vortex sheet strength of these new elements is determined from the solution to Equation 4.12 and corresponds to the circulation that was shed during the time-step.

Figure 4.3 illustrates the wake shedding. The newly created edges that are purple in color are assigned a circulation equal and opposite the corresponding element in $\vec{\Gamma}_{\text{new}} - \vec{\Gamma}_{\text{old}}$; this is analogous to the "bound circulation" often discussed in lifting-line theory. The newly created edges that are dark red in color are assigned a circulation equal to the sum of the corresponding edge circulations along the trailing edge (note the matching colors); this is analogous to the “shed circulation” in lifting-line theory. Finally, the newly created edges that are green in color are assigned a circulation such that the circulation around the perimeter of the new wake panels sums to zero. The gray edges are those remaining from previous iterations, and thus their circulation is unchanged.

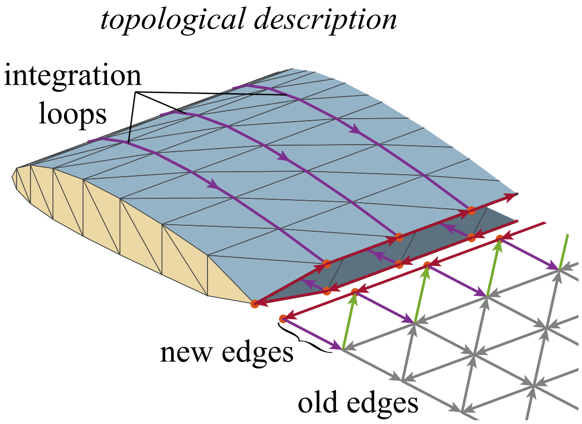


Figure 4.3: Triangulated body and wake vortex sheets illustrating the wake shedding procedure. Note that the groups of orange vertices are actually collocated (the wake is attached to a closed trailing edge).

4.3 Algorithm for Combined Bound and Free Vortex Sheets

We now have all the equations and algorithms necessary to describe the overarching procedure for the simulation, presented in Algorithm 4.1.

Algorithm 4.1 Main simulation

1. **Create the wing:** Create the wing triangulation and calculate the local triangle and vertex basis vectors.
2. **Initialize:** Initialize the bound vortex sheet by calculating A_{nf}, A_{ppc}, A_{Γ}, and setting up the quadratic program.
3. **First solution:** Calculate the first bound vortex sheet solution with Equation 4.8, letting $\vec{\Gamma}_{\text{old}} = \vec{0}$.
4. **Shed first wake row:** Shed the first row of wake panels from the trailing edge using the recently calculated $\vec{\Gamma}_{\text{new}}$ and the shedding procedure described in Section 4.2.
5. **Iterate:**
    - (a) **Free influence:** Obtain the surface vorticity on the free vortex sheet from the stored edge circulations as described in Section 2.3.2. Then, calculate the influence of the free vortex sheet on the other vortex sheets with the treecode.
    - (b) **Solve:** Solve for the bound vortex sheet strength. This should only require a couple iterations when using the previous solution as the initial guess.
    - (c) **Bound influence:** Calculate the influence of the bound vortex sheet on the free vortex sheet.
    - (d) **Displace:** Displace the vertices defining the free vortex sheet according to the chosen numerical ODE algorithm.
    - (e) **Shed:** Shed a new row of wake panels from the trailing edge.
    - (f) **Adaptive Paneling:** Apply the adaptive paneling algorithm described in Section 2.3.4 to the free vortex sheet.

4.4 Comparison to Rectangular Wing Experiment

As a first test of our method, we compared it to a set of experiments conducted by McAlister and Takahashi in the NASA Ames 7 ft x 10 ft subsonic wind tunnel [157]. Because they were interested in observing the formation of the vortex near a wing tip, the authors measured wake velocities and on-body surface pressures for various rectangular wings in a uniform freestream. This provides a straightforward comparison to our simulation. In the specific experiment to which we compared, McAlister and Takahashi used a straight rectangular wing with a NACA 0015 airfoil, a flat (square) tip, a 1.7 ft chord, and a 5.59 ft semispan (aspect ratio 6.576), operating at a 12° angle of attack and a Reynolds number of 2,500,000.

For our simulation we used the same wing and flow parameters, except that the freestream velocity and wing chord were both normalized to unity. The simulation progressed 1500 iterations with a time step of 0.02, which means the starting vortex ended approximately 30 chord-lengths or 4.5 span-lengths downstream. The wing comprised 11,424 triangles, and the wake comprised 228,547 triangles at the last iteration. Lastly, the smoothing parameter for the free vortex sheet was 0.1. The shadowgraph and radiograph of this simulation are presented in Figure 4.4, which indicate the presence of a well-formed vortex core composed of two trailing vortices and the starting vortex.

The comparison of measured and predicted surface pressures is presented in Figure 4.5, in which the data has been digitized from Figure 22(c) in McAlister [157]. McAlister noted that wall corrections were not applied to the pressure data. The chordwise pressure distributions are plotted individually for different spanwise locations indicated by y/s, where s is the semispan, and the predicted pressures are interpolated over the structured triangulation to match the pressure tap locations. In general there is strong agreement between measured and predicted values, except at the tip. McAlister deduced that part of the wake is shedding along the tip upstream of the trailing edge and forming a vortex, which provides the observed suction peak near the tip. The wake in our simulation was prescribed to shed from the trailing edge—and not upstream along the tip—thus, our simulation does not replicate this behavior. Another difference is observed at the trailing edge, where the predicted pressures approach stagnation but the measured pressures do not. This is due to our inviscid flow assumption, which is invalid at the trailing edge because the boundary layer thickness is not small compared to the airfoil thickness. Note that the pressure at $x/c = 1$ provided by McAlister is actually extrapolated from the neighboring pressure measurements.

Next, a comparison of wake velocities near one of the trailing vortices is provided in Figure 4.6, in which the data has been digitized from Figure 39 in McAlister [157]. The experimental velocities were measured with a two-color laser velocimeter in scans along the span direction (y) at different locations downstream of the trailing edge. The velocities predicted at the end of the simulation, which are sufficiently steady, are clearly well short of the measured values. The smoothing parameter $\delta = 0.1$ is providing too much smoothing and is overly restricting the peak velocity in the vortex core. If we instead use a smaller $\delta = 0.015$ with the same predicted wake and the circulation therein, a much closer match is observed. To be clear, the yellow curves in Figure 4.6 are calculated from the same wake half-edge circulations, only with less spreading of vorticity. Unfortunately, conducting the simulation with such a small smoothing parameter is quite difficult because of the rapid formation of small eddies and other chaotic features [71, 111, 158]. Note that in Figure 4.6 the origin for the x, y, z coordinates has been translated to the left tip trailing edge.

Lastly, a comparison between the measured and predicted spanwise circulation distribution is depicted in Figure 4.7, in which the data has been digitized from Figure 37(a) in McAlister [157]. This experiment used the same parameters as the previous two, except with a Reynolds number of 1,500,000. McAlister calculated circulation from velocity measurements taken around a chordwise rectangular path encasing the wing at different spanwise stations. For a direct comparison, we calculated circulation in the same manner, using off-body velocities calculated at the end of the simulation with $\delta = 0.1$, and these

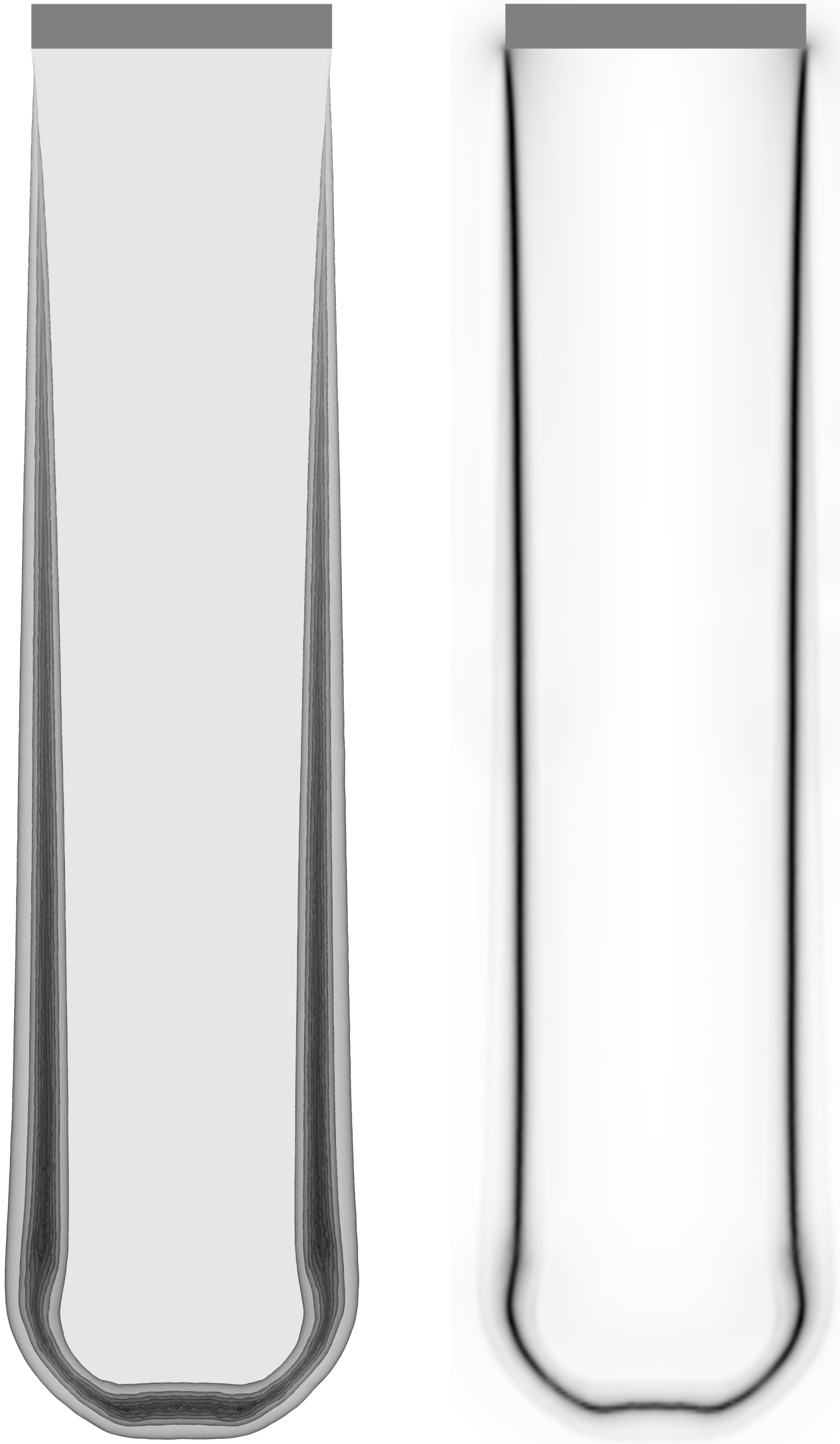


Figure 4.4: Shadowgraph and radiograph of the simulated wake.

predictions are indicated by the red square markers. Also, the yellow curve indicates the circulation calculated as $\vec{\Gamma} = A_{\Gamma}\vec{x}$ (Equation 4.1), which involves integrating the surface velocity along the bound vortex sheet (the purple lines in Figure 4.3). The predicted circulation closely matches the data except near the tip, where the simulated wake spreads vorticity more than is measured win the wind tunnel experiment.

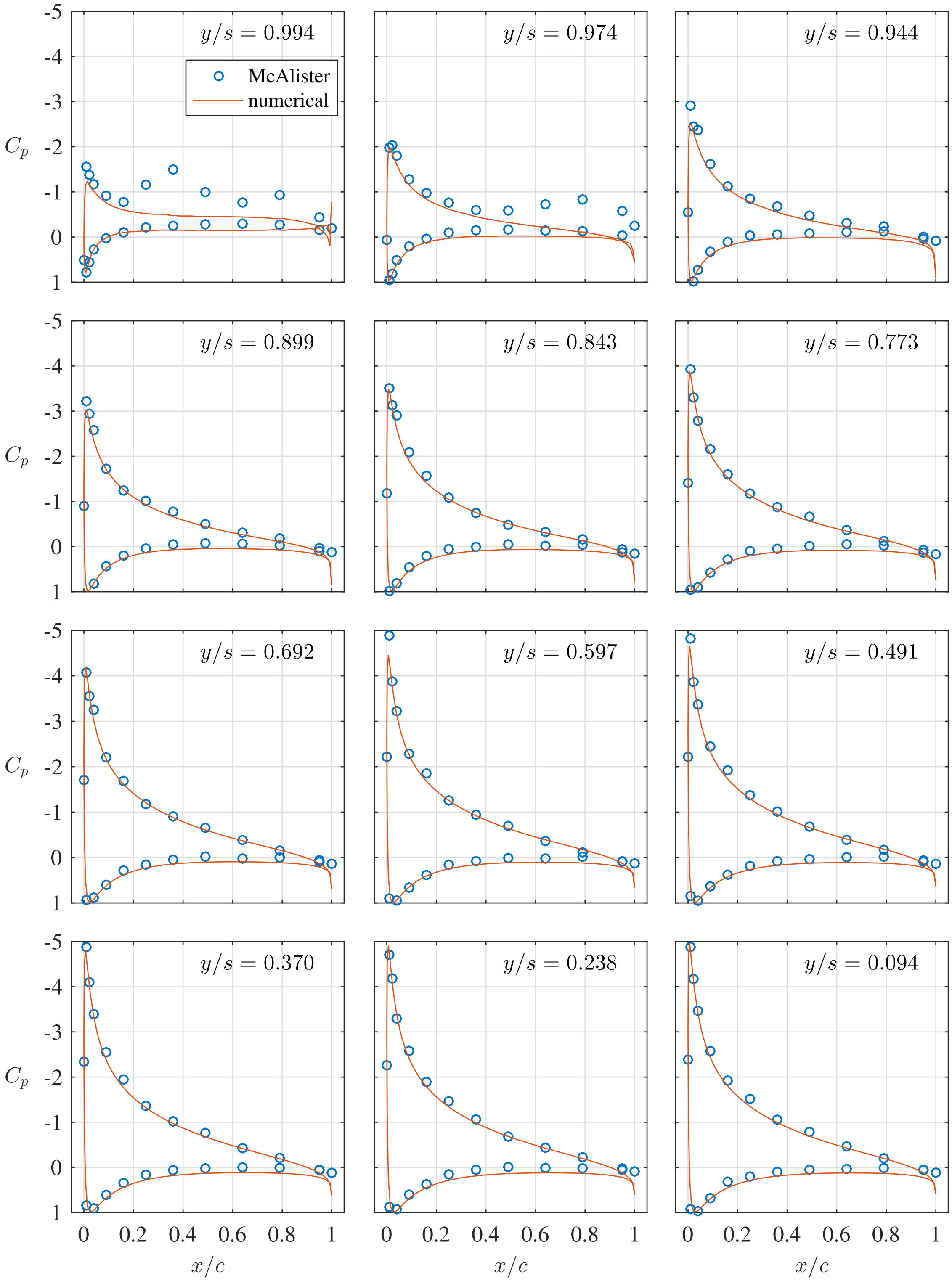


Figure 4.5: Comparison to Figure 22(c) in McAlister [157].

