Title

Fast models and reinforcement learning control of unsteady aerodynamics

Permalink

https://escholarship.org/uc/item/21r4z6xk

Author

Beckers, Diederik

Publication Date

2023

UNIVERSITY OF CALIFORNIA

Los Angeles

Fast models and reinforcement learning control of unsteady aerodynamics

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

by

Diederik Frank H Beckers

2023

ABSTRACT OF THE DISSERTATION

Fast models and reinforcement learning control of unsteady aerodynamics

by

Diederik Frank H Beckers
Doctor of Philosophy in Mechanical Engineering
University of California, Los Angeles, 2023
Professor Jeffrey D. Eldredge, Chair

Large amplitude flow disturbances and gusts can drastically alter the aerodynamic forces on airfoils or structures. The modeling and control of the aerodynamic response of the flow around the body is complicated by the inherent nonlinearities and high dimensionality of this system, which increase the cost and complexity of the numerical tools that enable this. This work introduces two numerical tools for the efficient modeling of unsteady aerodynamic flows and explores the use of deep reinforcement learning to perform airfoil pitch control during flow disturbances. The first numerical tool we introduce is a grid-based potential flow solver. We focus on the discrete streamfunction and model the flow around bodies and point vortices immersed in a cartesian grid. This tool reformulates the inviscid vortex-in-cell method as a saddle-point problem where constraints such as the no-flow-through and Kutta condition can be added as Lagrange multipliers. The second numerical tool uses a potential flow to model wind tunnel walls and irrotational wind tunnel gust generators in a viscous flow simulation. This technique allows us to still accurately model the flow around the test subject while accounting for the effects of the wind tunnel in a way that doesn't drastically increase the cost of the simulation. Lastly, in our explorative study of

reinforcement learning of airfoil pitch control, we try out the training of a control policy in a classical unsteady aerodynamics environment and a viscous, low-Reynolds number flow environment to minimize the lift variations caused by flow disturbances and compare the performance of controllers, or agents, that observe different types of information about the states of the system.

The dissertation of Diederik Frank H Beckers is approved.

Christopher R. Anderson

Tetsuya Iwasaki

Kunihiko Taira

Jeffrey D. Eldredge, Committee Chair

University of California, Los Angeles

2023

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

This work was only possible with the support of many individuals and institutions.

First and foremost, I would like to thank my advisor, Professor Jeff Eldredge. His invaluable advice and expertful insights helped me become a better scholar and communicator. Throughout my academic journey, he not only ensured my confidence and well-being but also readied me for a future career as a researcher. I could not have wished for a better advisor, and I am immensely grateful to him for giving me the opportunity five years ago to join his lab.

I also want to thank my committee members Professor Sam Taira, Professor teD Iwasaki, and Professor Chris Anderson for their comments and insights. Their excellence in their fields and dedication to their teaching inspired me throughout my time at UCLA and will continue to do so long after. The work in this thesis relies in many ways on the contributions they made to fluids, control, and mathematics and the courses they teach.

To my colleagues and peers, your shared enthusiasm and insightful conversations have greatly helped me during my studies. I would specifically like to thank the members of the SOFIA lab: Yuxuan, Sanjay, Hanieh, Mathieu, Yi-Jui, Rhuizhi, Wei, Albert, and Nikhil, and the members of the Taira lab who I had to pleasure of interacting with: Jean, Vedasri, and Kai.

I am thankful for the financial support and resources provided by the UCLA Graduate Division, the UCLA Mechanical and Aerospace Engineering Department, the UCLA Institute for Digital Research and Education, and the Air Force Office of Scientific Research, which enabled me to conduct this research. I also want thank the National Renewable Energy Laboratory for the opportunity to intern during the Summers of 2021 and 2022 and Mike Brazell and Marc Henry de Frahan for their excellent mentorship.

I want to thank my friends in Belgium and the United States, whose (remote) presence and visits not only added joy to every moment but also served as a source of stress relief and support, offering a willing ear whenever I needed it.

Lastly, I want to thank my partner, Nissryne, my parents, and my siblings for their unconditional love and support. You have always been there for me, especially during the most difficult moments. I am so lucky to have you.

VITA

2011–2014	B.S. (Mechanical Engineering), KU Leuven.
2014–2016	M.S. (Mechanical Engineering), KU Leuven.
2016–2017	Research Master (Turbomachinery and Propulsion), von Karman Institute for Fluid Dynamics.

PUBLICATIONS

Beckers, D., & Eldredge, J. D. (2023). Wind tunnel effects on gust-interaction simulations. *Theoretical and Computational Fluid Dynamics*. doi:10.1007/s00162-023-00668-9

Beckers, D., & Eldredge, J. D. (2022). Planar potential flow on Cartesian grids. *Journal of Fluid Mechanics*, 941, A19. doi:10.1017/jfm.2022.238

CHAPTER 1

Introduction

1.1 Background and motivation

A key aspect in the design and operation of vehicles, machines, and structures that interact with air flows is their behavior under varying ambient or operating conditions and the regulation of the aerodynamic forces acting on them. An evident example is the control of actuators on an aircraft wing or body to maintain a constant lift force when flying through gusty air. Another example is the pitching of entire wind turbine blades to react to changes in the incoming air to maintain optimal efficiency or, in extreme wind conditions, to align the blades with the wind to shut down the turbine such that its operating limits are not exceeded. These cases, and many others, are examples of flow control, where the control manipulates the flow, and the control objective depends on the flow behavior, e.g., transition delay, separation prevention, drag reduction, lift enhancement, etc. [Gad-el Hak, 2000]

Flow control lies at the intersection of fluid mechanics and control theory, and the performance of flow control systems is often limited by three major challenges introduced by these fields. Firstly, the fluid equations are nonlinear, which gives rise to intricate flow patterns spanning multiple scales and input-output behavior that cannot be superposed. Secondly, these patterns are inherently three-dimensional, and accurately modeling their behavior up to the smallest scales with direct numerical simulations is extremely expensive for most interesting flow problems. Finally, most control theory tools are developed for linear systems, including linear quadratic regulators, robust control, root locus analysis, and Bode design techniques.

Despite these challenges, numerous highly effective solutions exist that solve complex flow control problems thanks to many advances in both fields. Still, the key to pushing the envelope of many systems is the ongoing development of appropriate techniques to handle the challenges above by reducing the modeling cost while still accounting for the most important flow features and considering or exploiting the nonlinearities in the flow when designing control laws. These efforts are pivotal in unlocking enhanced performance across various systems, such as the supermaneuverability of air vehicles, safety during extreme conditions, and increased efficiency of energy-conversion systems and transportation, resulting in lower operating costs and reduced emissions.

This dissertation focuses on developing and analyzing methods that align with these ongoing efforts dealing with the flow control challenges and does so in the framework of two-dimensional, incompressible, unsteady aerodynamics. Specifically, the contributions of this work are:

- A fast method that relies on vortex modeling on a grid to reduce the cost of fluid simulations while still accounting for the dominant flow behavior,
- A numerical method to efficiently account for wind-tunnel walls in the modeling of experiments and an accompanying study investigating the influences of those walls on gust-airfoil interactions, and
- An exploratory study of using deep reinforcement learning to control the pitch of an airfoil under unsteady conditions in a nonlinear way.

1.2 Dissertation outline

In Chapter 2, we describe the unsteady aerodynamics flows we will deal with in the work and introduce several modeling approaches: classical unsteady aerodynamics, dynamic wake inviscid vortex methods, and a viscous flow model. These will be useful in the later chapters of this work.

Chapter 3 presents a grid-based treatment of potential flows in two dimensions and its use in an inviscid vortex model for simulating unsteady aerodynamic flows. For flows consisting of vortex elements, the treatment follows the vortex-in-cell approach and solves the streamfunction-vorticity Poisson equation on a Cartesian grid after transferring the circulation from the vortices onto the grid.