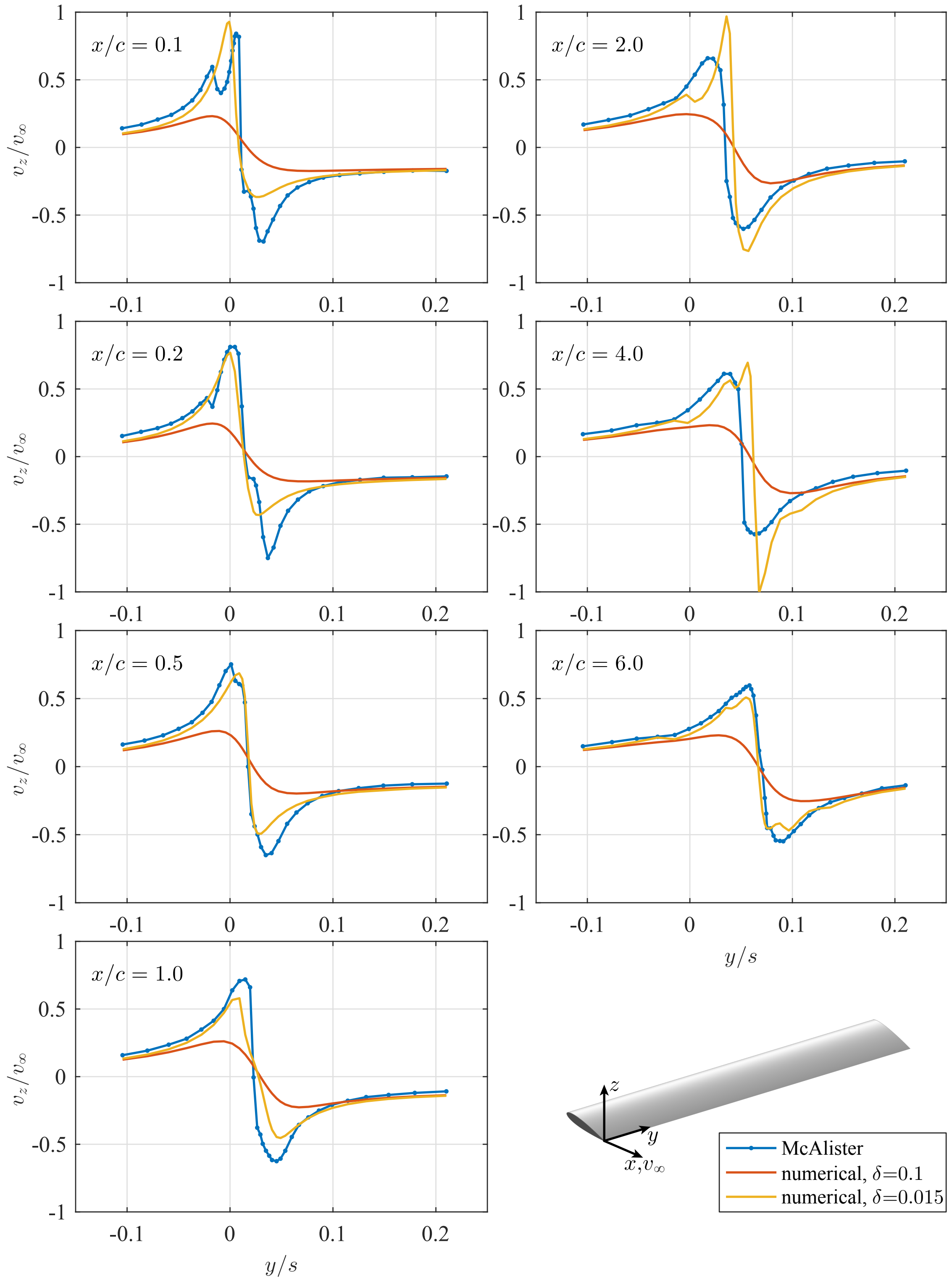


Figure 4.6: Comparison to Figure 39 in McAlister [157].

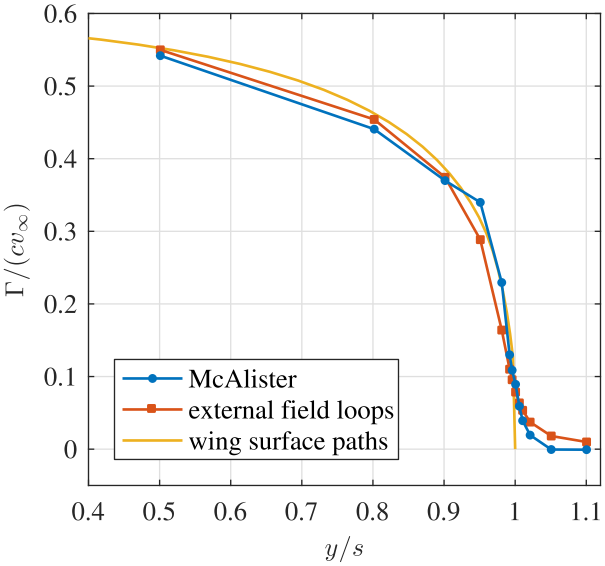


Figure 4.7: Comparison to Figure 37(a) in McAlister [157].

4.5 Comparison to Wind Turbine Experiment

As a second test of our method, and to more closely examine a complex wing–wake interaction, we compared to the results of the National Renewable Energy Laboratory (NREL) unsteady aerodynamics experiment (UAE) phase VI, which examined a horizontal axis wind turbine in the NASA Ames 80 ft x 120 ft wind tunnel. As documented by Hand et al. [159], the two-blade wind turbine has a 10 m diameter, straight taper, nonlinear twist, and uses the S809 airfoil.

The blade triangulation used in our simulation is depicted in Figure 4.8 alongside the line drawing from Figure 9 in Hand et al. [159]. We did not include the region near the root in which the blade transitions from a circular cross-section to the S809 airfoil because the blade twist reduces to 0° at the root, which results in a very large local angle of attack and would likely incur fully separated flow. Figure 4.8 also indicates the radial position of the pressure measurements used for comparison. The structured triangulation for each blade comprised 6502 triangles, including 402 triangles for the flat tips.

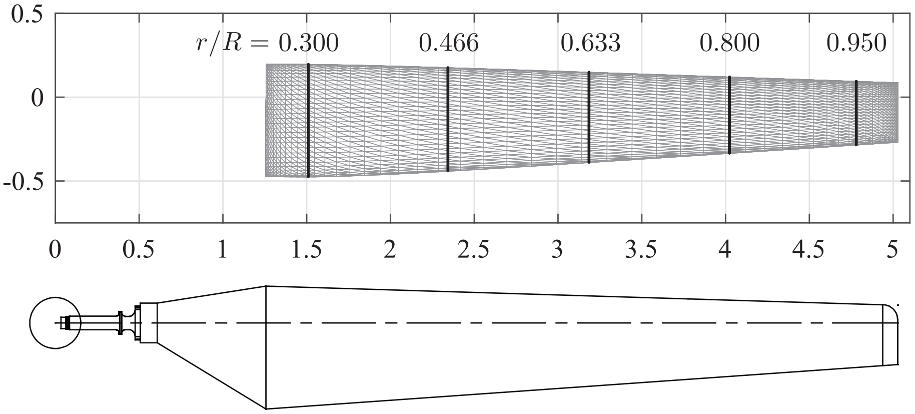


Figure 4.8: Blade triangulation and pressure measurement locations juxtaposed with Figure 9 in Hand et al. [159].

We compared to a subset of the experiments in which the blade rotated at 71.63 rpm in freestreams of 5 m/s and 7 m/s, with $0°$ yaw, and with a blade tip angle of $3°$. The tip Reynolds number is on the order of the one million. We used a time step of 0.0025 seconds, which corresponds to 335 time steps per blade revolution, and the simulation progressed for a total of four revolutions, or 1340 iterations. At the last iteration, the wakes together comprised 650,126 triangles for the 5 m/s case and 1,165,176 triangles for the 7 m/s case, and these wakes are presented in Figures 4.9 and 4.10, respectively. Strong vortices form behind both the root and tip of each blade, with the vortices at the tips carrying more vorticity.

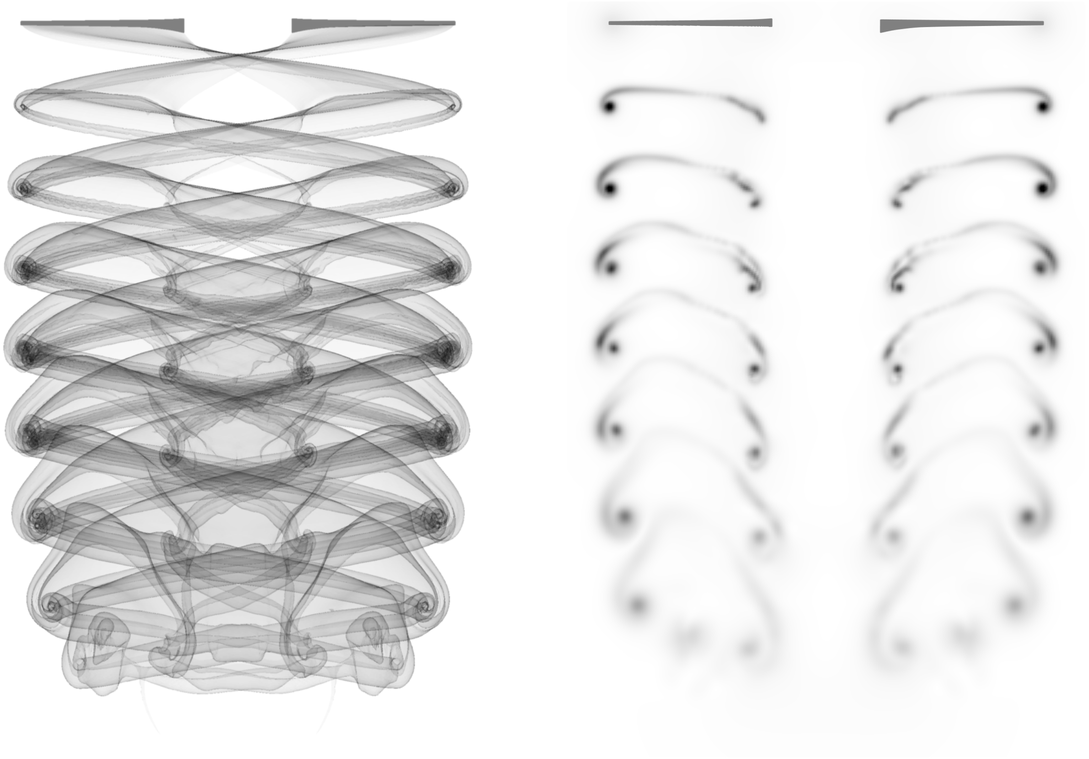
An unexpected phenomenon that arose in these simulations was the lasting effect of the starting vortex. When a blade passed over the starting vortex (after $180°$), there was a reduction in circulation around the blade, followed quickly by a restoration thereof, as would be expected. However, this transit past the starting vortex deposited a significant amount of vorticity into the wake which then affected the passing of the next blade (after another $180°$), and this process cascaded through each subsequent revolution of the rotor.

The radial and axial positions of the tip vortex, as defined by the peak concentration of vorticity, are presented in Figures 4.11a and 4.11b for the 7 m/s simulation. The persistent effect of the starting vortex can be seen in these figures at regular intervals of $180°$. These tip vortex locations are compared to predictions from Lynch et al. [160], who used FUN3D, the unstructured Navier–Stokes code from NASA Langley. The small fluctuations in the radial position prediction are due to the adaptive paneling scheme.

Finally, the predicted section pressures after four rotor revolutions are compared to the measurements from the NREL experiments. Figure 4.12 displays the comparisons for both the 5 m/s and 7 m/s cases. In general, the predictions are quite close to the measurements. As mentioned previously, the predicted pressures at the trailing edge tend toward stagnation, whereas the measured pressures stay closer to the ambient pressure. Note that for these rotating blades the pressure coefficient (Equation 3.21) is modified as

$$C_p := 1 - \frac{\boldsymbol{v} \cdot \boldsymbol{v}}{v_\infty^2 + v_{\text{loc}}^2} - \frac{2}{v_\infty^2 + v_{\text{loc}}^2} \frac{\partial \Phi}{\partial t}, \tag{4.16}$$

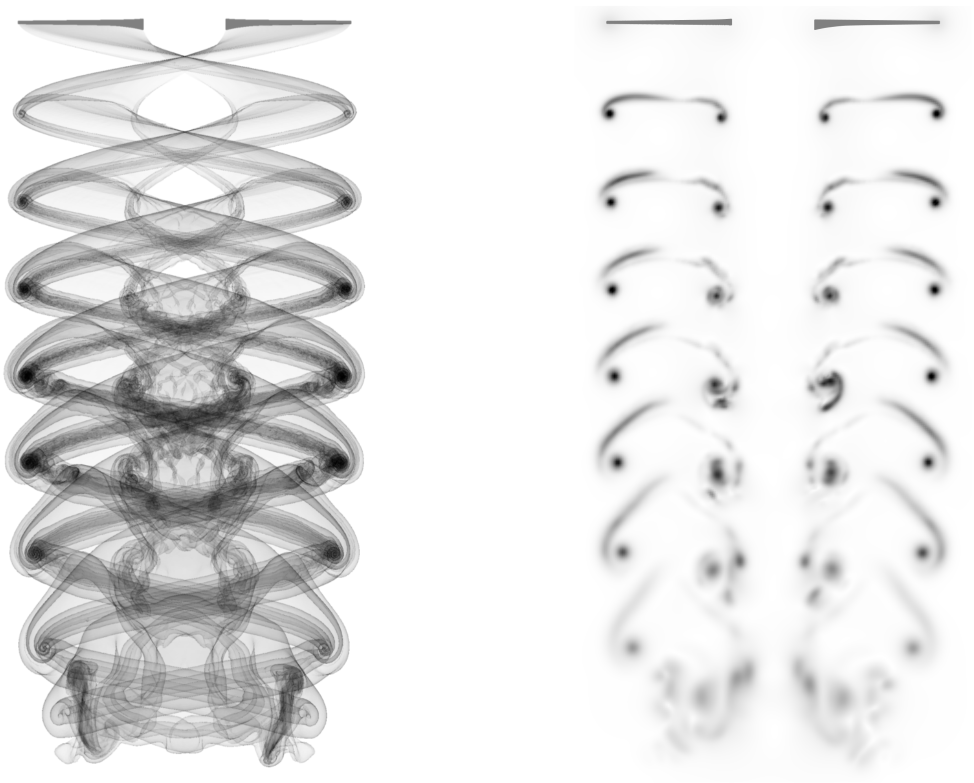
where v_{loc} is the local speed of the measurement point, which depends on the radial position along the blade. Integrating these surface pressures gives a predicted shaft torque of 988 Nm for the 7 m/s case, as compared to the 801 Nm reported by Simms [161]. This is an overprediction of 23%, which is likely due to not including the effect of viscous drag.



(a) Shadowgraph

(b) Cross-section (slice) of vorticity magnitude

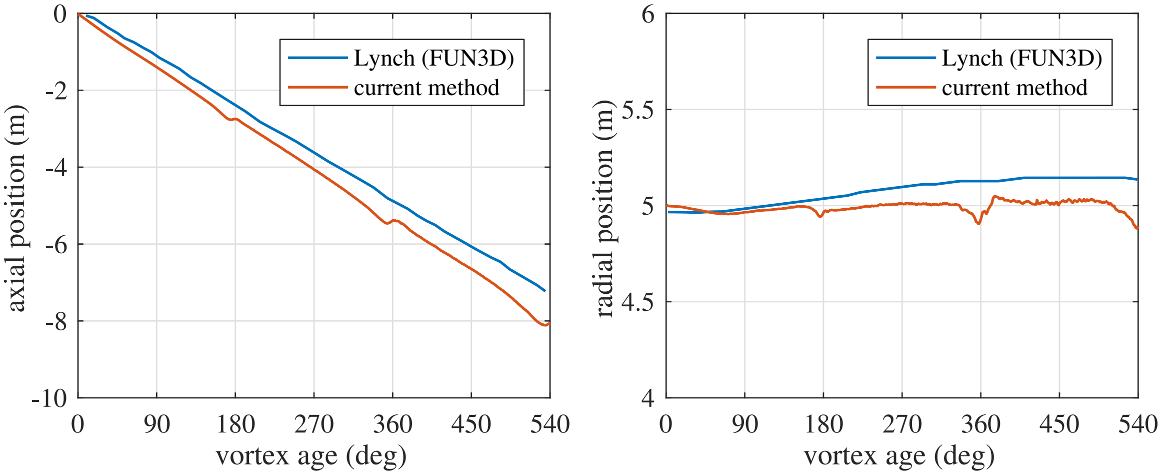
Figure 4.9: Wake visualizations for 5 m/s freestream.



(a) Shadowgraph

(b) Cross-section (slice) of vorticity magnitude

Figure 4.10: Wake visualizations for 7 m/s freestream.



(a) Vortex axial position

(b) Vortex radial position

Figure 4.11: Location of tip vortex compared to the prediction of Lynch et al. [160] for the 7 m/s case.