The aerodynamic response to flow disturbances is regularly investigated through wind-tunnel (or water-tunnel) experiments. The gusts generated in those experiments are often further analyzed using numerical simulations, but usually without fully accounting for the wind-tunnel walls or gust generator. In Chapter 4, we investigate wind-tunnel effects on the predicted lift response and flow field in unsteady aerodynamics using a computational framework that models the viscous flow around the airfoil but treats the tunnel walls and gust generation as inviscid boundary conditions. We apply this model to three examples and compare the predicted gust response with the responses predicted by a free-space viscous model and a classical unsteady aerodynamics model to highlight the wind-tunnel effects.

In Chapter 5, we use deep reinforcement learning to train a controller, or policy, to control the pitch of the airfoil to minimize the lift variations caused by disturbances. We train this policy in two types of environments: a classical unsteady aerodynamics environment and a viscous flow environment. The former is a linear model, and linear control theory can easily solve the control problem. However, the latter environment applies disturbances that induce a nonlinear aerodynamic response. This warrants the use of nonlinear control, such as deep reinforcement learning policies, which we explore in this chapter.

Finally, in Chapter 6, we summarize the tools that this work introduced and reiterate the main findings. We then conclude by providing an overview of possible future research directions.

CHAPTER 2

Unsteady aerodynamics: phenomena and modeling approaches

2.1 Flow description and variables

This work is mostly concerned with two-dimensional, external, unsteady, low Mach number viscous or inviscid fluid flows over airfoils. The main domain of application of these flows is aerodynamics. However, most methods we introduce are equally applicable to hydrodynamics or the study of other fluids.

We will assume the flows in this work have a uniform density ρ and uniform viscosity μ, or zero viscosity in the case of inviscid flows. The viscosity combined with a characteristic velocity U and a characteristic length L define the Reynolds number $\mathrm{Re} = \rho U L/\mu$ for viscous flows. For all flows in this work, the characteristic velocity will always be the nominal flow velocity relative to the airfoil, and the characteristic length will always be the chord length c of the airfoil. The models we consider in this work are either viscous flows at low Reynolds numbers or low-order inviscid flows. Real unsteady aerodynamics applications rarely meet the full requirements for these models. However, these models' low complexity and low computational cost justify their use under two conditions. Firstly, we avoid modeling flows that are known to be highly three-dimensional. Thus, we do not try to accurately predict flows over low-aspect-ratio wings or vortex dynamics for more than a couple of convective times after they formed. Secondly, we only seek to model the dominant flow behavior. In our study of unsteady aerodynamics, we only model the largest vortex structures near the surface, whose contributions matter most when predicting the forces on the airfoil.

A differentiable velocity field $\boldsymbol{u}$ on the unbounded domain $\boldsymbol{\Omega} = \{\boldsymbol{x} = (x, y)\}$ can be decomposed according to the Helmholtz decomposition

$$\boldsymbol{u} = \nabla\phi + \nabla \times \boldsymbol{\Psi}, \tag{2.1}$$

where ϕ is the scalar potential and $\boldsymbol{\Psi}$ is the vector potential. These potentials are governed by the Poisson equations

$$\nabla^2\phi = \Theta, \text{ and} \tag{2.2}$$

$$\nabla^2\boldsymbol{\Psi} = -\boldsymbol{\omega}, \tag{2.3}$$

where $\Theta \coloneqq \nabla \cdot \boldsymbol{u}$ is the rate of dilatation, and $\boldsymbol{\omega} \coloneqq \nabla \times \boldsymbol{u}$ is the vorticity, which quantifies the local rate of rotation of the flow (where the sign of the flow rotation follows the right-hand rule). An important integral quantity is the circulation, which is defined for an area $\mathcal{S}$, enclosed by a curve $\mathcal{C}$, as

$$\Gamma \coloneqq \int_{\mathcal{C}} \boldsymbol{u} \cdot \mathrm{d}\boldsymbol{l} = \int_{\mathcal{S}} \boldsymbol{\omega} \cdot \boldsymbol{n} \mathrm{d}S. \tag{2.4}$$

Note that in two dimensions, the velocity field has two components, u and v, the vorticity is a scalar ω, and the vector potential is also a scalar ψ, more commonly known as the streamfunction.

In the case of low-Mach number flows, the velocity satisfies the incompressibility constraint

$$\nabla \cdot \boldsymbol{u} = 0, \tag{2.5}$$

which, together with the momentum transport equation

$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \cdot \boldsymbol{u}\right) = -\nabla p + \mu\nabla^2\boldsymbol{u}, \tag{2.6}$$

forms the Navier-Stokes equations describing viscous flow dynamics. Here, the pressure p is a constraint force that enforces the incompressibility constraint (2.5) of the velocity field.

Alternatively, the flow can be described by the transport equation for vorticity, obtained by taking the curl of (2.6),

$$\rho\left(\frac{\partial\boldsymbol{\omega}}{\partial t} - \nabla\times(\boldsymbol{u}\times\boldsymbol{\omega})\right) = \mu\nabla^2\boldsymbol{\omega}. \tag{2.7}$$

The velocity field can then be reconstructed through (2.1) and (2.3), guaranteeing a divergence-free velocity since the divergence of a curl is always zero. A numerical approach to modeling these flows in the presence of surfaces is discussed later in this chapter.

Besides viscous incompressible flow, we will also focus on fast methods for *inviscid* incompressible flows. These are governed by the incompressible Euler equations (while still obeying the incompressibility constraint (2.5)) or the inviscid vorticity transport equations, obtained by removing the viscosity terms from (2.6) and (2.7), respectively. The flow description drastically simplifies when all vorticity is concentrated in singular elements consisting of points, lines and surfaces. In this case, the flow outside of these elements is a potential flow where ϕ and $\boldsymbol{\Psi}$ satisfy the Laplace equations

$$\nabla^2\phi = 0, \text{ and} \tag{2.8}$$

$$\nabla^2\boldsymbol{\Psi} = \boldsymbol{0}, \tag{2.9}$$

and the vortex elements in the flow satisfy Helmholtz' theorems: their strength is time-invariant and they are transported by the flow. This is exactly the approach followed by *vortex methods*, which we will describe later in this chapter.

2.2 Unsteady phenomena

To analyze a flow field, we can examine any of the fields introduced in the previous section, but it is often most enlightening to look at the contours of the vorticity or streamfunction fields, as we will often do in this work. The contours of the streamfunction represent the streamlines of the flow, which are tangent to the velocity at every point. Here we will look at these fields for a number of unsteady flows that exhibit unsteady aerodynamics phenomena

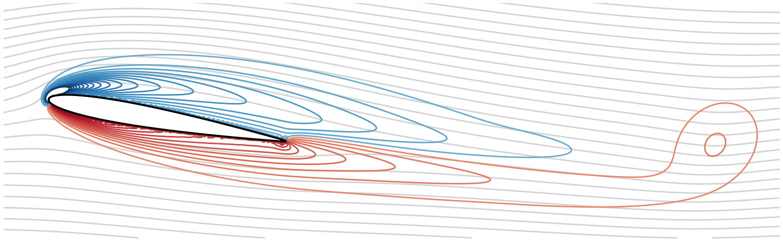


Figure 2.1: Streamlines (grey) and vorticity contours (colored) after two convective times for an impulsively started flow around an airfoil with an angle of attack of ten degrees at a Reynolds number of 400.

that are relevant for this work.

2.2.1 The Kutta condition and the starting vortex

Suppose an airfoil is suddenly brought into motion in a quiescent viscous fluid in an asymmetric way. The high velocity gradients that appear at the first instant when the fluid tries to navigate around the sharp trailing edge are immediately damped out by the viscosity, and the flow cannot remain attached. The vorticity formed in this process gets convected away from the edge by that flow, or, in other words, gets shed from the airfoil into the surrounding fluid, as clearly visible in Figure 2.1. This patch of shed vorticity is called the starting vortex, and the resulting condition, i.e., the flow leaving the trailing edge smoothly, is the Kutta condition. In an inviscid flow, this viscosity mechanism is absent, and the kutta condition must be added with mathematical constructs when modeling the flow.