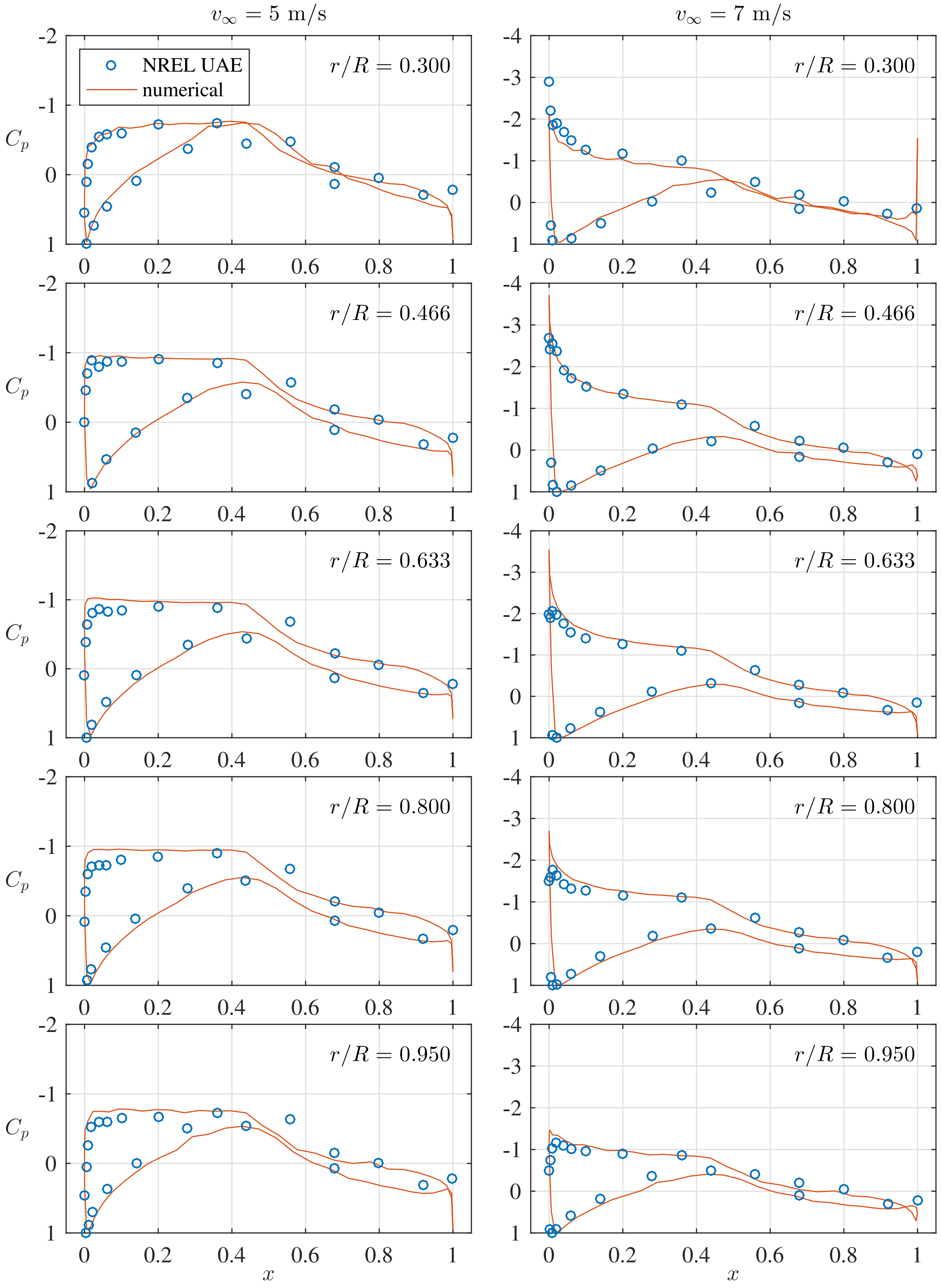


Figure 4.12: Section pressures for 5 m/s (left) and 7 m/s (right) cases compared with NREL UAE measurements [159]. Note the different vertical axis scales.

4.6 Example Simulation: Two Aircraft in Close Formation

As a demonstration of an intended application of our numerical method, consider two aircraft flying in a close formation, with the tailing aircraft slightly lower and to the right of the leading aircraft, as indicated in Figure 4.13. The example aircraft is a notional flying wing created in OpenVSP [162] and triangulated with the OpenVSP CFD unstructured meshing tool, resulting in 6390 triangles (each). The reference span length is 12.5 ft and the wing section is a NACA five-series reflexed airfoil with five degrees of washout at the tip. The aircraft are both prescribed an $11.5°$ angle of attack and their positions are held fixed during the simulation.

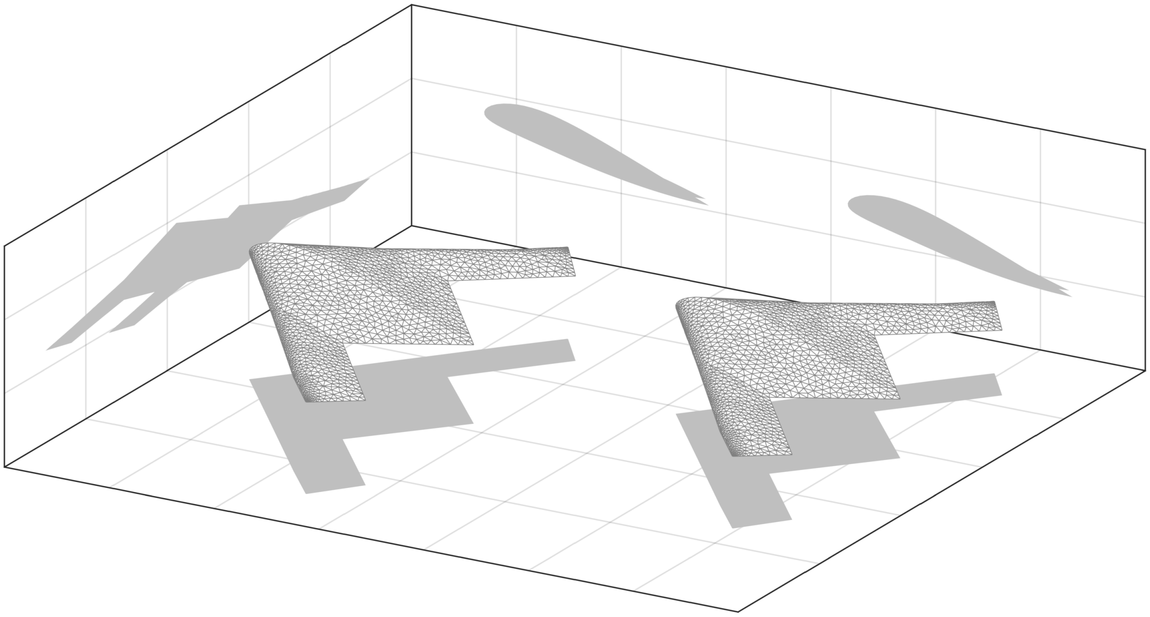


Figure 4.13: Two notional flying wings arranged in a close formation.

The simulation proceeded for 800 iterations with a time-step of $\Delta t = 0.04s$, for a total duration of 32 seconds, and with a freestream velocity of $v_\infty = 1$ ft/s. The smoothing parameter used for the wake is $\delta = 0.2$, which is in the same proportion to the reference length as the smoothing parameter used in the validation studies. The results of this simulation are presented in Figures 4.14–4.18. By the last iteration, the two wakes comprised 3,253,078 triangles.

The computing times for each step of Algorithm 4.1 are displayed in Figure 4.14. As expected, the influence velocity calculations dominate the computing time spent during each iteration, especially as the number of wake panels grows more rapidly near the end of the simulation. The adaptive paneling algorithm, which is executed entirely on the CPU, also consumes a significant amount of computing time refining the wake shed from the leading aircraft.

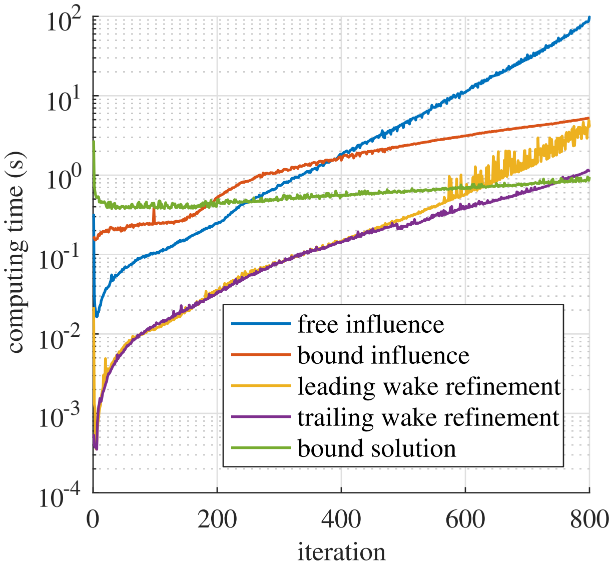


Figure 4.14: Individual computing times for each calculation during an iteration.

The radiograph, which shows the concentration of vorticity carried by the wake, is illustrated in Figure 4.15. In this figure, the wings are transparent so any vorticity passing beneath the wing can also be observed. Most of the wake from the leading aircraft passes beneath the trailing aircraft, except for the right tip vortex, which passes above the trailing aircraft. The tip vortices emerging from the right tips of each aircraft interact in an unstable and chaotic manner. The transition to chaos appears to happen shortly downstream from the trailing aircraft. The vorticity shed by the trailing aircraft lacks left–right symmetry—the right side is shedding far more vorticity than the left. This is likely due to the left tip vortex from the leading aircraft providing a beneficial upwash, while the right tip vortex alters the lift distribution much less favorably.

The shadowgraph, which reveals the shape of the vortex sheets, echoes the observations from the radiograph. The dark regions correspond to very large numbers of triangles, and suggest that the chaotic interactions of the two wakes lead to a much greater computational burden than an isolated plain wing (e.g., Figure 4.4). A second shadowgraph is presented in Figure 4.17 to clarify the path of the wake from the leading aircraft as it passes both above and below the trailing aircraft.

Finally, the surface pressures are presented in Figure 4.18. The images of the bottom surface of each wing are mirrored such that the tips are aligned with the corresponding top views (as if the bottom surface could be viewed from above). Both the highest and lowest pressures throughout the flow field are observed around the right midspan of the trailing aircraft. In fact, the peak positive pressure coefficient exceeds 1.0, indicating a significant unsteady motion. Of course, this is a direct result of the tip vortex from the leading wing. The asymmetric pressure distribution on the trailing aircraft results in a rolling moment toward the left. In fact, this rolling moment would be reversed if the impinging vortex passed beneath the right wing instead of above it.

The prediction of these surface pressures relied on an accurate representation of the wake shed by the leading aircraft, as well as accurate near-body velocities to carry this wake closely around the trailing aircraft. This provides the principal motivation for this thesis.

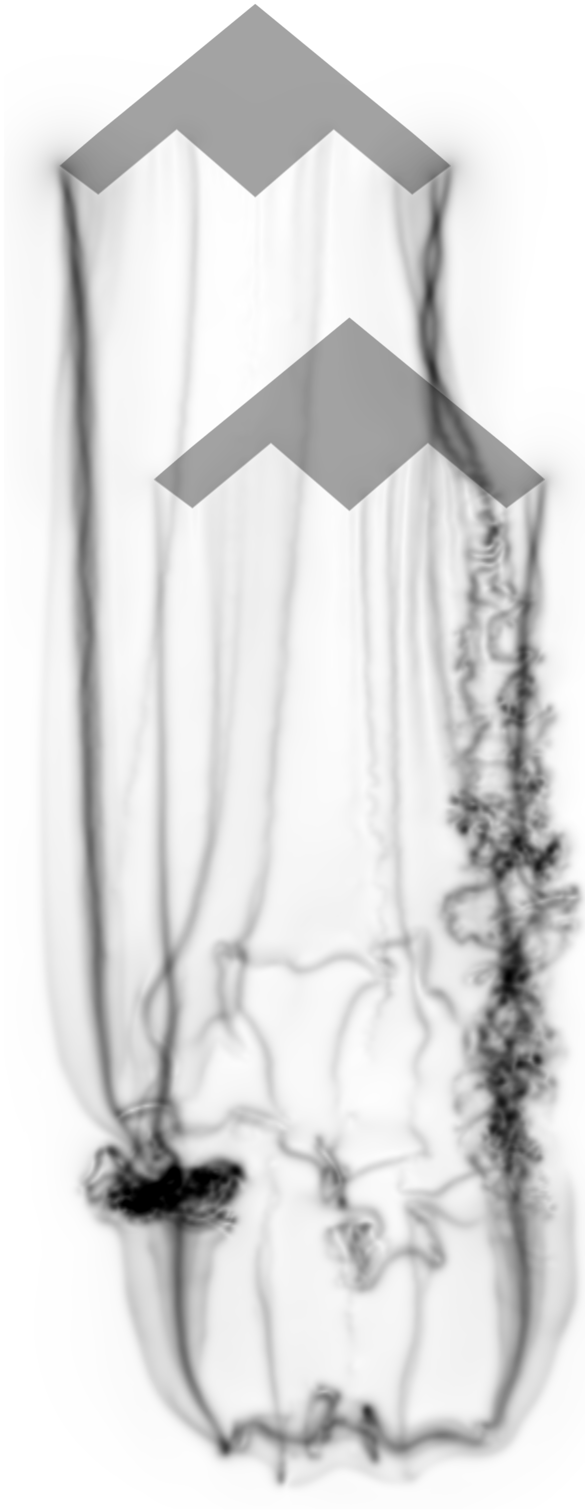


Figure 4.15: Radiograph of the simulated wakes. Darker regions indicate strong vorticity.

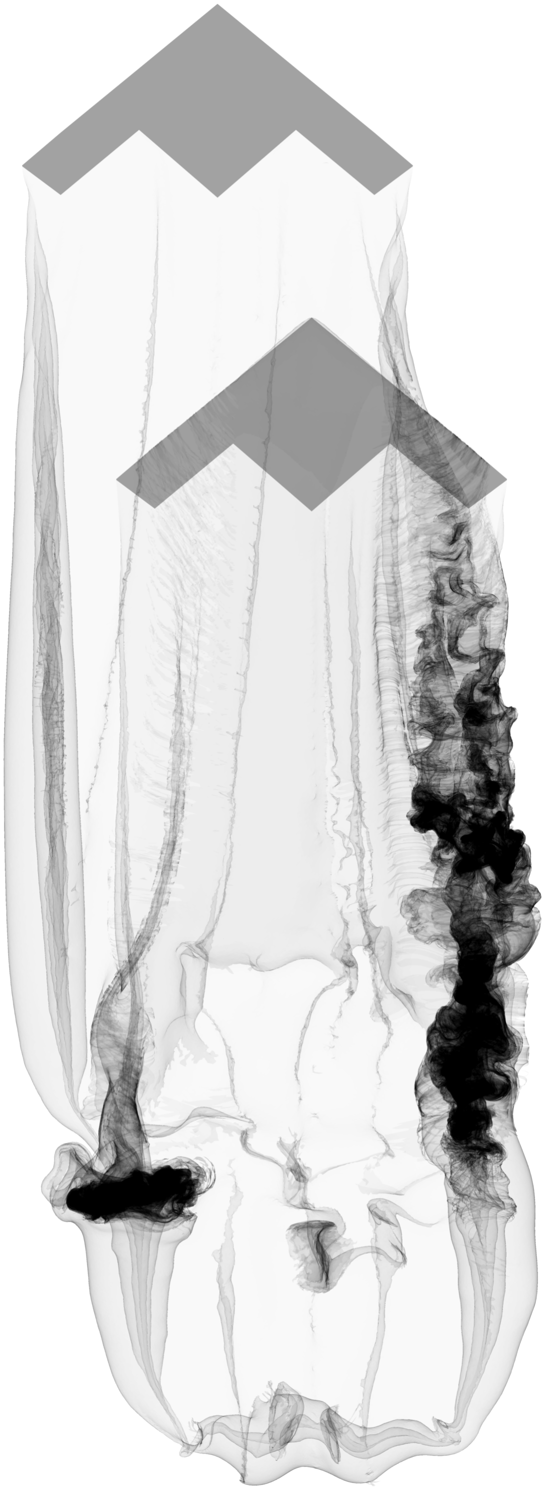


Figure 4.16: Shadowgraph of the simulated wakes. Darker regions indicate many layers of the vortex sheet.

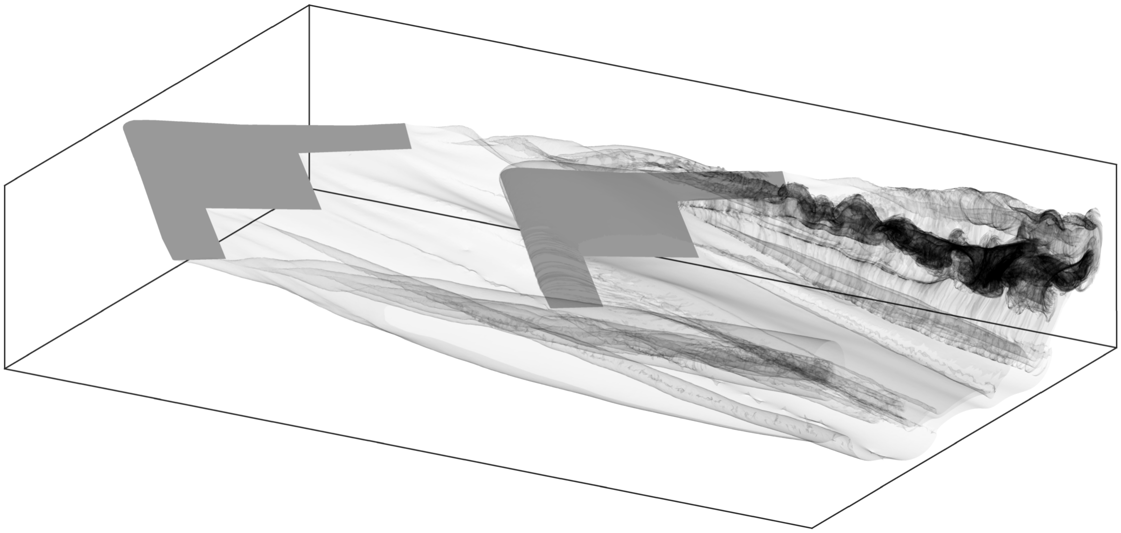


Figure 4.17: A shadowgraph with a closer view of the aircraft, which are not transparent.

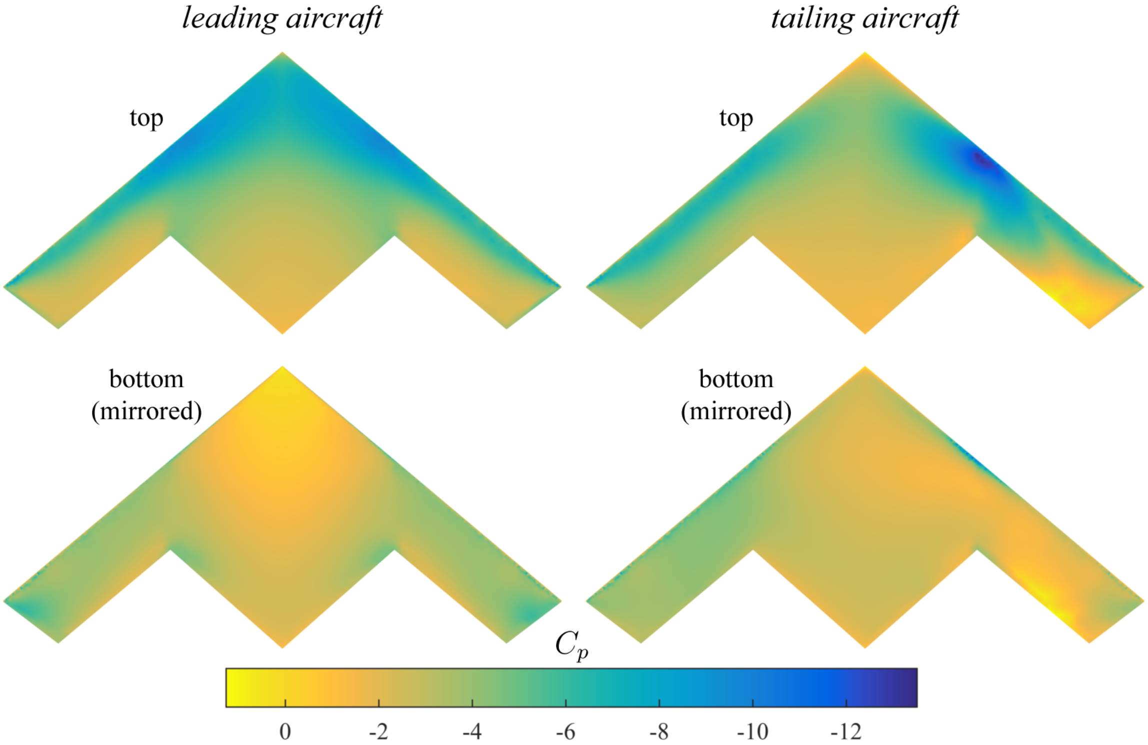


Figure 4.18: Surface pressures for each side of each aircraft. The bottom surfaces are left–right mirrored.

CHAPTER 5

CONCLUSIONS

In this thesis, we described a new surface vorticity boundary element method to predict intricate wake–body interactions for incompressible high Reynolds number flow. We were motivated by the impressive detail and rigor of vortex sheet simulations in the vortex methods literature [1], as compared to traditional panel code wake models. In a traditional force-free wake model [10, 13, 14], the wake is represented by a lattice of vortex filaments that progress through the flow in a Lagrangian manner while maintaining constant circulation. Plausible wake evolution behavior was achieved in these models by simply applying a viscous core model to the Biot–Savart equation to limit the effects of the velocity singularity near a filament. However, these wake models quickly become non-physical as they evolve. Thus, we expected that significant improvement could be achieved by incorporating concepts from the vortex methods literature.

On the other hand, the source–doublet panel method for representing potential flow around a body has been well developed into effective high-order models, but the surface vorticity method has not received similar attention. In this work, we recognized that bound vortex sheets and free vortex sheets are two separate descriptions of the same object, and, therefore, we hypothesized that triangular elements with linearly varying surface vorticity could be used to effectively described the entire flow, including not only wake elements but also elements on aerodynamic surfaces. These higher-order elements would provide accurate streamlines around a bound vortex sheet and also underpin the well-behaved evolution of a free vortex sheet. Therefore, we pursued a boundary element method that used the same surface vorticity elements for both wings and wakes. Many lessons have been learned along the way.

5.1 Free Vortex Sheets

High-order wake panels provide a more realistic induced velocity field than singular particles or filaments. However, we found that they are still insufficient for well-behaved wake evolution because thin vortex sheets naturally roll up into a curvature singularity [90]. Indeed, this had been discovered in the vortex methods literature [86], and the solution is to regularize the vortex sheet. We adopted a common regularization technique of including a smoothing parameter in the Biot-Savart kernel, which acts as a surrogate for viscosity in high Reynolds number flows and bounds the peak concentration of vorticity.

Yet, this regularization is still insufficient to provide an accurate representation of a vortex sheet. As the elements strain, they may become too large or too irregular to adequately resolve the sheet. The discretization must be refined to maintain an appropriate level of detail; the lengths of the edges in the triangulation should not significantly exceed the regularization length.

The main benefit of representing a free vortex sheet with linear surface vorticity panels is the ability to resolve vorticity accurately over a relatively large space, which would otherwise require many vortex particles for a similar level of accuracy, as demonstrated in Section 2.3.2. However, this advantage only applies when the vortex sheet is mostly flat, which corresponds to regions with little vorticity. In regions of concentrated vorticity, which cause the sheet to roll up into a spiral with many layers, higher-order triangular panels lose their advantage because large numbers of panels are still required. Furthermore, when the layers of the rolled-up sheet are much closer together than the regularization length (the smoothing parameter δ), a sheet representation becomes unnecessary and potentially misleading.

Our adaptive paneling algorithm included a Delaunay edge flipping scheme that helped maintain more regular triangles. It also had a beneficial effect of reducing the number of edges that needed to be split because the flipped edges tended to become shorter. Our implementation of this algorithm required the sheet to be a locally planar graph, which requires an edge to neighbor exactly two triangles, unless it is on the boundary. Unfortunately, maintaining a locally planar graph was very challenging, and it required many specialized and detailed numerical procedures to avoid erroneous results.

5.2 Bound Vortex Sheets

We pursued a higher-order surface vorticity representation of bound vortex sheets for two reasons: (1) a convenient congruity with the free vortex sheets, for which the equations governing the panels had already been developed and implemented, and (2) to provide accurate predictions of velocity, both on and near the surface of a wing. Fortunately, our expectations were met. Our method could also be considered a velocity-based boundary element method because surface vorticity and surface velocity are interchangeable via the surface normal vector.

Defining two degrees of freedom for surface vorticity at each vertex of a triangulation was a natural choice to ensure C^0 continuity of γ over the entire body, which is critical for accurate surface velocities. Additionally, the panel centroids are a natural choice to enforce no normal flow because the velocity induced by a panel is indeterminate at the edges and vertices. This should come as no surprise, because a triangular vortex sheet could not physically exist in isolation unless the strength dropped to zero at the edges. Selecting surface vorticity instead of doublet or source strength requires the imposition of the panel perimeter circulation constraint to enforce $\nabla \cdot \boldsymbol{\gamma} = 0$. Fortunately, this calculation contributes only a negligible cost to the solution and is relatively easy to implement.

As a result of these choices for constraints and degrees of freedom, there are approximately twice as many constraints as degrees of freedom. We obtain a surface vorticity solution by formulating a least squares minimization problem. Unfortunately, this requires the multiplication of two large and full matrices. At the same time, the least squares formulation is also liberating. Additional constraints can be freely included; for example, some panels could have multiple collocation points. Also, the least squares approach gives the freedom to control which residuals are the most important to minimize.

Our boundary element method is especially accurate and robust for closed surfaces, such as an ellipsoid, including with random or irregular triangulations. Though small in contribution, the vector rotation scheme was critical to the accuracy of the bound vortex sheet method (Section 3.2). However, the lifting solution for a wing was more sensitive and required more careful weighting of the residuals.

It is also worth clarifying the difference between the surface vorticity method and a traditional source–doublet panel method. Aside from their shared classification as boundary element methods, there are several fundamental differences. The potential formulation assumes the flow to be irrotational and then arrives at a Laplace equation for a scalar potential field. The surface vorticity method, however, assumes the vorticity to be concentrated as a vortex sheet at the surface of the body, and the problem formulation is arranged to achieve a valid bound vortex sheet (stagnant internal flow and a solenoidal surface vorticity field). Instead of solving a Laplace equation, a Poisson equation is solved via the Biot–Savart law. The surface vorticity method has an additional burden of constraining the degrees of freedom to produce a solenoidal surface vorticity field, but the resulting benefit is that surface velocity is directly obtained from the solution without requiring the gradient computations. In some versions of the panel method, wakes are prescribed to extend into the body; this is not necessary for the surface vorticity method. Indeed, the flow is stagnant inside a closed body and the wakes emanate externally from a prescribed shedding path (e.g., a trailing edge).

5.3 Combined Bound and Free Vortex Sheets

The bound and free vortex sheets are combined into a complete simulation via the wake shedding scheme at the wing trailing edge, which enforces the unsteady pressure matching Kutta condition. This constraint requires tracking the time rate of change of circulation around the wing. Ideally, any path starting from the lower surface and ending at the corresponding vertex on the upper surface would correctly predict the circulation, which is equivalent to the potential jump at the trailing edge. However, this path independence relies on the panel perimeter circulations being zero, which is not perfectly achieved. Unfortunately, the circulation calculation appears to be sensitive to the errors accumulated along these paths.

The large matrix M, inside the bound vortex sheet problem formulation 4.8, can be computed and inverted once, and then stored to be accessed at each iteration. Accordingly, computing the solution for the strength of the bound vortex sheet is inexpensive compared to the other computing tasks, as demonstrated in Figure 4.14. However, this computation is only possible when the aerodynamic bodies are not in relative motion. For example, M remains constant for isolated rotating helicopter blades, but M would vary with time if a non-rotating component were included (e.g., a fuselage). This limitation could be remedied with a least-squares GMRES approach [163–165].

5.4 Realizability Constraints

Throughout this thesis, several parameters, thresholds, and other choices were presented without clear guidance for their values. However, these choices can affect the quality of the simulation.

For non-lifting flow, the solution is insensitive to the quality of the triangulation, but this is not the case for lifting flow. To achieve a quality solution for a wing, it is important to refine the triangulation close to the tips and to smoothly continue the resolution all the way around the tip, whether the tip is round or square. Also, there is not an obvious choice for the vertex normal vector at the trailing edge of the tips, but this normal vector determines the plane in which surface velocity must reside. Placing the normal vector in the plane of the wing chord and facing outward typically produced the best results.

For lifting flow, it is important to choose the weighting factor (Equation 3.51) carefully. When the panel perimeter circulation residuals are normalized by the average panel area, the weighting factor should be between 0.85 and 0.98.

The smoothing parameter δ is analogous to the thickness of a vortex sheet, and is, therefore, connected to the length scale of the simulation. The spans of the wings used in this thesis were close to 10 (feet or meters), and $\delta = 0.1$ offered a reasonable balance between accuracy (vortex core size) and computational cost (number of computational elements). Below $\delta = 0.05$, the free vortex sheet tended toward rapid instability. Experimental measurements could be used to tune δ to achieve the measured peak vorticity.

Similarly, the threshold lengths for the adaptive paneling edge splitting and removal schemes must also achieve a balance between cost and accuracy. In most simulations, we used a maximum edge length of 0.125 and a minimum of 0.01. Also, we found that newly created panels should forgo any adaptive paneling for several iterations to prevent any noise from feeding back into the bound vortex sheet solution.

Lastly, the time-step is also an important parameter. It contributes to the unsteady terms in the bound vortex sheet solution and the vertex displacements of the free vortex sheets (e.g., a forward Euler step). For the simulations in this thesis with a freestream velocity of unity, a time-step between 0.01 and 0.02 provided sufficient detail while maintaining stability.

5.5 Summary of Contributions

The principal contribution of this thesis is the accurate and detailed simulation of complex wake–body interactions, unaccompanied by any volume meshes.

The wake is represented by triangular panels with linearly varying surface vorticity, which maintain constant circulation along every half-edge, a generalization of the constant strength elements used by Stock [63]. The triangular panels retain a nearly regular shape via the adaptive paneling algorithm, which includes a novel Delaunay edge flipping scheme.

Our higher-order surface vorticity boundary element method for determining the bound vortex sheet around a lifting body is a significant generalization of existing low-order surface vorticity methods, and is based on several enabling numerical methods that were novel to this work.

The analytic solution to the velocity induced by the linear strength elements allowed for efficient Biot–Savart calculations without the need for a quadrature scheme. The panel perimeter circulation constraint provides a simple approach to guaranteeing a solenoidal surface vorticity field, which was considered one of the main obstacles facing a surface vorticity method [151]. The vector rotation scheme ensures a fully continuous surface vorticity field across edges and vertices.

For these models and algorithms to successfully provide a detailed depiction of the flow, a large number of computational elements is required. The associated computations would not have been practical without the treecode and its implementation on a graphics processing unit.

The hypothesis stated in the introduction asserted that interacting aerodynamic flows with strong couplings between wakes and lifting bodies can be modeled efficiently and effectively as vortex sheets discretized into triangular panels with linearly varying surface vorticity. We have provided significant evidence to support this hypothesis, but we have also documented several areas for improvement.

5.6 Future Work

There are several remaining tasks that are worth pursuing. A well known challenge faced by panel methods (and by extension, boundary element methods) is the accurate calculation of induced drag [10, 166]. A common remedy when using straight prescribed wakes is to calculate induced drag in the Trefftz plane [167, 168]. Unfortunately, a Trefftz plane calculation is not available for unsteady wakes and a new technique may be required.

The least squares minimization formulation balanced the normal flow residuals with the panel perimeter circulation residuals via a weighting factor. We showed that the results were insensitive to this weighting factor for non-lifting flow, but for lifting flow the weighting factor had to be chosen more carefully. A more intricate weighting scheme could help produce better results for lifting flows and for unstructured triangulations.

Regions of strong vorticity tend to cause a free vortex sheet to roll up into a spiral with many layers, requiring a large number of panels. If fine detail in this region is not of interest, such as a starting vortex that has traveled far downstream, the smoothing parameter could increase with time, thereby slowing the growth rate.

In the adaptive paneling scheme, the decision to split or remove an edge is based entirely on the edge length, as compared to a global threshold. Of course, there are many other criteria that could be used, and including the circulation strength along the edge could be advantageous. For example, edges that carry little circulation should be favored for removal.

Appendices

APPENDIX A

EVALUATION OF THE BIOT–SAVART INTEGRAL

The analytic solution to the Biot–Savart integral in Equation 3.15 is critical to the implementation of the proposed method, particularly for the self-induction terms. Solutions to various forms of the Biot–Savart integral are presented by Suh [169]; however, the solution approach employed in this work is derived from that of Johnson [142] in the development of PANAIR. This method is valid for any polygon, but is described here only for a triangle.