As soon as it appears in the flow, the starting vortex induces a velocity field that alters the flow around the airfoil. Again, the modified flow cannot navigate the trailing edge, and more vorticity gets shed to keep the flow smooth at the edge. In turn, this vorticity again triggers the release of more vorticity and so on. This continuous process generates a free shear layer, i.e., a layer over which high tangential velocity gradients persist that are gradually diffused by viscosity. Its inviscid equivalent is an infinitesimally-thin free vortex

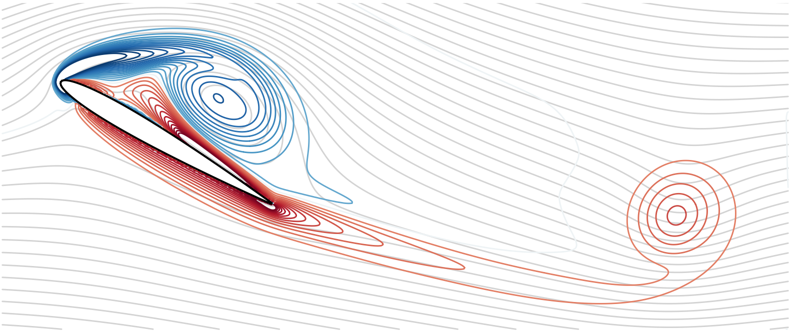


Figure 2.2: Streamlines (grey) and vorticity contours (colored) after two convective times for an impulsively started flow around an airfoil with an angle of attack of 30 degrees at a Reynolds number of 400.

sheet over which a jump in tangential velocity exists.

2.2.2 Flow separation and the leading-edge vortex

A phenomenon somewhat related to the Kutta condition can occur at the leading edge, which is often rounded. After the flow impinges on this part of the airfoil, its boundary layer usually has a favorable pressure gradient for a short part of the curved edge before it creates an adverse pressure gradient to slow down the outer flow. Depending on the Reynolds number, the curvature of the edge, and the airfoil's angle of attack, the adverse pressure gradient can be strong enough to cause a reversed flow which means the flow separates, shown in Figure 2.2.

When flow separation occurs, several interesting phenomena can take place. The separated flow forms a shear layer that is attached to the separation point and rolls up toward the surface, forming the leading-edge vortex (LEV). When it is initially developing, the shear layer remains close to the surface, and the outer flow that navigates around it can reattach to the surface at a downstream point. The flow under the shear layer forms a recirculation area called a separation bubble.

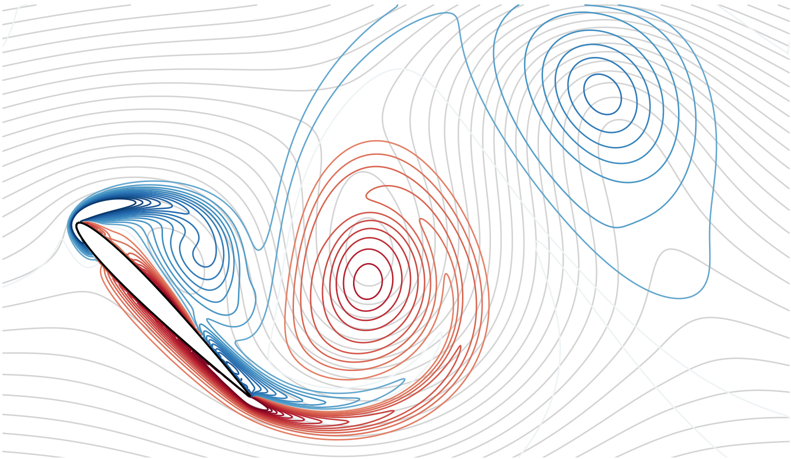


Figure 2.3: Streamlines (grey) and vorticity contours (colored) after five convective times for an impulsively started flow around an airfoil with an angle of attack of 45 degrees at a Reynolds number of 400.

Under the right conditions, the separation bubble keeps growing, and the reattachment point moves downstream until it passes the trailing edge, at which point the flow is completely separated. The LEV keeps accumulating vorticity until it sheds into the surrounding fluid that convects it away from the body, shown in Figure. The exact mechanics of this nonlinear process can vary, but it usually has a significant impact on the force, including a temporary lift increase while the low-pressure region of the LEV moves over the top surface and a subsequent drop when the flow fully separates [Eldredge and Jones, 2019].

The shed LEV can trigger the release of another strong trailing-edge vortex. At high enough Reynolds numbers (while remaining laminar) and a sustained angle of attack, this can result in the periodic release of vortices from both edges, generating a von Karman vortex street, depicted in Figure 2.3.

2.3 Inviscid vortex modeling

We will now focus on the modeling approaches of flows where the vorticity is modeled using singular elements such as point vortices, vortex sheets, or a combination of both. In the case of an inviscid and incompressible model, the irrotational flow outside of these singular vortex elements is a potential flow. Singular vortex elements that represent the free vorticity in an inviscid vortex model are advected by the local flow velocity according to Helmholtz's second theorem and can be tracked as Lagrangian points. This local flow velocity can be computed as part of the overall velocity field, which depends on the vorticity according to the Biot-Savart integral

$$\boldsymbol{u}(\boldsymbol{x}) = \int_{\mathcal{V}} \boldsymbol{K}(\boldsymbol{x}-\boldsymbol{y}) \times \boldsymbol{\omega}(\boldsymbol{y}) \mathrm{d}V(\boldsymbol{y}), \tag{2.10}$$

where $\boldsymbol{K}$ is the velocity kernel, which is equal to $\boldsymbol{x}/\left(4\pi|\boldsymbol{x}|^3\right)$ in three-dimensional flows and $\boldsymbol{x}/\left(2\pi|\boldsymbol{x}|^2\right)$ in two-dimensional flows.

In case a vortex method inserts new vortex elements in the flow behind a bluff body or at sharp edges, Kelvin's circulation theorem dictates that the circulation should be conserved. Singular potential flow elements can also serve to enforce the no-penetration condition, with the most common choice in vortex methods being a distribution of singular vorticity on the body, denoted as the bound vortex sheet, which will have the equal and opposite circulation of the vorticity that was inserted into the flow. The strength of a vortex sheet is usually denoted by γ and represents the local jump in velocities over the sheet

$$\gamma = \boldsymbol{n} \times (\boldsymbol{u}^{+} - \boldsymbol{u}^{-}), \tag{2.11}$$

where + is in the direction of the normal vector $\boldsymbol{n}$. The Biot-Savart integral (2.10) allows to formulate a boundary integral equation that can be solved for the unknown strength of the vortex sheet. Note that the inviscid flow around an airfoil is generally non-unique but that the Kutta condition constraint restricts these solutions to a unique solution where the flow leaves the trailing edge smoothly.

In what follows, we will discuss some relevant methods for this work. Firstly, we will assume the strict conditions of classical unsteady aerodynamics and look at the linear Wagner/Theodorsen model for the lift response to arbitrary small perturbations of an airfoil in motion. We describe the formulation in the time-domain, which we will make use of in Chapter 4, and derive a state-space form, which we will use as one of the systems for the reinforcement learning in Chapter 5. Then we will introduce vortex methods that can model more complex, nonlinear flow behavior, which serves as the background for the grid-based vortex model in Chapter 3.

2.3.1 Classical unsteady aerodynamics

In the early 20th century, aerodynamicists developed simple formulations to predict the forces in unsteady flows around airfoils under some specific assumptions. This theory is known as classical unsteady aerodynamics [von Kármán and Sears, 1938], and its assumptions are as follows. As depicted in Figure 2.4, the airfoil is a flat plate with chord length c at a small angle of attack α, where $|\alpha| \ll 1$, and its main motion is a constant velocity U in the negative x-direction. Any other motions are perturbations of the angle of attack with an angular velocity $\dot{\alpha}$ applied at a pivot point at a coordinate $(d, 0)$ in the body-fixed reference frame positioned at the center of the airfoil with its $\tilde{x}$-axis along the chord and the vertical motion $\dot{h}$ of that pivot point. These perturbations are considered small compared to the forward velocity of the plate, i.e., $|\dot{\alpha}| \ll U/c$ and $|\dot{h}| \ll U$. The wake is assumed to lie along the $\tilde{x}$-axis of the body, and the wake vorticity convects with the velocity U away from the airfoil.

Classical unsteady aerodynamics theory provides an expression for the bound vortex sheet strength when the Kutta condition is applied at the trailing edge under the previously mentioned assumptions:

$$\int_{c/2}^{\infty} \frac{(\tilde{x} + c/2)^{1/2}}{(\tilde{x} - c/2)^{1/2}} \gamma_{\mathcal{S}}(\tilde{x}) \mathrm{d}\tilde{x} = -\Gamma^{\mathrm{qs}} \tag{2.12}$$

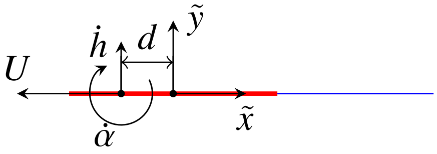


Figure 2.4: Diagram of the airfoil and wake in classical unsteady aerodynamics.

where the quasi-steady circulation $\Gamma^{\mathrm{qs}} := -\pi c U \alpha_{\mathrm{eff}}$ depends on the effective angle of attack

$$\alpha_{\mathrm{eff}} := -\frac{\dot{h}}{U} + \alpha + \left(\frac{c}{4} - d\right)\frac{\dot{\alpha}}{U}, \tag{2.13}$$

Assuming that the wake convects with the velocity U and equation (2.12) has to be satisfied at any instant of time, the wake strength can be determined. Furthermore, the classical theory also provides a general expression for the lift on the plate:

$$f_y = \underbrace{-\frac{\pi}{4}\rho c^2 \left(\ddot{h} + d\ddot{\alpha} - U\dot{\alpha}\right)}_{\text{added-mass lift } f_y^{\mathrm{am}}} \underbrace{-\rho U \Gamma^{\mathrm{qs}}}_{\substack{\text{quasi-steady} \\ \text{lift } f_y^{\mathrm{qs}}}} \underbrace{-\frac{1}{2}\rho U c \int_{c/2}^{\infty} \frac{\gamma_{\mathcal{S}}(\tilde{x}_{\mathcal{S}})}{(\tilde{x}_{\mathcal{S}}^2 - c^2/4)^{1/2}} \mathrm{d}\tilde{x}_{\mathcal{S}}}_{\text{wake lift } f_y^{\mathcal{S}}}. \tag{2.14}$$

The last two terms (the quasi-steady and wake lift) together form the circulatory lift, or $f_y^{\mathrm{circ}} = f_y^{\mathrm{qs}} + f_y^{\mathcal{S}}$.

The distributions of the pressure jump across the plate corresponding to the added-mass and circulatory lift were first derived by Neumark [1952] and are for rigid-body motion given by

$$[p^{\mathrm{am}}]_-^+(\tilde{x}_{\mathcal{P}}) = -2\rho\left(\ddot{h} + d\ddot{\alpha} - U\dot{\alpha}\right)\left(c^2/4 - \tilde{x}_{\mathcal{P}}^2\right)^{1/2} \tag{2.15}$$

$$[p^{\mathrm{circ}}]_-^+(\tilde{x}_{\mathcal{P}}) = \left(-2\rho U^2 \alpha_{\mathrm{eff}} + \frac{\rho U}{\pi}\int_{c/2}^{\infty} \frac{\gamma_{\mathcal{S}}(\tilde{x}_{\mathcal{S}})}{(\tilde{x}_{\mathcal{S}}^2 - c^2/4)^{1/2}} \mathrm{d}\tilde{x}_{\mathcal{S}}\right) \frac{\left(c/2 - \tilde{x}_{\mathcal{P}}\right)^{1/2}}{\left(c/2 + \tilde{x}_{\mathcal{P}}\right)^{1/2}}. \tag{2.16}$$

The full pressure is then $[p]_-^+ = [p^{\mathrm{am}}]_-^+ + [p^{\mathrm{circ}}]_-^+$. Note that the first factor in parentheses in Eq. (2.16) is equal to $-2f_y^{\mathrm{circ}}/\pi c$ and thus the circulatory pressure at any point along the airfoil linearly scales with the circulatory lift.

For an arbitrary, smoothly-varying α_{eff}, a more practical expression can be obtained for the circulatory lift if we formulate it as the convolution of the rate of change of Γ^{qs} with the non-dimensional lift response to a step change in Γ^{qs}, known as Wagner's function Φ [Wagner, 1925], plus the contribution of the initial value of Γ^{qs} times Wagner's function evaluated at the current time:

$$f_y = f_y^{\text{am}} - \rho U \Gamma^{\text{qs}}(0)\Phi(t^*) - \rho U \int_0^{t^*} \dot{\Gamma}^{\text{qs}}(\tau^*)\Phi(t^* - \tau^*)\mathrm{d}\tau^*. \tag{2.17}$$

There is no analytical form of Wagner's function available, but several approximations exist, including one proposed by Jones [1938] that is expressed as

$$\Phi(t^*) \approx 1 - 0.165\mathrm{e}^{-0.091t^*} - 0.335\mathrm{e}^{-0.6t^*}. \tag{2.18}$$

The circulatory lift in (2.17) can be formulated in the Laplace domain as a multiplication of a transfer function with the quasi-steady lift:

$$\begin{aligned}
\mathcal{L}\{f_y^{\text{circ}}\} &= -\rho U \Gamma^{\text{qs}}(0)\mathcal{L}\{\Phi\} - \rho U \mathcal{L}\{\Phi\}\mathcal{L}\{\dot{\Gamma}^{\text{qs}}\} && (2.19)\\
&= -\rho U s \mathcal{L}\{\Phi\}\mathcal{L}\{\Gamma^{\text{qs}}\} && (2.20)\\
&= s\mathcal{L}\{\Phi\}\mathcal{L}\{f_y^{\text{qs}}\} && (2.21)
\end{aligned}$$

where s is the Laplace variable. The transfer function $C(s) = s\mathcal{L}\{\Phi\}$ based on the Wagner function approximation (2.18) was also derived by Jones [1938] and can be expressed as

$$\begin{aligned}
C(s) &= s\int_0^{\infty} \Phi(\tau^*)e^{-s\tau^*}\mathrm{d}\tau^* && (2.22)\\
&\approx \frac{0.5s^2 + 0.5616s + 0.0546}{s^2 + 0.691s + 0.0546}. && (2.23)
\end{aligned}$$

Furthermore, Jones [1938] noted, like Garrick [1938] before him, that C evaluated on the imaginary axis, i.e. its steady-state behavior for a purely oscillatory, quasi-steady circulation, is equivalent to Theodorsen's function [Theodorsen, 1935].

For control purposes, we seek the state-space representation of (2.22) and express the lift in terms in terms of the lift coefficient $c_L = 2f_y/(\rho U^2 c)$. The controllable canonical

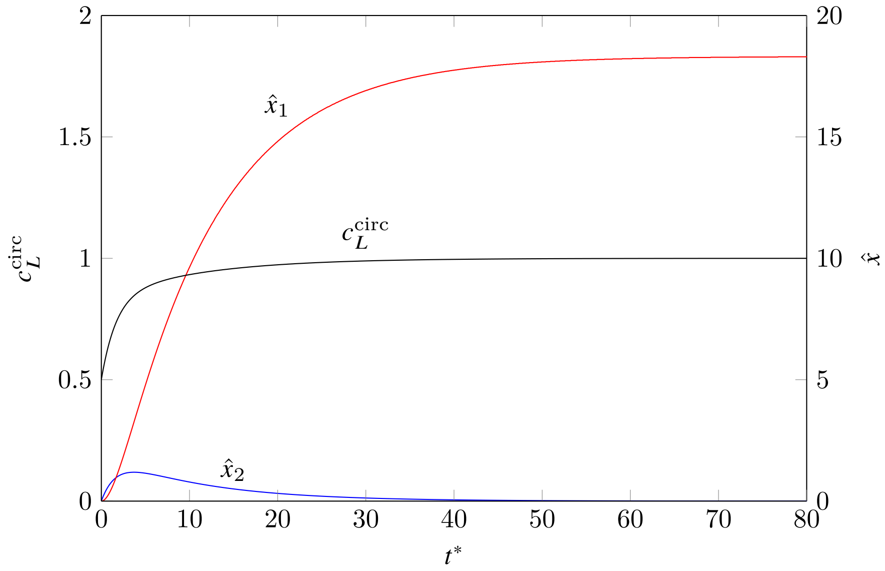


Figure 2.5: Unit step response of the circulatory lift coefficient and the states of the controllable canonical state-space representation for the Wagner transfer function by Jones [1938].