A.1 Geometric Description

Consider the velocity $\boldsymbol{v}_A$ induced at point P_A from a sheet of vorticity $\boldsymbol{\gamma}$ linearly distributed over surface S is given by the Biot–Savart law as

$$\boldsymbol{v}_A = \frac{1}{4\pi}\int_S \frac{\boldsymbol{r} \times \gamma(\boldsymbol{r})}{||\boldsymbol{r}||^3} dS(\boldsymbol{r}) \tag{A.1}$$

with $\boldsymbol{r}$ directed from P_A to the differential dS. To solve this integral, we will decompose it into several more manageable integrals and introduce several coordinates.

Surface S is defined by the triangle (P_1, P_2, P_3), as indicated in Figure A.1. Plane P contains S, the first two vectors of orthonormal basis $(\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3)$, and point P_B, which is the projection of P_A onto the plane. The position vectors $\boldsymbol{p}_A$, $\boldsymbol{p}_1$, etc. describe the position of their respective points from a common origin $\mathcal{O}$, and vector $\boldsymbol{r}$ describes the relative position from P_A to the differential element dS. The projection of $\boldsymbol{p}_A$ onto plane P defines point P_B as $\boldsymbol{p}_B := \boldsymbol{p}_A - (\boldsymbol{p}_A - \boldsymbol{p}_1) \cdot \boldsymbol{b}_3 \boldsymbol{b}_3$. The height of P_A above P_B is $z := (\boldsymbol{p}_A - \boldsymbol{p}_B) \cdot \boldsymbol{b}_3 = -\boldsymbol{r} \cdot \boldsymbol{b}_3$, which gives $\boldsymbol{\rho} = \boldsymbol{r} + z\boldsymbol{b}_3$, where $\boldsymbol{\rho}$ is an additional vector directed to dS.

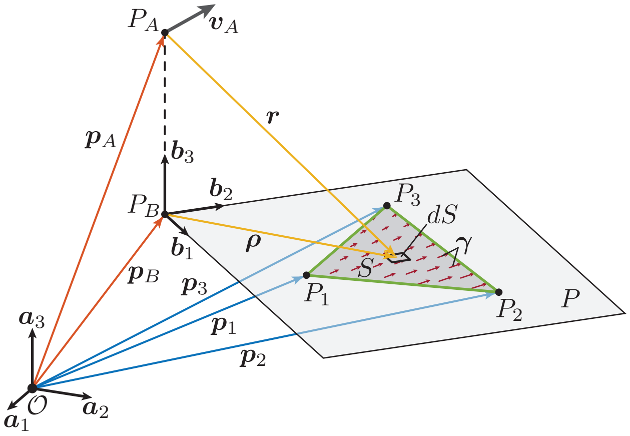


Figure A.1: Geometric relations between triangle (P_1, P_2, P_3) and point P_A measured relative to origin $\mathcal{O}$.

A.2 Specification of Domains of Integration

Instead of integrating on S directly, three new areas S_1, S_2, and S_3 are created to cover exactly S as shown in Figure A.2, where S_1 := triangle (P_B, P_1, P_2), S_2 := triangle (P_B, P_2, P_3), and S_3 := triangle (P_B, P_3, P_1). With vectors $\boldsymbol{q}_i := \boldsymbol{p}_i - \boldsymbol{p}_B,\ i = 1, 2, 3,$ the orientation of triangle S_i is given by the sign of

$$\boldsymbol{b}_3 \cdot (\boldsymbol{q}_i \times \boldsymbol{q}_{i+1}),\ i = 1, 2, 3 \text{ and } \boldsymbol{q}_4 := \boldsymbol{q}_1. \tag{A.2}$$

In the example illustrated in Figure A.2, S_3 has a negative orientation while S_1 and S_2 have a positive orientation, and, therefore, $S = \cup_{i=1}^{3} S_i$. The proof of this equality for a general triangle follows from the application of Equation A.2 to compute the area of S.

Then, the integral in Equation A.1 is rewritten as

$$\frac{1}{4\pi}\int_S \frac{\boldsymbol{r} \times \boldsymbol{\gamma}}{||\boldsymbol{r}||^3} dS = \frac{1}{4\pi}\sum_{i=1}^{3}\int_{S_i} \frac{\boldsymbol{r} \times \boldsymbol{\gamma}}{||\boldsymbol{r}||^3} dS_i. \tag{A.3}$$

In the next section, these integrals will be further decomposed.

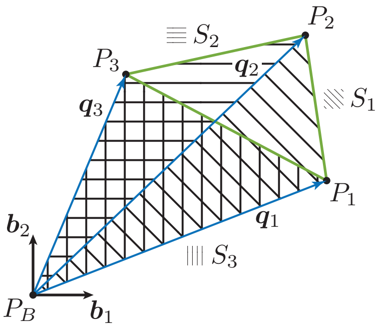


Figure A.2: Three surfaces, S_1, S_2, and S_3, used to cover exactly S. In this case, S_3 has opposite orientation of S_1 and S_2.

A.3 Coordinates for Integration

Without loss of generality, this section will focus on the integration of surface S_1, which corresponds to triangle (P_B, P_1, P_2) and vectors $\boldsymbol{q}_1 := \boldsymbol{p}_1 - \boldsymbol{p}_B$ and $\boldsymbol{q}_2 := \boldsymbol{p}_2 - \boldsymbol{p}_B$. Define a new orthonormal basis $(\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3)$ as

$$\boldsymbol{c}_3 := \boldsymbol{b}_3, \tag{A.4}$$

$$\boldsymbol{c}_2 := \frac{\boldsymbol{q}_2 - \boldsymbol{q}_1}{||\boldsymbol{q}_2 - \boldsymbol{q}_1||}, \tag{A.5}$$

$$\boldsymbol{c}_1 := \boldsymbol{c}_2 \times \boldsymbol{c}_3, \tag{A.6}$$

which is illustrated in Figure A.3. Three parameters are established to set the bounds of integration in the subsequent calculations: $a := \boldsymbol{q}_1 \cdot \boldsymbol{c}_1$, $l_1 := \boldsymbol{q}_1 \cdot \boldsymbol{c}_2$, and $l_2 := \boldsymbol{q}_2 \cdot \boldsymbol{c}_2$.

For the purpose of integration, a Cartesian coordinate system is created on plane P to describe $\boldsymbol{\rho}$ with independent variables $u := \boldsymbol{\rho} \cdot \boldsymbol{c}_1$, $v := \boldsymbol{\rho} \cdot \boldsymbol{c}_2$. Recall the height coordinate z, which gives $\boldsymbol{r} = u\boldsymbol{c}_1 + v\boldsymbol{c}_2 - z\boldsymbol{c}_3$ and $\boldsymbol{\rho} = u\boldsymbol{c}_1 + v\boldsymbol{c}_2$. The coordinates u, v are the independent variables of integration, and so $dS_1 = dudv$. Additionally, we also introduce a pair of polar coordinates (ρ, ϕ), where ϕ is measured from the line along $\boldsymbol{c}_1$ toward the

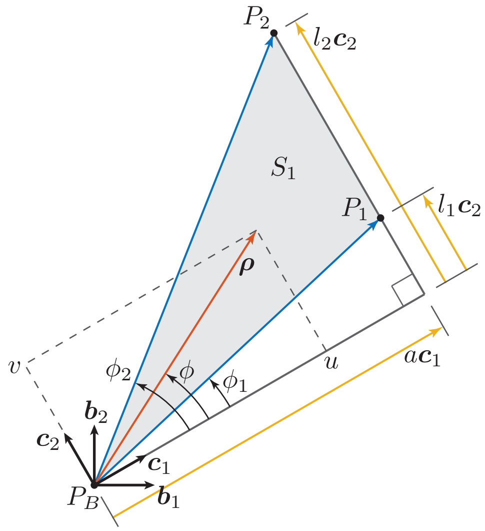


Figure A.3: Cartesian coordinates (u, v) and polar coordinates (ρ, ϕ) for integration on S_1.

direction of $\boldsymbol{c}_2$, $\rho = ||\boldsymbol{\rho}||$, and the substitution relations are: $u = \rho\cos\phi$, $v = \rho\sin\phi$, and $dS_1 = \rho d\rho d\phi$.

Therefore, surface vorticity strength is expressed in terms of two scalar functions as $\boldsymbol{\gamma}(u, v) = f(u, v)\boldsymbol{c}_1 + g(u, v)\boldsymbol{c}_2$ which vary linearly as

$$f(u, v) = a_{00} + a_{10}u + a_{01}v, \tag{A.7}$$

$$g(u, v) = b_{00} + b_{10}u + b_{01}v. \tag{A.8}$$

The numerator in Equation A.3 may be expanded as

$$\boldsymbol{r} \times \boldsymbol{\gamma} = zg(u, v)\boldsymbol{c}_1 - zf(u, v)\boldsymbol{c}_2 + [ug(u, v) - vf(u, v)]\,\boldsymbol{c}_3 \tag{A.9}$$

$$\begin{aligned} &= z(b_{00} + b_{10}u + b_{01}v)\boldsymbol{c}_1 - z(a_{00} + a_{10}u + a_{01}v)\boldsymbol{c}_2 \\ &\quad + (b_{00}u + b_{10}u^2 + b_{01}uv - a_{00}v - a_{10}uv - a_{01}v^2)\boldsymbol{c}_3 \end{aligned} \tag{A.10}$$

One difficulty in solving for the antiderivatives arises from the possibility of the denominator in the integrand to reach zero. To avoid this, a parameter $\delta > 0$ is introduced to force the denominator to be positive. Accordingly, the height of P_A above P_B is modified to $h := \sqrt{z^2 + \delta^2}$. The solutions subsequently obtained will be exact in the limit as $\delta \to 0$. Along with $\rho = \sqrt{u^2 + v^2}$ we have

$$r := ||\boldsymbol{r}|| = \sqrt{u^2 + v^2 + z^2 + \delta^2} = \sqrt{\rho^2 + h^2}. \tag{A.11}$$

The integral may now be expressed as

$$\int_{S_1} \frac{\boldsymbol{r} \times \boldsymbol{\gamma}}{||\boldsymbol{r}||^3} dS_1 = \int_{S_1} \frac{1}{r^3} \big[z(b_{00} + b_{10}u + b_{01}v)\boldsymbol{c}_1 - z(a_{00} + a_{10}u + a_{01}v)\boldsymbol{c}_2$$

$$+(b_{00}u + b_{10}u^2 + b_{01}uv - a_{00}v - a_{10}uv - a_{01}v^2)\boldsymbol{c}_3 \big]. \tag{A.12}$$

Because each power of u and v involve antiderivatives of different forms, we define the H integrals in a form similar to Johnson [142]:

$$H_{mn} := \int_{S_1} \frac{u^m v^n du dv}{r^3}, \tag{A.13}$$

which gives

$$\int_{S_1} \frac{\boldsymbol{r} \times \boldsymbol{\gamma}}{||\boldsymbol{r}||^3} dS_1 = z(b_{00}H_{00} + b_{10}H_{10} + b_{01}H_{01})\boldsymbol{c}_1 - z(a_{00}H_{00} + a_{10}H_{10} + a_{01}H_{01})\boldsymbol{c}_2$$

$$+(b_{00}H_{10} + b_{10}H_{20} + b_{01}H_{11} - a_{00}H_{01} - a_{10}H_{11} - a_{01}H_{02})\boldsymbol{c}_3. \tag{A.14}$$

A.4 Integral solutions

The H integrals are solved by switching to the polar coordinates presented in the previous section

$$H_{mn} = \int_{S_1} \frac{u^m v^n du dv}{r^3} = \int_{\phi_1}^{\phi_2} \int_0^{\rho} \frac{(\cos^m \phi)(\sin^n \phi)(\rho^{m+n+1}) d\rho d\phi}{(\rho^2 + h^2)^{3/2}}. \tag{A.15}$$

The inner integral is solved first. There are three cases corresponding to ρ, ρ^2, and ρ^3

$$\int_0^\rho \frac{\rho d\rho}{(\rho^2+h^2)^{3/2}} = \frac{\sqrt{\rho^2+h^2}-h}{h\sqrt{\rho^2+h^2}} \tag{A.16}$$

$$\int_0^\rho \frac{\rho^2 d\rho}{(\rho^2+h^2)^{3/2}} = \ln(\rho+\sqrt{\rho^2+h^2}) - \frac{\rho}{\sqrt{\rho^2+h^2}} - \ln h \tag{A.17}$$

$$\int_0^\rho \frac{\rho^3 d\rho}{(\rho^2+h^2)^{3/2}} = \frac{\rho^2+2h^2}{\sqrt{\rho^2+h^2}} \tag{A.18}$$

Next, the outer integrals are solved by switching coordinates one last time from ϕ back to a rectilinear coordinate l (see Figure A.3) with the relations

$$\cos\phi = \frac{a}{\sqrt{l^2+a^2}} \qquad \tan\phi = l/a \tag{A.19}$$

$$\sin\phi = \frac{l}{\sqrt{l^2+a^2}} \qquad d\phi = \frac{a dl}{\sqrt{l^2+a^2}} \tag{A.20}$$

The solutions are

$$H_{00} = \frac{1}{h}\tan^{-1}\left(\frac{al}{a^2+h^2+h\sqrt{l^2+a^2+h^2}}\right)\Bigg|_{l_1}^{l_2}$$

$$H_{10} = \frac{l}{\sqrt{l^2+a^2}}\ln\left(\sqrt{l^2+a^2+h^2}+\sqrt{l^2+a^2}\right) - \ln\left(l+\sqrt{l^2+a^2+h^2}\right) - \frac{l\ln h}{\sqrt{l^2+a^2}}\Bigg|_{l_1}^{l_2}$$

$$H_{01} = \frac{a}{\sqrt{l^2+a^2}}\left[\ln h - \ln\left(\sqrt{l^2+a^2+h^2}+\sqrt{l^2+a^2}\right)\right]\Bigg|_{l_1}^{l_2}$$

$$H_{20} = \frac{al\left(l^2+a^2+h^2-h\sqrt{l^2+a^2+h^2}\right)}{(l^2+a^2)\sqrt{l^2+a^2+h^2}} - h\tan^{-1}\frac{l}{a} + h\tan^{-1}\left(\frac{hl}{a\sqrt{l^2+a^2+h^2}}\right)\Bigg|_{l_1}^{l_2}$$

$$H_{11} = \frac{-a^2\left(l^2+a^2+h^2-h\sqrt{l^2+a^2+h^2}\right)}{(l^2+a^2)\sqrt{l^2+a^2+h^2}}\Bigg|_{l_1}^{l_2}$$

$$H_{02} = -h\tan^{-1}\frac{l}{a} + h\tan^{-1}\left(\frac{hl}{a\sqrt{l^2+a^2+h^2}}\right)\Bigg|_{l_1}^{l_2}$$

APPENDIX B

TREECODE DESCRIPTION AND PERFORMANCE

A treecode is an algorithm used to efficiently approximate the influence of a set of particles, and it proceeds in two major steps: first, a tree is constructed over a set of particles, and then it is applied to a set of influence points. A *tree* is a hierarchy of *cells*, starting with the *root* cell. In an octree, which is used for three-dimensional space, each cell can have up to eight child cells, which, in turn, can also each have up to eight child cells, etc. Parent cells are *internal* to the tree, whereas cells without any children are *external*, and these external cells contain the particles. In keeping with the botany analogy, external cells are also called *leaves*.

In this thesis, we construct a tree in the same manner as Barnes and Hut [122], with the exception that we allow N_{max} particles per external cell instead of only one. The instructions for constructing a tree are presented in Algorithm B.1. The steps that are serial in nature are performed on the central processing unit (CPU), whereas the steps that are parallel in nature are performed on the graphics processing unit (GPU).

In Step 3 of Algorithm B.1, the particle moments are calculated by the Cartesian Taylor series expansion of the Biot–Savart law, Equation 2.65, as described by Lindsay [128]. The order of the Taylor series approximation, p, requires $p(p+1)(p+2)/6$ moment coefficients to be calculated for each cell; thus, p must be chosen carefully.

Then, as described in Algorithm B.3, the influence at a point $\boldsymbol{p}$ is calculated by starting at the root and traversing the tree, until a cell is considered sufficiently far from $\boldsymbol{p}$, or until an external cell is reached and the exact particle–point calculations are used. A point is considered sufficiently far from a cell if the following inequality is true:

$$L < \theta d, \tag{B.1}$$

where L is the characteristic length of the cell, and d is the distance from the center of the cell to the point. Therefore, higher values of θ encourage more approximations, whereas lower values are more conservative.