state-space representation for $C(s)$ is

$$\begin{bmatrix} \dot{\hat{x}}_1 \\ \dot{\hat{x}}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} -0.691 & -0.0546 \\ 1 & 0 \end{bmatrix}}_{\hat{\mathbf{A}}} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{\hat{\mathbf{B}}} c_L^{\text{qs}} \tag{2.24}$$

$$c_L^{\text{circ}} = \underbrace{\begin{bmatrix} 0.2161 & 0.0273 \end{bmatrix}}_{\hat{\mathbf{C}}} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0.5 \end{bmatrix}}_{\hat{\mathbf{D}}} c_L^{\text{qs}}. \tag{2.25}$$

The response of this system to a unit step increase of c_L^{qs} is shown in Figure 2.5.

Following Brunton and Rowley [2013], we can use (2.22) to create a multi-input, single-output transfer function for the complete lift expression by applying the Laplace transform to (2.17) for each input:

$$\begin{aligned}\mathbf{G}(s) &= \begin{bmatrix} G_{\ddot{h}}(s) & G_{\ddot{\alpha}}(s) \end{bmatrix} && (2.26)\\ &= \begin{bmatrix} -\frac{\pi c}{2U^2} - \frac{2\pi}{sU}C(s) & \frac{\pi c}{2U}\left(\frac{1}{s}+\frac{d}{U}\right) + 2\pi\left(\frac{1}{s^2}+\frac{1}{sU}\left(\frac{c}{4}-d\right)\right)C(s) \end{bmatrix}, && (2.27)\end{aligned}$$

where $G_{\ddot{h}}$ and $G_{\ddot{\alpha}}$ are the transfer functions that give the reponse of the output c_L to the input signals $\ddot{h}$ and $\ddot{\alpha}$, respectively. We can also cast this result also into a minimal state-space realization:

$$\begin{bmatrix} \dot{\alpha}_{\text{eff}} \\ \ddot{\alpha} \\ \dot{\hat{\mathbf{x}}} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & \mathbf{0} \\ 0 & 0 & \mathbf{0} \\ 2\pi\hat{\mathbf{B}} & \mathbf{0} & \hat{\mathbf{A}} \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} \alpha_{\text{eff}} \\ \dot{\alpha} \\ \hat{\mathbf{x}} \end{bmatrix} + \underbrace{\begin{bmatrix} -\frac{1}{U} & \frac{1}{U}\left(\frac{c}{4}-d\right) \\ 0 & 1 \\ \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\mathbf{B}} \begin{bmatrix} \ddot{h} \\ \ddot{\alpha} \end{bmatrix} \qquad (2.28)$$

$$c_L = \underbrace{\begin{bmatrix} 2\pi\hat{\mathbf{D}} & \frac{\pi c}{2U} & \hat{\mathbf{C}} \end{bmatrix}}_{\mathbf{C}} \begin{bmatrix} \alpha_{\text{eff}} \\ \dot{\alpha} \\ \hat{\mathbf{x}} \end{bmatrix} + \underbrace{\begin{bmatrix} -\frac{\pi c}{2U^2} & -\frac{\pi c d}{2U^2} \end{bmatrix}}_{\mathbf{D}} \begin{bmatrix} \ddot{h} \\ \ddot{\alpha} \end{bmatrix} \qquad (2.29)$$

2.3.2 Dynamic wake vortex methods

Although classical unsteady aerodynamics theory provides powerful tools to model unsteady flows, it cannot model vortex shedding from the leading edge nor the dominating nonlinear flow behavior that arises when the interactions between vortex elements become stronger. In such cases, the flat wake model is insufficient, and a wake model has to be adopted where the vortex elements are convected by the local velocity. These potential flow around these elements can be modeled in different ways. This can be done by either using direct interaction between the potential flow elements or by calculating the velocity on a grid over the entire domain and the eventual choice dictates the treatment of boundary and edge conditions.

In the first approach, the Green's function of the Laplacian is applied to the Poisson equation in the velocity-vorticity formulation to give the Biot-Savart integral, which provides the exact solution for a velocity field that satisfies unbounded boundary conditions. Biot-Savart vortex methods often smooth the Biot-Savart kernel, equivalent to replacing point vortices by vortex blobs [Chorin and Bernard, 1973], to suppress Kelvin-Helmholtz instabilities below a certain wavelength resulting from the interactions between closely spaced vortex elements. The solution generally requires $O(N^2)$ operations, with N the number of vortex elements, to sum the influences of each discretized vortex element on every other element. With fast multipole methods, it scales optimally as $O(N)$ but with a large prefactor and overhead cost. Inviscid vortex methods of this kind can straightforwardly use the potential flow tools to enforce the no-penetration, such as conformal mapping or solving the integral equation for a surface singularity distribution through analytical inversion, panel discretization, or with Fourier expansions. For a detailed review of this subject, the reader is referred to Cottet and Koumoutsakos [2000] and Eldredge [2019].

A second approach to calculate the flow velocity follows from the discretization of the Poisson equation in the velocity-vorticity formulation or streamfunction-vorticity formulation on an Eulerian grid over the domain of interest and is called a vortex-in-cell (VIC) approach, first developed by Christiansen [1973]. The procedure requires first to transfer the circulation from Lagrangian vortex elements onto the grid, then to solve the discrete Poisson equation, and finally to interpolate the velocity (or the curl of the streamfunction) back to the vortex elements. It introduces discretization errors but only requires $O(M \log M)$ operations, with M the number of grid points, to solve the Poisson equation with current numerical techniques and $O(N)$ operations to perform the regularization and interpolation. Similar to the regularized Biot-Savart kernel, the grid spacing together with the vorticity regularization scheme determine the cut-off wavelength below which the Kelvin-Helmholtz instabilities get suppressed.

The early work on VIC methods focused on inviscid vortex dynamics using Fourier-based Poisson solvers with Dirichlet or periodic boundary conditions [Meng and Thomson, 1978, Baker, 1979, Couët et al., 1981] and the analysis of different interpolation kernels [Ebiana and Bartholomew, 1996]. After viscous schemes for vortex methods were introduced, VIC methods increasingly replaced Biot-Savart methods in an effort to speed up vortex methods for viscous flows, leading to methods with over a billion vortex particles [Chatelain et al., 2008]. This also stimulated the development of VIC methods for external flows over bodies, mostly for viscous flows. The most straightforward way to include a body in the flow is to use a body-fitted mesh as in Cottet and Poncet [2004], who apply the Helmholtz decomposition on the flow in their VIC method and place a Neumann boundary condition for the scalar potential on the body to account for its presence and employ an analytic boundary condition for the far-field. However, a body-fitted mesh is case-specific and, therefore, the same work (and later also Poncet [2009]), develops an immersed boundary method by introducing a singular distribution of sources that represents the influence of the body and is smeared onto a cartesian grid using a discrete approximation to the Dirac delta function. The result is a source term that is inserted in the Poisson equation for the scalar potential which is solved on the grid. Similar to the immersed boundary methods, Brinkman penalization methods do not require body-fitted meshes. In vortex methods, the Brinkman penalization method [Coquerelle and Cottet, 2008, Rossinelli et al., 2010, Gazzola et al., 2011, Rasmussen et al., 2011, Chatelin and Poncet, 2014] adds a volume forcing term to the vorticity transport equation that includes a penalization parameter, equivalent to the porosity of the body. However, the method suffers from a strong time step restriction, which motivated Hejlesen et al. [2015] to use an iterative Brinkman penalization method, which Spietz et al. [2017] extended to three dimensions. Gillis et al. [2017] formulates this method as a linear system and uses a recycling iterative solver to obtain the solution more efficiently.