Algorithm B.1 Construct a tree

1. **Root Cell:** The root cell is the first cell of the tree from which the first set of children are created. The root cell is the entry point into the tree as it is stored as a linked list. Create the root cell as a cube slightly larger than the extent of all of the particles.
2. **Create Tree:** Loop through every particle and place them in the root cell with Algorithm B.2. This will create the tree structure as cells fill with particles and split into children.
3. **Particle Moments:** Loop over every particle and calculate the moment relative to the origin. Perform this computation on the GPU and copy the result back to the CPU.
4. **Aggregate Moments:** Aggregate all the moments of the children of each cell, starting with the root cell. Descend down the tree until the external cells are reached, and then recursively wind back up to the root cell.
5. **Copy:** Copy the tree from the CPU memory to the GPU memory.
6. **Translate Moments:** Translate the moments of each cell from the origin to the cell centers. Perform this calculation on the GPU.

An example two-dimensional quadtree is illustrated in Figure B.1a to demonstrate the structure of a tree. In this example, the maximum number of particles permitted in a cell is $N_{\text{max}} = 2$. Note that external cells are not all of the same size, and some contain fewer than N_{max} particles. The application of this example tree to a particular point is demonstrated

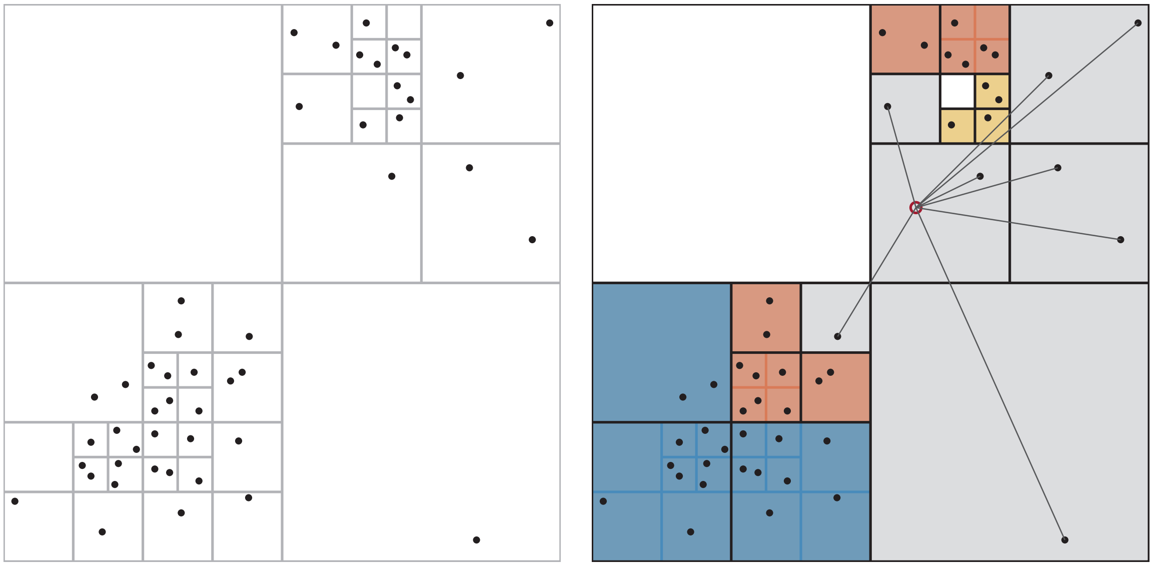
Algorithm B.2 Put particle P into cell C

```
if $C$ is external then
   if $C$ is not over capacity then
      $P$ is now in $C$
   else
      Divide $C$ into eight child octants, which are empty and external cells
      Put $P$ and any pre-existing particles in $C$ into the appropriate children of $C$
      $C$ is now an internal cell
   end if
else
   Put $P$ into the appropriate child of $C$
end if
```

Algorithm B.3 Calculate $\boldsymbol{v}$, the influence of cell C at point $\boldsymbol{p}$

```
if $C$ is internal then
   if $\boldsymbol{p}$ is sufficiently far from $C$ then
      calculate $\boldsymbol{v}$ as the point–cluster interaction from $C$ on $\boldsymbol{p}$
   else
      for each octant child $O$ of C do
         calculate the influence of $O$ at point $\boldsymbol{p}$
      end for
   end if
else
   for each particle $Q$ in $C$ do
      $\boldsymbol{v} = \boldsymbol{v} +$ the direct influence of $Q$ on $\boldsymbol{p}$
   end for
end if
```

in Figure B.1b. The cells in which an approximation is used are colored according to their size. The gray cells are closer than the approximation threshold, and, therefore, the exact particle–point influence is used for the particles within them. In this example, the influence of 48 particles is calculated by 16 direct particle summations and 11 cluster approximations.



(a) Quadtree structure for a set of particles.

(b) Approximate calculation of velocity at a point.

Figure B.1: Example quadtree.

To determine appropriate values for the parameters N_{max}, p, and θ that efficiently balance cost and accuracy, a set of experiments were conducted for a full factorial table with:

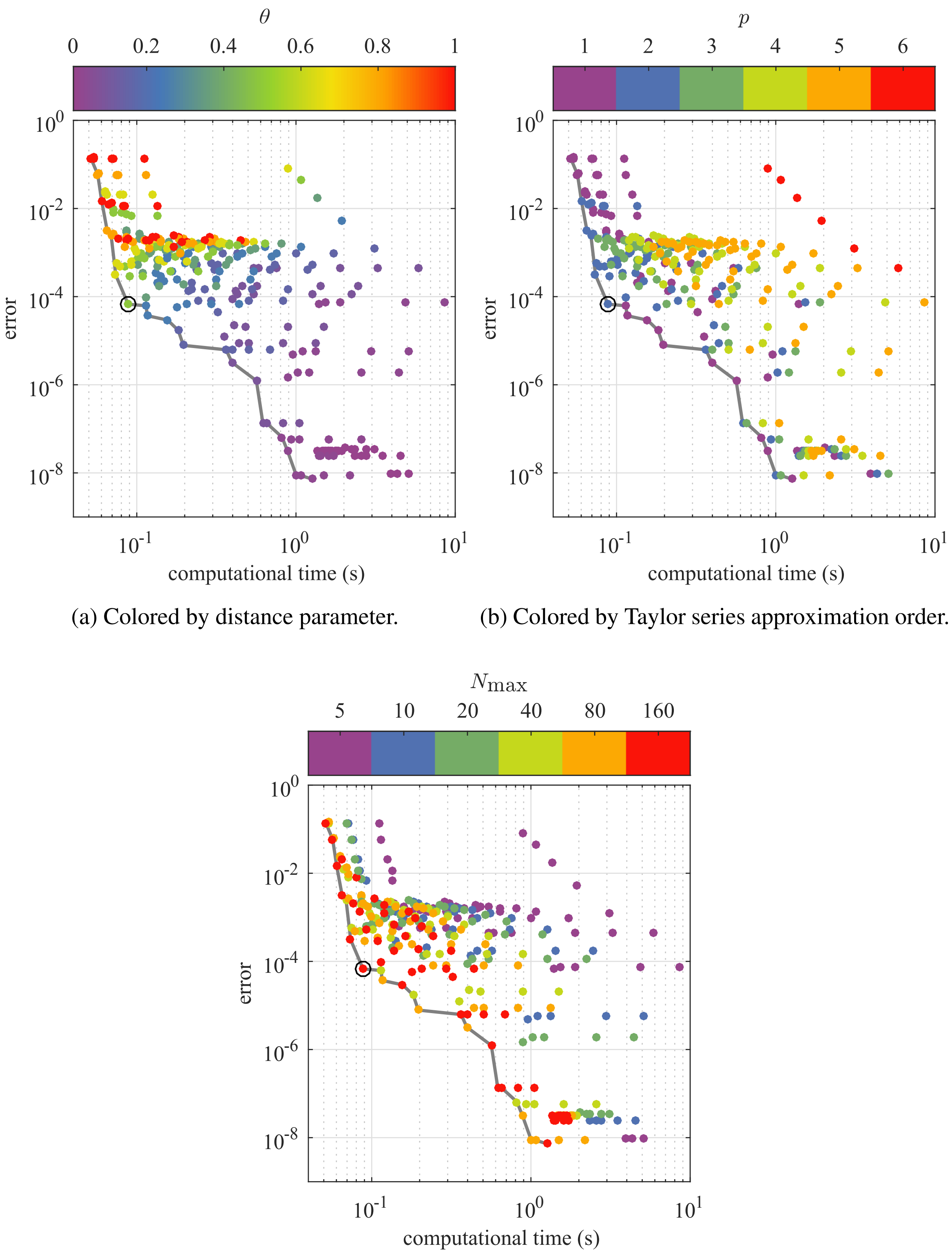
$$N_{\text{max}} = 5, 10, 20, 40, 80, 160 \tag{B.2}$$

$$p = 1, 2, 3, 4, 5, 6 \tag{B.3}$$

$$\theta = 0.01, 0.04, 0.09, \ldots, 0.81, 1.00 \tag{B.4}$$

The results are presented in Figures B.2a–B.2c, in which each figure presents the computational time and an aggregate error metric for a set of $750,000$ particles arranged in a torus.

The separate figures are each colored by a different parameter to reveal their independent effects. The θ parameter closely follows the cost–accuracy Pareto frontier (the gray line), while p and N_{max} trend in opposite directions. The parameter values chosen for all applications of the treecode in this thesis are $N_{\text{max}} = 160$, $p = 2$, and $\theta = 0.5$, as indicated by the black circle.



(a) Colored by distance parameter.

(b) Colored by Taylor series approximation order.

(c) Colored by leaf cell capacity.

Figure B.2: Results for the treecode parameter experiments colored by each parameter.

APPENDIX C

POTENTIAL FLOW AROUND AN ELLIPSOID

The solution for the potential flow around an ellipsoid originates with Lamb [66], and was popularized by Munk [170]. This section is a summary of the derivation presented by Clarke [171]. To begin, consider a tri-axial ellipsoid with semi-principal axes lengths a, b, and c and assume without loss of generality that $a > b > c$. The ellipsoid is described in Cartesian coordinates (x, y, z) with basis $(\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3)$ as

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1. \tag{C.1}$$

The ellipsoid can also be described by confocal ellipsoidal coordinates (λ, μ, ν) which relate to the Cartesian coordinates as

$$x^2 = \frac{(a^2+\lambda)(a^2+\mu)(a^2+\nu)}{(a^2-b^2)(a^2-c^2)}, \tag{C.2}$$

$$y^2 = \frac{(b^2+\lambda)(b^2+\mu)(b^2+\nu)}{(b^2-c^2)(b^2-a^2)}, \tag{C.3}$$

$$z^2 = \frac{(c^2+\lambda)(c^2+\mu)(c^2+\nu)}{(c^2-a^2)(c^2-b^2)}, \tag{C.4}$$

where $-a^2 < \nu < -b^2 < \mu < -c^2 < \lambda < \infty$, and the ellipsoid is obtained when $\lambda = 1$. The outward normal vector on the ellipsoid surface is generated by the partial derivatives of the coordinates (x, y, z) with respect to λ

$$\frac{\partial x}{\partial \lambda} = \frac{x}{2(a^2+\lambda)}, \tag{C.5}$$

$$\frac{\partial y}{\partial \lambda} = \frac{y}{2(b^2+\lambda)}, \tag{C.6}$$

$$\frac{\partial z}{\partial \lambda} = \frac{z}{2(c^2+\lambda)}. \tag{C.7}$$

The normal vector $\boldsymbol{n}$ is

$$\boldsymbol{n} = \frac{1}{h_\lambda}\left(\frac{\partial x}{\partial \lambda}\boldsymbol{b}_1 + \frac{\partial y}{\partial \lambda}\boldsymbol{b}_2 + \frac{\partial z}{\partial \lambda}\boldsymbol{b}_3\right), \tag{C.8}$$

where

$$h_\lambda = \frac{1}{4}\left(\frac{x^2}{(a^2+\lambda)^2} + \frac{y^2}{(b^2+\lambda)^2} + \frac{z^2}{(c^2+\lambda)^2}\right). \tag{C.9}$$

Also, the parameters α_0, β_0, and γ_0 are computed with the following elliptic integrals

$$\alpha_0 = abc\int_0^\infty \frac{d\lambda}{(a^2+\lambda)\sqrt{(a^2+\lambda)(b^2+\lambda)(c^2+\lambda)}}, \tag{C.10}$$

$$\beta_0 = abc\int_0^\infty \frac{d\lambda}{(a^2+\lambda)\sqrt{(b^2+\lambda)(b^2+\lambda)(c^2+\lambda)}}, \tag{C.11}$$

$$\gamma_0 = abc\int_0^\infty \frac{d\lambda}{(a^2+\lambda)\sqrt{(c^2+\lambda)(b^2+\lambda)(c^2+\lambda)}}. \tag{C.12}$$

Next, consider a freestream with velocity $\boldsymbol{v}_\infty$ defined component-wise with respect to basis $(\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3)$ as

$$\boldsymbol{v}_\infty = U_\infty \boldsymbol{b}_1 + V_\infty \boldsymbol{b}_2 + W_\infty \boldsymbol{b}_3. \tag{C.13}$$

Finally, the velocity $\boldsymbol{v}$ at any point on the ellipsoid surface is given in terms of the surface normal vector

$$\boldsymbol{v}_{U_\infty} = \frac{2U_\infty}{2-\alpha_0}\left[\boldsymbol{b}_1 - (\boldsymbol{b}_1 \cdot \boldsymbol{n})\boldsymbol{n}\right], \tag{C.14}$$

$$\boldsymbol{v}_{V_\infty} = \frac{2V_\infty}{2-\beta_0}\left[\boldsymbol{b}_2 - (\boldsymbol{b}_2 \cdot \boldsymbol{n})\boldsymbol{n}\right], \tag{C.15}$$

$$\boldsymbol{v}_{W_\infty} = \frac{2W_\infty}{2-\gamma_0}\left[\boldsymbol{b}_3 - (\boldsymbol{b}_3 \cdot \boldsymbol{n})\boldsymbol{n}\right], \tag{C.16}$$

$$\boldsymbol{v} = \boldsymbol{v}_{U_\infty} + \boldsymbol{v}_{V_\infty} + \boldsymbol{v}_{W_\infty}. \tag{C.17}$$

REFERENCES

[1] G.-H. Cottet and P. D. Koumoutsakos, *Vortex Methods: Theory and Practice*. Cambridge University Press, 2000.

[2] J. R. Mansfield, O. M. Knio, and C. Meneveau, “Dynamic LES of Colliding Vortex Rings Using a 3D Vortex Method,” *Journal of Computational Physics*, vol. 152, no. 1, pp. 305–345, 1999.

[3] S. M. Belotserkovskii and M. I. Nisht, “Nonstationary Nonlinear Theory of a Thin Wing of Arbitrary Planform,” *Fluid Dynamics*, vol. 9, no. 4, pp. 583–589, 1974.

[4] J. Katz, “Lateral Aerodynamics of Delta Wings with Leading-Edge Separation,” *AIAA Journal*, vol. 22, no. 3, pp. 323–328, 1984.

[5] C. P. Mracek, “A Vortex Panel Method for Potential Flows with Applications to Dynamics and Control,” PhD thesis, Virginia Polytechnic Institute and State University, 1988.

[6] Z. Mittelman and I. Kroo, “Unsteady Aerodynamics and Control of Delta Wings with Tangential Leading-Edge Blowing,” in *16th Atmospheric Flight Mechanics Conference*, AIAA, Boston, MA, Aug. 1989.

[7] T. Sarpkaya, “Computational Methods with Vortices—The 1988 Freeman Scholar Lecture,” *Journal of Fluids Engineering*, vol. 111, no. 1, pp. 5–52, 1989.

[8] D. Mook and A. Nayfeh, “Numerical Simulations of Dynamic/Aerodynamic Interactions,” *Computing in Science & Engineering*, vol. 1, no. 2, pp. 461–482, 1990.

[9] S. C. Smith and I. M. Kroo, “Computation of Induced Drag for Elliptical and Crescent-shaped Wings,” *Journal of Aircraft*, vol. 30, no. 4, pp. 446 –452, 1993.

[10] S. C. Smith and I. M. Kroo, “Induced Drag Computations on Wings with Accurately Modeled Wakes,” *Journal of Aircraft*, vol. 34, no. 2, pp. 253 –255, 1997.

[11] S. Preidikman, “Numerical Simulations of Interactions among Aerodynamics, Structural Dynamics, and Control Systems,” PhD thesis, Virginia Polytechnic Institute and State University, 1998.

[12] G. Bramesfeld, “A Higher Order Vortex-Lattice Method with a Force-Free Wake,” PhD thesis, Pennsylvania State University, 2006.

[13] D. L. Ashby, M. R. Dudley, S. K. Iguchi, L. Browne, and J. Katz, "Potential Flow Theory and Operation Guide for the Panel Code PMARC," National Aeronautics and Space Administration, Ames Research Center, Technical Memorandum TM-102851, Jan. 1991.

[14] J. Katz and A. Plotkin, *Low-Speed Aerodynamics*, Second. Cambridge University Press, 2001.

[15] R. A. Roccia, S. Preidikman, and J. C. Massa, "Modified Unsteady Vortex-Lattice Method to Study Flapping Wings in Hover Flight," *AIAA Journal*, 2013.

[16] D. Maniaci, "Wind turbine design using a free-wake vortex method with winglet application," PhD thesis, The Pennsylvania State University, 2013.

[17] A. Dhruv, C. J. Blower, and A. M. Wickenheiser, "A Three Dimensional Iterative Panel Method for Bio-Inspired Multi-Body Wings," in *Proceedings of the ASME 2014 Smart Materials, Adaptive Structures and Intelligent Systems*, ASME, Newport, RI, Sep. 2014.

[18] R. Paul, J. Murua, and A. Gopalarathnam, "Unsteady and Post-Stall Aerodynamic Modeling for Flight Dynamics Simulation," in *AIAA SciTech*, AIAA, National Harbor, MD, Jan. 2014.

[19] H. Hesse and R. Palacios, "Reduced-Order Aeroelastic Models for Dynamics of Maneuvering Flexible Aircraft," *AIAA Journal*, vol. 52, no. 8, pp. 1717–1732, Aug. 2014.

[20] Y. Hirato, M. Shen, and S. Aggarwal, "Initiation of Leading-Edge-Vortex Formation on Finite Wings in Unsteady Flow," in *AIAA SciTech*, AIAA, Kissimmee, FL, Jan. 2015.

[21] T. R. Quackenbush, A. H. Boschitsch, D. A. Wachspress, and K. Chua, "Rotor Design Optimization Using a Free Wake Analysis," National Aeronautics and Space Administration, Tech. Rep., 1993.

[22] D. A. Wachspress, T. R. Quackenbush, and A. H. Boschitsch, "First-Principles Free-Vortex Wake Analysis for Helicopters and Tiltrotors," in *American Helicopter Society 59th Annual Forum*, American Helicopter Society, vol. 59, Phoenix, AZ, May 2003, pp. 1763–1786.

[23] M. J. Bhagwat and J. G. Leishman, "Time–Accurate Modeling of Rotor Wakes Using a Free-Vortex Wake Method," in *18th AIAA Applied Aerodynamics Conference*, Denver, CO, 2000.

[24] S. Y. Wie, S. Lee, and D. J. Lee, “Potential Panel and Time-Marching Free-Wake-Coupling Analysis for Helicopter Rotor,” *Journal of Aircraft*, vol. 46, no. 3, pp. 1030–1041, 2009.

[25] H. Abedi, “Development of Vortex Filament Method for Aerodynamic Loads on Rotor Blades,” Master’s thesis, Chalmers University of Technology, 2013.

[26] S. C. Smith, “A Computational and Experimental Study of Nonlinear Aspects of Induced Drag,” PhD thesis, Stanford University, Jun. 1995.

[27] J. Conway and F. Tezok, “A Time-marching Scheme for the CANAERO Three-dimensional Vortex Sheets Panel Method,” in *38th Aerospace Sciences Meeting & Exhibit*, AIAA, Reno, Nevada, Jan. 2000.

[28] O. Levinski, “Prediction of Buffet Loads on Twin Vertical Tails Using a Vortex Method,” Defense Science & Technology Organization, Austrailia, Tech. Rep. DSTO-RR-0217, Jul. 2001.

[29] M. Roura, A Cuerva, A Sanz-Andrés, and A. Barrero-Gil, “A Panel Method Free-Wake Code for Aeroelastic Rotor Predictions,” *Wind Energy*, vol. 13, no. 4, pp. 357–371, 2010.

[30] J. Tan and H. Wang, “Simulating Unsteady Aerodynamics of Helicoptor Rotor with Panel/Viscous Vortex Particle Method,” *Aerospace Science and Technology*, vol. 30, pp. 255–268, 2013.

[31] J. P. Moore IV, “An Arbitrarily High-Order, Unstructured, Free-Wake Panel Solver,” Master’s thesis, Massachusetts Institute of Technology, Sep. 2013.

[32] C. R. Satterwhite, “Development of CPANEL, An Unstructured Panel Code, Using a Modified TLS Velocity Formulation,” Master’s thesis, California Polytechnic State University, San Luis Obispo, CA, Aug. 2015.

[33] C. Sousa, “Unsteady Panel Code Utilizing a Vortex Particle Wake,” Master’s thesis, California Polytechnic State University, San Luis Obispo, CA, Dec. 2016.

[34] N. Ramos-García, J. N. Sørensen, and W. Z. Shen, “Three-Dimensional Viscous-Inviscid Coupling Method for Wind Turbine Computations,” *Wind Energy*, vol. 19, no. 1, pp. 67–93, 2016.

[35] D. J. Willis, “An Unsteady, Accelerated, High Order Panel Method with Vortex Particle Wakes,” PhD thesis, Massachusetts Institute of Technology, Jun. 2006.

[36] D. J. Willis, J. Peraire, and J. K. White, “A Combined pFFT–Multipole Tree Code, Unsteady Panel Method with Vortex Particle Wakes,” *International Journal for Numerical Methods in Fluids*, vol. 53, pp. 1399–1422, 2007.

[37] S. G. Voutsinas, “Vortex Methods in Aeronautics: How to Make Things Work,” *International Journal of Computational Fluid Dynamics*, vol. 20, no. 1, pp. 3–18, 2006.

[38] J. S. Calabretta and R. A. McDonald, “A Three Dimensional Vortex Particle-Panel Method for Modeling Propulsion-Airframe Interaction,” in *48th AIAA Aerospace Sciences Meeting*, AIAA, Orlando, FL, 2010, pp. 1–16.

[39] E. H. Martin, “Assessment of Panel and Vortex Particle Methods for the Modelling of Stationary Propeller Wake Wash,” Master’s thesis, Memorial University of Newfoundland, 2015.

[40] J. R. Mansfield, O. M. Knio, and C. Meneveau, “Towards Lagrangian Large Vortex Simulation,” in *ESAIM: Proceedings*, EDP Sciences, vol. 1, 1996, pp. 49–64.

[41] D. G. Opoku, D. G. Triantos, F. Nitzsche, and S. G. Voutsinas, “Rotorcraft Aerodynamic and Aeroacoustic Modelling Using Vortex Particle Methods,” in *Proc. of the ICAS 2002 Congress, The International Council of the Aeronautical Sciences, Toronto, Canada*, 2002.

[42] J. Zhao and C. He, “A Viscous Vortex Particle Model for Rotor Wake and Interference Analysis,” in *American Helicopter Society 64th Annual Forum*, AHS, Montreal, Canada, May 2008.

[43] C. He and J. Zhao, “Modeling Rotor Wake Dynamics with Viscous Vortex Particle Method,” *AIAA Journal*, vol. 47, no. 4, pp. 902–915, 2009.

[44] J. Zhao and C. He, “A Viscous Vortex Particle Model for Rotor Wake and Interference Analysis,” *Journal of the American Helicopter Society*, vol. 55, no. 1, pp. 12 007–12 007, 2010.

[45] S. Park, Y. Chu, and D. Lee, “Investigation of Main Rotor-Tail Rotor Interaction in Hover Flight using Vortex Particle Method,” in *5th Asian/Australian Rotorcraft Forum*, Asian/Australian Rotorcraft Forum, 2016.

[46] M. J. Stock, A. Gharakhani, and C. P. Stone, “Modeling Rotor Wakes with a Hybrid OVERFLOW-Vortex Method on a GPU Cluster,” in *28th AIAA Applied Aerodynamics Conference*, AIAA, Chicago, IL, Jun. 2010.

[47] J. Zhao and C. He, “A Hybrid Solver with Combined CFD and Viscous Vortex Particle Method,” in *67th Annual Forum of the American Helicopter Society*, AHS, Virginia Beach, VA, May 2011.

[48] N. Rajmohan, J. Zhao, and C. He, “A Coupled Vortex Particle/CFD Methodology for Studying Coaxial Rotor Configurations,” in *Fifth Decennial AHS Aeromechanics Specialists Conference, San Francisco, California*, 2014.

[49] C. He and N. Rajmohan, “Modeling the Aerodynamic Interaction of Multiple Rotor Vehicles and Compound Rotorcraft with Viscous Vortex Particle Method,” in *American Helicopter Society 72nd Annual Forum*, AHS, West Palm Beach, FL, May 2016.

[50] F. Caradonna, “Developments and Challenges in Rotorcraft Aerodynamics,” in *38th Aerospace Sciences Meeting and Exhibit*, AIAA, Reno, NV, Jan. 2000, p. 109.

[51] R. C. Strawn and M. J. Djomehri, “Computational Modeling of Hovering Rotor and Wake Aerodynamics,” *Journal of Aircraft*, vol. 39, no. 5, pp. 786–793, 2002.

[52] R. M. Eshcol, C. Zhou, J. Kim, and L. N. Sankar, “A Comparative Study of Two Hover Prediction Methodologies,” in *54th AIAA Aerospace Sciences Meeting*, AIAA, San Diego, CA, Jan. 2016.

[53] R. C. Strawn, F. X. Caradonna, and E. P. Duque, “30 Years of Rotorcraft Computational Fluid Dynamics Research and Development,” *Journal of the American Helicopter Society*, vol. 51, no. 1, pp. 5–21, 2006.

[54] N. S. Hariharan, T. A. Egolf, and L. N. Sankar, “Simulation of Rotor in Hover: Current State, Challenges and Standardized Evaluation,” in *52nd Aerospace Sciences Meeting*, AIAA, National Harbor, MD, Jan. 2014.

[55] C. G. Speziale, “On the Advantages of the Vorticity–Velocity Formulation of the Equations of Fluid Dynamics,” *Journal of Computational Physics*, vol. 73, pp. 476–480, 1987.

[56] T. B. Gatski, “Review of Incompressible Fluid Flow Computations Using the Vorticity–Velocity Formulation,” *Applied Numerical Mathematics*, vol. 7, pp. 227–239, 1991.

[57] R. E. Brown, “Rotor Wake Modeling for Flight Dynamic Simulation of Helicopters,” *AIAA Journal*, vol. 38, no. 1, pp. 57–63, 2000.

[58] F. Scheurich and R. E. Brown, “Modelling the Aerodynamics of Vertical-axis Wind Turbines in Unsteady Wind Conditions,” *Wind Energy*, vol. 16, no. 1, pp. 91–107, 2013.

[59] G. R. Whitehouse and A. H. Boschitsch, “Innovative Grid-Based Vorticity–Velocity Solver for Analysis of Vorticity-Dominated Flows,” *AIAA Journal*, vol. 53, no. 6, pp. 1655–1669, Jun. 2015.

[60] M. J. Smith, R. Shenoy, A. R. Kenyon, and R. E. Brown, “Vorticity-transport and Unstructured RANS Investigation of Rotor–fuselage Interactions,” in *35th European Rotorcraft Forum*, 2009.

[61] G. R. Whitehouse, A. H. Boschitsch, M. J. Smith, C. E. Lynch, and R. E. Brown, “Investigation of Mixed Element Hybrid Grid-based CFD Methods for Rotorcraft Flow Analysis,” in *66th Annual Forum of the American Helicopter Society*, 2010.

[62] G. R. Whitehouse and H. Tadghighi, “Investigation of Hybrid Grid-Based CFD Methods for Rotorcraft Flow Analysis,” in *American Helicopter Society Aeromechanics Specialists Meeting*, Jan. 2010.

[63] M. J. Stock, “A Regularized Inviscid Vortex Sheet Method for Three Dimensional Flows with Density Interfaces,” PhD thesis, University of Michigan, 2006.

[64] H. Feng, “Vortex Sheet Simulations of 3-D Flows Using an Adaptive Triangular Panel/Particle Method,” PhD thesis, The University of Michigan, 2007.

[65] A. Leonard, “Vortex Methods for Flow Simulation,” *Journal of Computational Physics*, 1980.

[66] H. Lamb, *Hydrodynamics*. Cambridge University Press, 1895.

[67] M. J. Stock, W. J. Dahm, and G. Tryggvason, “Impact of a Vortex Ring on a Density Interface Using a Regularized Inviscid Vortex Sheet Method,” *Journal of Computational Physics*, vol. 227, pp. 9021–9043, 2008.

[68] H. von Helmholtz, “On Integrals of the hydrodynamical equations, which express vortex-motion,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 33, no. 226, pp. 485–512, 1867.

[69] G. K. Batchelor, *An Introduction to Fluid Dynamics*. Cambridge University Press, 1967.

[70] J. Casey and P. M. Naghdi, “On the Lagrangian Description of Vorticity,” *Archive for Rational Mechanics and Analysis*, vol. 115, pp. 1–14, 1991.

[71] S.-I. Sohn, “Two Vortex-Blob Regularization Models for Vortex Sheet Motion,” *Physics of Fluids*, vol. 26, 2014.

[72] C. Pozrikidis, "Theoretical and Computational Aspects of the Self-induced Motion of Three-Dimensional Vortex Sheets," *Journal of Fluid Mechanics*, vol. 425, pp. 335–366, 2000.

[73] T. R. Kane and D. A. Levinson, *Dynamics, Theory and Applications*. McGraw Hill, 1985.

[74] L. Rosenhead, "The Formation of Vortices from a Surface of Discontinuity," *Proceedings of the Royal Society of London*, vol. 134, no. 823, pp. 170–192, Nov. 1931.

[75] F. L. Westwater, "The Rolling Up of the Surface of Discontinuity behind an Aerofoil of Finite Span," Aeronautical Research Council, Tech. Rep. R&M 1692, Aug. 1935.

[76] K. Kuwahara and H. Takami, "Numerical Studies of Two-Dimensional Vortex Motion by a System of Point Vortices," *Journal of the Physical Society of Japan*, vol. 34, pp. 247–253, 1973.

[77] L. Prandtl, "Applications of Modern Hydrodynamics to Aeronautics," National Advisory Committee for Aeronautics, Technical Report 116, 1923.

[78] G. Birkhoff and J. Fisher, "Do Vortex Sheets Roll Up?" *Rendiconti del Circolo Matematico di Palermo*, vol. 8, no. 1, pp. 77–90, 1959.

[79] G. Birkhoff, "Helmholtz and Taylor Instability," in *Proceedings of Symposia in Applied Mathematics*, 1962, pp. 55–76.

[80] R. R. Clements and D. J. Maull, "The Representation of Sheets of Vorticity by Discrete Vortices," *Progress in Aerospace Sciences*, vol. 16, no. 2, pp. 129–146, 1975.

[81] P. T. Fink and W. K. Soh, "A New Approach to Roll-Up Calculations of Vortex Sheets," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 362, pp. 195–209, 1978.

[82] P. G. Saffman and G. R. Baker, "Vortex Interactions," *Annual Review of Fluid Mechanics*, vol. 11, pp. 95–122, 1979.

[83] G. R. Baker, "A Test of the Method of Fink and Soh for Following Vortex-Sheet Motion," *Journal of Fluid Mechanics*, vol. 100, pp. 209–220, 1980.

[84] R. Krasny, "A Study of Singularity Formation in a Vortex Sheet by the Point-Vortex Approximation," *Journal of Fluid Mechanics*, 1986.

[85] D. W. Moore, “On the Point Vortex Method,” *SIAM Journal on Scientific and Statistical Computing*, 1981.

[86] A. J. Chorin and P. S. Bernard, “Discretization of a Vortex Sheet with an Example of Roll-Up,” *Journal of Computational Physics*, vol. 13, pp. 423–429, 1973.

[87] C. W. Oseen, “Uber Wirbelbewegung in Einer Reibenden Flssigkeit,” *Arkiv fr Matematik, Astronomi och Fysik*, vol. 7, no. 14, pp. 14–21, 1911.

[88] N. Rott, “On the Viscous Core of a Line Vortex,” *Zeitschrift für Angewandte Mathematik und Physik (ZAMP)*, vol. 9, no. 5, pp. 543–553, 1958.

[89] A. M. Bloom and H. Jen, “Roll-Up of Aircraft Trailing Vortices using Artificial Viscosity,” *Journal of Aircraft*, vol. 11, no. 11, pp. 714–716, Nov. 1974.

[90] R. Krasny, “Desingularization of Periodic Vortex Sheet Roll-Up,” *Journal of Computational Physics*, vol. 65, pp. 292–313, 1986.