The immersed boundary method and Brinkman penalization method both smear out the influence of the interface onto nearby grid points. LeVeque and Li [1994] developed the immersed interface method (IIM) to overcome this issue and to obtain a higher spatial order of accuracy than the immersed boundary method. The premise of this method is to discretize the jump conditions caused by the interface with finite differences instead of discretizing the

Dirac delta function, and the result is a sharp representation of the interface with a second or higher-order accuracy. Marichal et al. [2014] applies the explicit-jump IIM [Wiegmann and Bube, 2000] in his potential flow method. The influence of the interface was condensed into an extra source term in the streamfunction Poisson equation and it was recognized that the term is equivalent to a bound vortex sheet strength regularized to the grid. The bound vortex sheet strength, streamfunction field, and outer boundary condition on the streamfunction are then computed iteratively. The work presents results from the flow over a cylinder and an airfoil, for which the Kutta condition enforced through discretization of the streamfunction normal derivatives at the trailing edge. Gillis et al. [2018] extends this method, but applies the IIM on the scalar potential instead. An explicit formula for the singular distribution of sources on the interface is then obtained by applying the Sherman-Morrison-Woodbury decomposition formula to the Poisson equation, similar to Poncet [2009]. Furthermore, by solving the Poisson equation using the lattice Green's function [Katsura and Inawashiro, 1971], which automatically satisfies far-field boundary conditions, the method is no longer iterative and the cost is greatly reduced. Gillis et al. [2019] applies this method again to the streamfunction in two dimensions and employs it in a viscous VIC method.

Chapter 3 introduces a saddle-point system formulation for grid-based potential flow solvers where the circulation of the body or shed vortices can be added as a Lagrange multiplier to enforce the Kutta condition constraint.

2.4 Viscous modeling

In a viscous flow, the pressure combines with the viscous stresses to give the stress tensor $\boldsymbol{t} = -p\boldsymbol{I} + \mu\left(\nabla\boldsymbol{u} + \nabla^{\top}\boldsymbol{u}\right)$. The surface traction $\boldsymbol{t}\cdot\boldsymbol{n}$ is the projection of the stress tensor at a surface onto the normal vector $\boldsymbol{n}$ of that surface and provides the means to compute the force $\boldsymbol{f}$ and moment $\boldsymbol{m}$ about the origin exerted by the fluid on the surface:

$$\boldsymbol{f} = \int \boldsymbol{t} \cdot \boldsymbol{n} \, \mathrm{d}S, \tag{2.30}$$

$$\boldsymbol{m} = \int \boldsymbol{x} \times (\boldsymbol{t} \cdot \boldsymbol{n}) \, \mathrm{d}S. \tag{2.31}$$

In this work, the nominal flow relative to the airfoil is always parallel to and pointing in the same direction as the positive x-axis of the inertial reference frame, and we regularly decompose the force vector into its inertial x- and y-components, which we refer to as the drag and lift, respectively. Note that in a viscous flow, the surface traction is the constraint force that enforces the no-slip condition $\boldsymbol{u} = \boldsymbol{u}_b$ on the surface.

In general, modeling the incompressible Navier-Stokes equations (2.5) and (2.6) is often more complicated than modeling the potential flow in inviscid vortex methods. Not many viscous flows have an analytical solution, especially in unsteady aerodynamics, and elementary flow solutions can generally not be superposed. Furthermore, the numerical modeling of viscous flows is a broad field containing many different approaches. In this work, we only discuss the *immersed boundary projection method* and its variant, the *immersed layers projection method*.

The immersed boundary projection method Taira and Colonius [2007] reformulates the original immersed boundary method by Peskin [1972] as a saddle-point system that can be solved using the LU-decomposition, similar to the analysis of Perot [1993] of the fractional-step method. The original immersed boundary method discretizes the Navier-Stokes equations on a staggered Cartesian grid and represents the interface with discrete Lagrangian points that are not bound to the grid points and where point forcing is applied to the fluid to enforce the no-slip condition. The forcing is transferred to the grid points using a *regularization* or *immersion* operator that applies a convolution with the *discrete Dirac delta function*, and a similar *interpolation* operator transfers the fluid velocity to the position of the surface points. As a result, this method doesn't require the shapes of the mesh and surface to agree and is ideal for numerically modeling the flow around moving or geometrically complex interfaces.

The equivalent continuous formulation of the immersed boundary treatment of the Navier-Stokes equations is

$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t}+\boldsymbol{u}\cdot\nabla\cdot\boldsymbol{u}\right)=-\nabla p+\mu\nabla^2\boldsymbol{u}+\int_{\boldsymbol{\Omega}}\boldsymbol{\sigma}\left(\boldsymbol{X}(\xi)\right)\boldsymbol{\delta}(\boldsymbol{X}-\boldsymbol{x})\mathrm{d}S(\xi), \tag{2.32}$$

$$\nabla\cdot\boldsymbol{u}=0, \tag{2.33}$$

$$\boldsymbol{u}\left(\boldsymbol{X}(\xi)\right)=\int_{\partial\boldsymbol{\Omega}_{\mathrm{b}}}\boldsymbol{u}(\boldsymbol{x})\boldsymbol{\delta}(\boldsymbol{x}-\boldsymbol{X})\mathrm{d}\boldsymbol{x}=\boldsymbol{u}_b\left(\boldsymbol{X}(\xi)\right). \tag{2.34}$$

where $\boldsymbol{\sigma}$ is the surface force that acts as a Lagrange multiplier to enforce the no-slip condition, $\boldsymbol{\delta}$ is the multi-dimensional Dirac Delta function, $\boldsymbol{X}(\xi)$ is the position of the body, parametrized by ξ, and $\boldsymbol{u}_b$ is the velocity of the surface.

Alternatively, the immersed boundary treatment can be formulated for the vorticity-streamfunction form of the Navier-Stokes equations Colonius and Taira [2008] as

$$\rho\left(\frac{\partial \boldsymbol{\omega}}{\partial t}-\nabla\times\left(\boldsymbol{u}\times\boldsymbol{\omega}\right)\right)=\mu\nabla^2\boldsymbol{\omega}-\nabla\times\int_{\boldsymbol{\Omega}}\boldsymbol{\sigma}\left(\boldsymbol{X}(\xi)\right)\boldsymbol{\delta}(\boldsymbol{X}-\boldsymbol{x})\mathrm{d}S(\xi), \tag{2.35}$$

$$\boldsymbol{u}\left(\boldsymbol{X}(\xi)\right)=\int_{\partial\boldsymbol{\Omega}_{\mathrm{b}}}\boldsymbol{u}(\boldsymbol{x})\boldsymbol{\delta}(\boldsymbol{x}-\boldsymbol{X})\mathrm{d}\boldsymbol{x}=\boldsymbol{u}_b\left(\boldsymbol{X}(\xi)\right). \tag{2.36}$$

Eldredge [2022] points out that in the immersed boundary method, the surface force function $\boldsymbol{\sigma}$ is equal to $(\boldsymbol{t}^+-\boldsymbol{t}^-)\cdot\boldsymbol{n}$, the sum of the surface tractions exerted from either side (with + in the direction of the surface normal), and that it is thus impossible to obtain the surface traction on the exterior surface from $\boldsymbol{\sigma}$ without knowing $\boldsymbol{t}^-$ from the interior flow. The immersed layers method Eldredge [2022] is developed to remedy this issue by accounting for the jump in velocity over the interface. However, in our work, this surface force will always represent the physical surface traction on the surface. That is because the surfaces we study are either static airfoils or moving infinitesimally-thin airfoils. Nevertheless, the viscous flow simulations in this work are performed using a viscous flow solver code that relies on the immersed layers projection treatment of the vorticity-streamfunction form of the Navier-Stokes equations and which is available on Github [Eldredge, 2023].