[91] C. R. Anderson, “A Vortex Method for Flows with Slight Density Variations,” *Journal of Computational Physics*, vol. 61, pp. 417–44, 1985.

[92] L Rosenhead, “The Spread of Vorticity in the Wake behind a Cylinder,” *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, pp. 590–612, 1930.

[93] R. Krasny, “Vortex Sheet Roll-Up Due to the Motion of a Flat Plate,” in *International Symposium on Nonsteady Fluid Dynamics*, J. A. Miller and D. P. Telionis, Eds., vol. 92, 1990.

[94] O. Hald and V. M. Del Prete, “Convergence of Vortex Methods for Euler’s Equations,” *Mathematics of Computation*, vol. 32, no. 143, pp. 791–809, 1978.

[95] O. H. Hald, “Convergence of Vortex Methods for Euler’s Equations. II.,” *SIAM Journal on Numerical Analysis*, 1979.

[96] R. Krasny, “Computation of Vortex Sheet Roll-Up in the Trefftz Plane,” *Journal of Fluid Mechanics*, 1987.

[97] G. Winckelmans, R. Cocle, L. Dufrense, and R. Capart, “Vortex Methods and their Applications to Trailing Wake Vortex Simulations,” *C. R. Physique*, vol. 6, pp. 467–486, 2005.

[98] G. Tryggvason, “Simulation of Vortex Sheet Roll-Up by Vortex Methods,” *Journal of Computational Physics*, 1989.

[99] D. W. Moore, “The Spontaneous Appearance of a Singularity in the Shape of an Evolving Vortex Sheet,” *Proceedings of the Royal Society of London*, vol. 365, pp. 105–119, 1979.

[100] M. J. Shelley, “A Study of Singularity Formation in Vortex-Sheet Motion by a Spectrally Accurate Vortex Method,” *Journal of Fluid Mechanics*, vol. 244, pp. 493–526, 1992.

[101] J. W. Rottman and P. K. Stansby, “On the ‘δ-Equations’ for Vortex Sheet Evolution,” *Journal of Fluid Mechanics*, vol. 247, pp. 527–549, 1993.

[102] S. J. Cowley, G. R. Baker, S. Tanveer, and M. Page, “On the Origin of Singularities in Vortex Sheet Motion,” in *26th AIAA Fluid Dynamics Conference*, AIAA, San Diego, CA, Jun. 1995.

[103] M. Nitsche, “Singularity Formation in a Cylindrical and a Spherical Vortex Sheet,” *Journal of Computational Physics*, vol. 173, no. 1, pp. 208–230, 2001.

[104] J. T. Beale and A. Majda, “Vortex Methods. I: Convergence in Three Dimensions,” *Mathematics of Computation*, vol. 39, no. 159, pp. 1–27, 1982.

[105] J. T. Beale, “On the Accuracy of Vortex Methods at Large Times,” in *IMA Workshop on Computational Fluid Dynamics and Reacting Gas Flows*, Springer–Verlag, 1988.

[106] G. S. Winckelmans and A. Leonard, “Contributions to Vortex Particle Methods for the Computation of Three-Dimensional Incompressible Unsteady Flows,” *Journal of Computational Physics*, vol. 109, no. 2, pp. 247–273, 1993.

[107] G.-H. Cottet, P. Koumoutsakos, and M. L. O. Salihi, “Vortex Methods with Spatially Varying Cores,” *Journal of Computational Physics*, vol. 162, no. 1, pp. 164–185, 2000.

[108] M. E. Agishtein and A. A. Migdal, “Dynamics of Vortex Surfaces in Three Dimensions Theory and Simulations,” *Physica D*, vol. 40, pp. 91–118, 1989.

[109] M. Brady, A. Leonard, and D. I. Pullin, “Regularized Vortex Sheet Evolution,” *Journal of Computational Physics*, vol. 146, pp. 520–545, 1998.

[110] K. Lindsay and R. Krasny, “A Particle Method and Adaptive Treecode for Vortex Sheet Motion in Three-Dimensional Flow,” *Journal of Computational Physics*, vol. 172, pp. 879–907, 2001.

[111] H. Feng, L. Kaganovskiy, and R. Krasny, "Azimuthal Instability of a Vortex Ring Computed by a Vortex Sheet Panel Method," *Fluid Dynamics Research*, vol. 41, 2009.

[112] O. A. Kandil, L.-C. Chu, and T. Tureaud, "A Nonlinear Hybrid Vortex Method for Wings at Large Angle of Attack," *AIAA Journal*, vol. 22, no. 3, pp. 329–336, 1984.

[113] L. Kaganovskiy, "Adaptive Panel Method for Particle Simulation of Three Dimensional Vortex Sheet Motion," PhD thesis, University of Michigan, 2006.

[114] L. Kaganovskiy, "Adaptive Panel Representation for 3D Vortex Ring Motion and Instability," *Mathematical Problems in Engineering*, vol. 2007, 2007.

[115] L. Kaganovskiy, "Adaptive Panel Representation for Oblique Collision of Two Vortex Rings," *International Journal of Non-Linear Mechanics*, vol. 46, no. 1, pp. 9–13, 2011.

[116] J. Jost, *Compact Riemann Surfaces: An Introduction to Contemporary Mathematics*. Springer Science & Business Media, 2006.

[117] G. Tryggvason, W. J. A. Dahm, and K. Sbeih, "Fine Structure of Vortex Sheet Rollup by Viscous and Inviscid Simulation," *Journal of Fluids Engineering*, vol. 113, pp. 31–36, Mar. 1991.

[118] M. Nitsche and R. Krasny, "A Numerical Study of Vortex Ring Formation at the Edge of a Circular Tube," *Journal of Fluid Mechanics*, vol. 276, pp. 139–161, 1994.

[119] H. Edelsbrunner, *Geometry and Topology for Mesh Generation*. Cambridge University Press, 2001.

[120] R. Diestel, *Graph Theory*. Springer-Verlag, 2000.

[121] A. W. Appel, "An Efficient Program for Many-body Simulation," *SIAM Journal on Scientific and Statistical Computing*, vol. 6, no. 1, pp. 85–103, Jan. 1985.

[122] J. Barnes and P. Hut, "A Hierarchical $O(N \log N)$ Force-calculation Algorithm," *Nature*, vol. 324, pp. 446–449, 1986.

[123] L. Greengard and V. Rokhlin, "A Fast Algorithm for Particle Simulations," *Journal of Computational Physics*, vol. 73, no. 2, pp. 325–348, 1987.

[124] F. Zhao, "An $O(N)$ Algorithm for Three-Dimensional N-Body Simulations," PhD thesis, Massachusetts Institute of Technology, 1987.

[125] J. K. Salmon and M. S. Warren, “Fast Parallel Tree Codes for Gravitational and Fluid Dynamical N-Body Problems,” *International Journal of High Performance Computing Applications*, vol. 8, no. 2, pp. 129–142, 1994.

[126] C. I. Draghicescu and M. Draghicescu, “A Fast Algorithm for Vortex Blob Interactions,” *Journal of Computational Physics*, vol. 116, no. 1, pp. 69–78, 1995.

[127] G. S. Winckelmans, J. K. Salmon, M. S. Warren, A. Leonard, and B. Jodoin, “Application of Fast Parallel and Sequential Tree Codes to Computing Three-Dimensional Flows with the Vortex Element and Boundary Element Methods,” in *ESAIM: Proceedings*, EDP Sciences, vol. 1, 1996, pp. 225–240.

[128] K. Lindsay, “A Three-Dimensional Cartesian Tree-Code and Applications to Vortex Sheet Roll-Up,” PhD thesis, University of Michigan, 1997.

[129] J. Board and K. Schulten, “The Fast Multipole Algorithm,” *Computing in Science & Engineering*, vol. 2, no. 1, pp. 76–79, 2000.

[130] M. J. Stock and A. Gharakhani, “Graphics Processing Unit-Accelerated Boundary Element Method and Vortex Particle Method,” *Journal of Aerospace Computing, Information, and Communication*, vol. 8, no. 7, pp. 224–236, 2011.

[131] J. Choi, A. Chandramowlishwaran, K. Madduri, and R. Vuduc, “A CPU–GPU Hybrid Implementation and Model-driven Scheduling of the Fast Multipole Method,” in *Proceedings of Workshop on General Purpose Processing Using GPUs*, ACM, 2014, p. 64.

[132] Q. Nie and G. Baker, “Application of Adaptive Quadrature to Axi-symmetric Vortex Sheet Motion,” *Journal of Computational Physics*, vol. 143, no. 1, pp. 49–69, 1998.

[133] A. M. O. Smith and J. Pierce, “Exact Solution of the Neumann Problem: Calculation of Non-circulatory Plane and Axially Symmetric Flows about Or Within Arbitrary Boundaries,” Douglas Aircraft Co., Long Beach, CA, Tech. Rep. E.S. 26988, Apr. 1958.

[134] A. M. O. Smith, “Incompressible Flow about Bodies of Arbitrary Shape,” in *IAS National Summer Meeting*, 1962.

[135] J. L. Hess and A. M. O. Smith, “Calculation of Nonlifting Potential Flow about Arbitrary Three-Dimensional Bodies,” *Journal of Ship Research*, vol. 8, no. 2, pp. 22–24, 1964.

[136] 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.

[137] J. L. Hess, “Calculation of Potential Flow About Arbitrary Three-Dimensional Lifting Bodies,” McDonnell Douglass Corporation, Long Beach, CA, Tech. Rep. MDC J5679-01, Oct. 1972.

[138] J. L. Hess, “Panel Methods in Computational Fluid Dynamics,” *Annual Review of Fluid Mechanics*, vol. 22, no. 1, pp. 255–274, 1990.

[139] A. M. O. Smith, “The Panel Method: Its Original Development,” in *Applied computational aerodynamics*, American Institute of Aeronautics and Astronautics, vol. 1, Washington, DC, 1990, pp. 3–17.

[140] O. D. Kellogg, *Foundations of Potential Theory*. Frederick Ungar Publishing Company, 1929.

[141] B. Maskew, “Program VSAERO Theory Document,” National Aeronautics and Space Administration, Tech. Rep., 1987.

[142] F. T. Johnson, “A General Panel Method for the Analysis and Design of Arbitrary Configurations in Incompressible Flows,” National Aeronautics and Space Administration, Contractor Report NASA-CR-3079, 1980.

[143] R. L. Carmichael and L. L. Erickson, “PANAIR—A Higher Order Panel Code for Predicting Subsonic or Supersonic Linear Potential Flow About Arbitrary Configurations,” in *14th Fluid and Plasma Dynamics Conference*, AIAA, Palo Alto, CA, Jun. 1981.

[144] E. Martensen, “The Calculation of the Pressure Distribution on a Cascade of thick Airfoils by means of Fredholm Integral Equations of the Second Kind,” National Aeronautics and Space Administration, Tech. Rep., 1971.

[145] W Prager, “The Pressure Distribution on Bodies in Plane Potential Flow,” *Physik. Z*, vol. 29, pp. 865–869, 1928.

[146] K. Jacob and F. W. Riegels, “The Calculation of the Pressure Distribution Over Aerofoil Sections of Finite Thickness with and without Flaps and Slats,” *Z. Flugwiss*, vol. 11, no. 9, pp. 357–367, 1963.

[147] D. H. Wilkinson, “Numerical Solution of the Analysis and Design Problems for the Flow Past One or More Aerofoils on Cascades,” Aeronautical Research Council, Tech. Rep. R&M 3545, 1967.

[148] D. H. Wilkinson, “A Numerical Solution of the Analysis and Design Problems for the Flow Past One or More Aerofoils or Cascades. Part 1- The Analysis Problem. Part 2- The Design Problem (numerical Solution of Analysis and Design Problems

for Flow Past One or More Airfoils or Cascades)," Aeronautical Research Council, Tech. Rep., 1968.

[149] R. I. Lewis and P. G. Ryan, "Surface Vorticity Theory for Axisymmetric Potential Flow Past Annular Aerofoils and Bodies of Revolution with Application to Ducted Propellers and Cowls," *Journal of Mechanical Engineering Science*, vol. 14, no. 4, pp. 280–296, 1972.

[150] V. Hill, "A Surface Vorticity Theory for Propeller Ducts and Turbofan Engine Cowls in Non-Axisymmetric Incompressible Flow," *Journal of Mechanical Engineering Science*, vol. 20, no. 4, pp. 201–219, 1978.

[151] M. Drela, *Flight Vehicle Aerodynamics*. MIT Press, 2014.

[152] H. Schlichting, *Boundary-Layer Theory*, Seventh, F. J. Cerra, Ed. McGraw-Hill Book Company, 1976.

[153] L. Prandtl and O. G. Tietjens, *Fundamentals of Hydro- and Aeromechanics*, H. W. Craver, Ed. Dover, 1957.

[154] W. M. Kutta, "Uber eine mit den grundlagen des flug problems in beziehung stehende zweidimensionale strömung," 1910.

[155] J. L. Hess, "The Problem of Three-Dimensional Lifting Potential Flow and Its Solution by Means of Surface Singularity Distribution," *Computer Methods in Applied Mechanics and Engineering*, vol. 4, no. 3, pp. 283–319, 1974.

[156] J. Nocedal and S. J. Wright, *Numerical Optimization*, P. Glynn and S. M. Robinson, Eds. Springer, 1999.

[157] K. W. McAlister and R. K. Takahashi, "NACA0015 Wing Pressure and Trailing Vortex Measurements," National Aeronautics and Space Administration, Ames Research Center, Moffett Field, California, Technical Paper NASA TP-3151, Nov. 1991.

[158] R. Krasny and M. Nitsche, "The Onset of Chaos in Vortex Sheet Flow," *Journal of Fluid Mechanics*, vol. 454, pp. 47–69, 2002.

[159] M. Hand, D. Simms, L. Fingersh, D. Jager, J. Cotrell, S Schreck, and S. Larwood, "Unsteady Aerodynamics Experiment Phase VI: Wind Tunnel Test Configurations and Available Data Campaigns," National Renewable Energy Laboratory, Golden, CO, Technical Report NREL/TP-500-29955, 2001.

[160] C. E. Lynch, D. T. Prosser, and M. J. Smith, "An Efficient Actuating Blade Model for Unsteady Rotating System Wake Simulations," *Computers & Fluids*, vol. 92, pp. 138–150, 2014.

[161] D. Simms, S. Schreck, M. Hand, and L. J. Fingersh, "NREL Unsteady Aerodynamics Experiment in the Nasa-Ames Wind Tunnel: A Comparison of Predictions to Measurements," National Renewable Energy Laboratory, Golden, CO, Technical Report NREL/TP-500-29494, Jun. 2001.

[162] A. S. Hahn, "Vehicle Sketch Pad: A Parametric Geometry Modeler for Conceptual Aircraft Design," in *48th AIAA Aerospace Sciences Meeting*, AIAA, Orlando, FL, Jan. 2010.

[163] K. Hayami, J.-F. Yin, and T. Ito, "GMRES Methods for Least Squares Problems," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 5, pp. 2400–2430, 2010.

[164] Y. Saad and M. H. Schultz, "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 7, no. 3, pp. 856–869, 1986.

[165] D. C.-L. Fong and M. Saunders, "LSMR: An Iterative Algorithm for Sparse Least-Squares Problems," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2950–2971, 2011.

[166] M. J. Smith, "A Fourth Order Euler/navier–stokes Prediction Method for the Aerodynamics and Aeroelasticity of Hovering Rotor Blades," PhD thesis, Georgia Institute of Technology, Feb. 1994.

[167] J. A. Blackwell, Jr., "Numerical Methods to Calculate the Induced Drag or Optimal Span Loading for Arbitrary Non-planer Aircraft," Lockheed-Georgia Company, Tech. Rep., 1976.

[168] D. J. Pate and B. J. German, "Improved Computation of Induced Drag for Wakes of Arbitrary Shape," in *AIAA Aviation*, AIAA, Los Angeles, CA, Aug. 2013.

[169] J.-C. Suh, "The Evaluation of the Biot–Savart Integral," *Journal of Engineering Mathematics*, vol. 37, pp. 375–395, 2000.

[170] M. M. Munk, "Remarks on the Pressure Distribution over the Surface of an Ellipsoid, Moving Translationally Through a Perfect Fluid," National Advisory Committee for Aeronautics, Technical Note NACA-TN-196, Jun. 1924.

[171] D. B. Clarke, “Experimental and Computational Investigation of Flow about Low Aspect Ratio Ellipsoids at Transcritical Reynolds Numbers,” PhD thesis, University of Tasmania, 2009.