The saddle-point system that results from discretizing the equations using the immersed-boundary projection method is solved using a block-LU decomposition. The projection step generates a Poisson equation that can be solved using a single inversion of the system, which can be reused for every subsequent simulation timestep, provided that the surfaces are not in motion. When the surfaces are in motion, the grid operators change at every time step, and the resulting system would have to be inversed every time. However, if in the body-fixed reference frame, the boundary conditions don't change, the problem can be solved as if the body were not moving, and again only a single inversion of the system will be necessary for the whole simulation.

CHAPTER 3

Grid-based modeling of potential flows

In this chapter we present a grid-based treatment for planar potential flow. The focus of this work is on the flow around point vortices, but the treatment can easily be extended to account for sources and sinks. In the case of the flow around vortices, the treatment follows a vortex-in-cell approach. In our implementation, the streamfunction-vorticity Poisson equation is solved for the streamfunction on the grid using the lattice Green's function, such that unbounded boundary conditions are accounted for. To enforce the no-penetration condition on surfaces in the flow, the treatment is presented by using the immersed-boundary projection method and our implementation is therefore approximately first-order accurate in space [Colonius and Taira, 2008]. This approach consists in adding an extra singular vorticity source term to the streamfunction-vorticity Poisson equation that is distributed over the discrete surface points and is smeared onto the nearby grid nodes. This extra vorticity term represents the bound vortex sheet strength and assumes the role of a Lagrange multiplier in this method. The modified Poisson equation combined with the no-penetration constraint then forms a saddle-point system that can be solved with the Schur's complement method. Note that the IIM introduces a similar modification to the Poisson equation [Marichal et al., 2014] and the Sherman-Morrison-Woodbury decomposition can produce an expression for the discrete vortex sheet strength [Gillis et al., 2019] that is equivalent to the formula we obtain by using the Schur's compelement method. Consequently, one could use the IIM to obtain a second-order method instead. Drawing inspiration from the analytical treatment of the Kutta condition in Biot-Savart methods, this work then introduces a new way of enforcing the Kutta condition in a discrete potential flow treatment by decomposing the discrete vortex sheet strength into a singular and non-singular part and constraining the non-singular part. This amounts to algebraically constraining the system arising from the immersed-boundary projection method to make it well-behaved.

3.1 Basic two-dimensional potential flow problem

We consider here a staggered, Cartesian grid with uniform cell size Δx and of infinite extent. The space corresponding to data at cell vertices (nodes) on this grid is denoted by $\mathcal{N}$, and the physical coordinates of these nodes by x and y. Furthermore, we consider a finite number N_v of Lagrangian, singular point vortices. The space of scalar data on these points is denoted by $\mathcal{W}^{N_v}$ and we define $\Gamma_v \in \mathcal{W}^{N_v}$ as the vector containing the strengths of the point vortices in our grid-based treatment. The basic (unbounded) potential flow problem is expressed as

$$\mathsf{L}\mathsf{s} = -\mathsf{w}, \tag{3.1}$$

where L is the discrete 5-point Laplacian operator, $\mathsf{s} \in \mathcal{N}$ is the discrete streamfunction, and $\mathsf{w} \in \mathcal{N}$ the discrete vorticity. The discrete velocity field v, whose components lie on the faces of the cells with the corresponding normals, is computed from s by the discrete curl operation,

$$\mathsf{v} = \mathsf{C}\mathsf{s}. \tag{3.2}$$

The operator C applies centered differences between the nodes to obtain the velocity components at the intermediate centers of cell faces. We denote the space of data that lie on cell faces by $\mathcal{F}$, so $\mathsf{C} : \mathcal{N} \mapsto \mathcal{F}$. Figure 3.1(a) shows the staggered grid structure.

The grid differencing operators L and C and others to be defined are scaled by the grid spacing, and thus represent second-order approximations of the corresponding continuous operators, and the discrete streamfunction s, vorticity w, and velocity v are each approximations of their continuous counterparts. The total flow circulation is equal to the sum of grid vorticity multiplied by the cell area Δx^2. To support the work that follows, we define an inner product on the grid nodes,

$$\langle \mathsf{w}_1, \mathsf{w}_2 \rangle_{\mathcal{N}} = \Delta x^2 \mathsf{w}_1^T \mathsf{w}_2, \tag{3.3}$$

for w_1, $\mathsf{w}_2 \in \mathcal{N}$. The total circulation can then be written compactly as

$$\Gamma_{\mathsf{w}} = \langle \mathsf{1}, \mathsf{w} \rangle_{\mathcal{N}}, \tag{3.4}$$

where $\mathsf{1} \in \mathcal{N}$ is a grid vector of ones.

The particular solution of equation (3.1) can be written down immediately with the help of the lattice Green's function for L [Katsura and Inawashiro, 1971, Cserti, 2000, Liska and Colonius, 2014a]. We denote this simply by the inverse operator,

$$\mathsf{s} = -\mathsf{L}^{-1}\mathsf{w}. \tag{3.5}$$

It can be shown that, with a suitable truncation of the grid, both L and its inverse are symmetric operators. However, L is only positive semi-definite, and an additional homogeneous solution, $\mathsf{s}_\infty \in \mathcal{N}$—for example, corresponding to a uniform flow—can be added to the particular solution of this equation $-\mathsf{L}^{-1}\mathsf{w} + \mathsf{s}_\infty$. This homogeneous solution allows us to satisfy boundary conditions at (discrete) infinity. The size of the domain in our simulations is therefore not relevant, as long as it includes the features that are of interest.

As in the vortex-in-cell approach [Christiansen, 1973], the discrete vorticity is obtained by immersing the vortex elements into the grid and transferring their circulation to the nearby nodes using a tensor product of two one-dimensional, discrete Dirac delta functions:

$$\mathsf{w} = \frac{1}{\Delta x^2} \sum_{q=1}^{N_v} \Gamma_{v,q} d\left(\frac{\mathsf{x} - X_q}{\Delta x}\right) d\left(\frac{\mathsf{y} - Y_q}{\Delta x}\right), \tag{3.6}$$

where d is a discrete Dirac delta function[1]. By the properties imposed on d,

$$\langle \mathsf{1}, \mathsf{w} \rangle_{\mathcal{N}} = \sum_{q=1}^{N_v} \Gamma_{v,q}. \tag{3.7}$$

[1] In particular, $d(x/\Delta x)/\Delta x$ represents a Dirac sequence as $\Delta x \to 0$.

In this work, d is the M_4' function from Monaghan [1985], depicted in Figure 3.1(b); panel (a) depicts how data from a Lagrangian point is regularized onto the grid nodes. For compact notation, we define $\mathsf{d}_q \in \mathcal{N}$, the grid vorticity field generated when a vortex of unit strength is immersed into the grid, so that we can write (3.6) as

$$\mathsf{w} = \sum_{q=1}^{N_v} \Gamma_{v,q} \mathsf{d}_q. \tag{3.8}$$

Even more compactly, we define the *regularization operator* ${}^{\mathcal{N}}\mathsf{R}_{\mathcal{W}} : \mathcal{W}^{N_v} \mapsto \mathcal{N}$, whose N_v columns are d_q, $q = 1, \ldots, N_v$, which allows us to write (3.8) as

$$\mathsf{w} = {}^{\mathcal{N}}\mathsf{R}_{\mathcal{W}} \Gamma_v, \tag{3.9}$$

the discrete streamfunction field as $\mathsf{s} = -\mathsf{L}^{-1}\,{}^{\mathcal{N}}\mathsf{R}_{\mathcal{W}} \Gamma_v + \mathsf{s}_\infty$ and the velocity field as

$$\mathsf{v} = -\mathsf{C}\mathsf{L}^{-1}\,{}^{\mathcal{N}}\mathsf{R}_{\mathcal{W}} \Gamma_v + \mathsf{C}\mathsf{s}_\infty. \tag{3.10}$$

Figure 3.2 shows a spatial grid refinement analysis for a flow consisting of point vortices of random strength that are randomly positioned in the lower-left quadrant of the domain. The analysis verifies that the discretization technique of the Poisson equation is second-order accurate in Δx. We compute the error as $\epsilon_{\mathsf{s}} = \|\mathsf{s} - \psi(\mathsf{x},\mathsf{y})\|_2 / \|\psi(\mathsf{x},\mathsf{y})\|_2$, where ψ is the exact solution for the streamfunction. We only consider the values in the upper right quadrant of the domain to exclude the positions of the point vortices, because the exact singularities at these positions are not comparable to the regularized, discrete version.

To obtain the velocity at the locations of the point vortices, the discrete velocity field v should first be interpolated from cell faces to the nodes (using simple averages). Then, the velocity can be interpolated onto the point vortices with the *interpolation operator* ${}^{\mathcal{W}}\mathsf{E}_{\mathcal{N}} : \mathcal{N} \mapsto \mathcal{W}^{N_v}$, which is the transpose of the regularization operator, ${}^{\mathcal{W}}\mathsf{E}_{\mathcal{N}} = {}^{\mathcal{N}}\mathsf{R}_{\mathcal{W}}^T$, to obtain an overall interpolation scheme that is consistent with (3.9).

It is worth noting here that, if we wish to include sources and sinks, then we can make additional use of a discrete scalar potential, lying at cell centers, $\phi \in \mathcal{C}$, whose governing

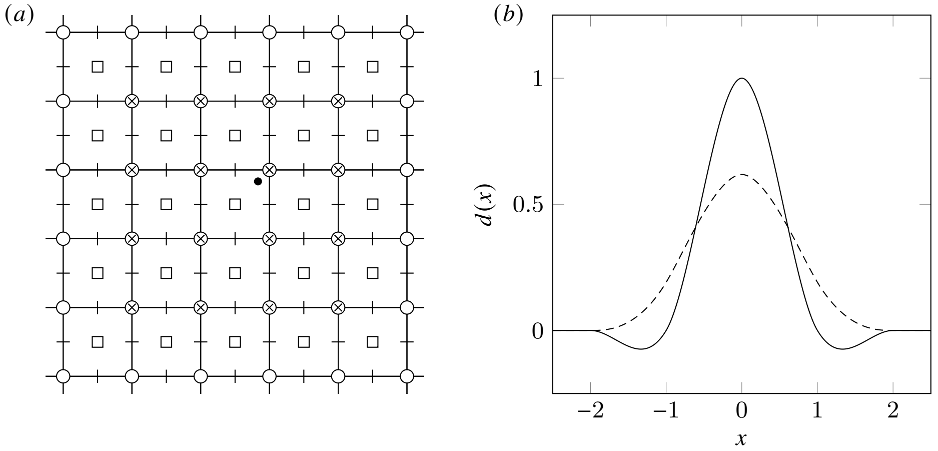


Figure 3.1: (*a*) Schematic of the grid and the regularization of data from a Lagrangian point (●) onto the grid with a discrete Dirac delta function of radius two. Symbols (○), (□), (|), and (-) denote the locations holding the components of the nodes $\mathcal{N}$, cell centers $\mathcal{C}$, horizontal faces $\mathcal{F}_x$, and vertical faces $\mathcal{F}_y$, respectively. Symbols ($\times$) denote the nodes that are affected by the regularization. (*b*) Two examples of discrete Dirac delta functions of radius two: the M_4' function (——) from Monaghan [1985] and the smoothed three-point function (- - - -) from Yang et al. [2009].

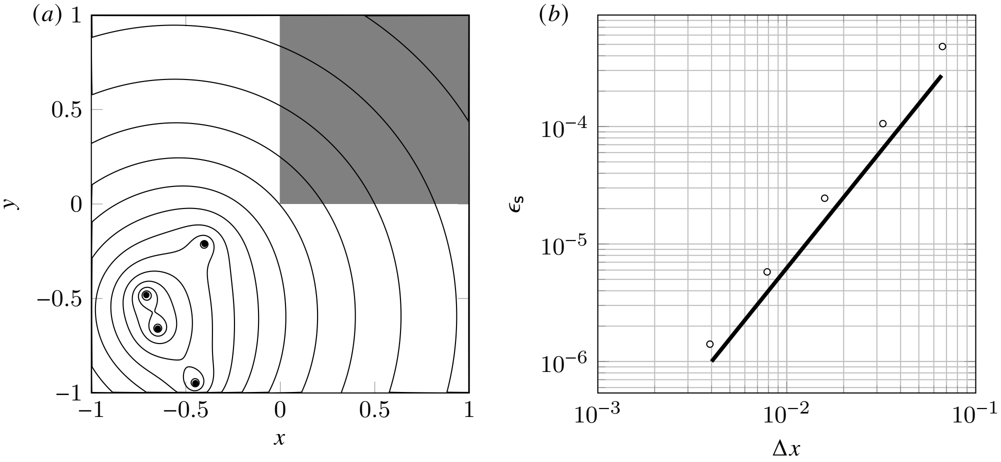


Figure 3.2: (*a*) Contours (——) of the discrete streamfunction for randomly positioned point vortices (●) of random strengths, and (*b*) its error (○) over the shaded area for different grid spacings. Overlaid is an error (——) that scales as Δx^2.

equation is the discretized version of (2.2), analogous to (3.1). The sources and sinks can be immersed into a rate of dilatation field at cell centers in similar manner to vortices to cell nodes (3.6). The velocity field due to the combination of vortices, sinks, and sources can then be composed using the discrete version of the Helmoltz decomposition (2.1):

$$\mathsf{v} = \mathsf{G}\phi + \mathsf{C}\mathsf{s}. \tag{3.11}$$

Analogous to the discrete curl operator, the discrete gradient operator G applies centered differences to scalar data at cell centers $\mathcal{C}$ to obtain the velocity components at the cell faces, so $\mathsf{G} : \mathcal{C} \mapsto \mathcal{F}$.

3.2 Potential flow with an impenetrable surface

Now, let us suppose we have a rigid impenetrable surface $\mathcal{S}$, on which we seek to enforce the no-penetration condition for the streamfunction. The no-penetration condition asserts that the normal components of the fluid velocity and this surface velocity must be equal. For rigid bodies, this surface motion can be alternatively described by a streamfunction, and the no-penetration condition in the absence of sources and sinks can be imposed equivalently (in two dimensions) by setting the fluid streamfunction equal to that of the surface ψ_b up to a uniform value. In continuous form, this is described by the Dirichlet problem

$$\nabla^2 \psi = -\omega \tag{3.12}$$

$$\psi\left(\boldsymbol{x}\right) = \psi_b\left(\boldsymbol{x}\right), \, \boldsymbol{x} \in \mathcal{S}. \tag{3.13}$$

In the presence of sources and sinks, one can again solve the auxiliary problem for the scalar potential and enforce the no-penetration condition only for the flow due to the sources and sinks. The overall velocity field (2.1) then satisfies the no-penetration condition. Note that the boundary value problem for the scalar potential is a Neumann problem and requires a slightly different numerical treatment [Poncet, 2009, Gillis et al., 2018].

The continuous Dirichlet problem (3.12)–(3.13) can be solved via Green's theorem with boundary integrals. We will now solve the discrete version of this equation using the immersed boundary projection method, resulting in completely analogous operations.

3.2.1 Discrete surface and its immersion in the grid

We enforce the no-penetration condition at a finite number N_s of discrete surface forcing points; the space of scalar data on these Lagrangian points is denoted by $\mathcal{S}^{N_s}$. In particular, let us define $\mathsf{r}_x, \mathsf{r}_y \in \mathcal{S}^{N_s}$ as the vectors of x and y coordinates of the surface points. Each surface point p is associated with a small straight segment of length ΔS_p. Some of the calculations will require information about the local surface orientation. For this purpose, we define vectors $\mathsf{n}_x, \mathsf{n}_y \in \mathcal{S}^{N_s}$ of components of the discrete surface unit normals. We also define the space $\mathcal{V}^{N_s}$ to hold vector-valued data, such as velocity, on the immersed surface points. For convenience, let us also define unit vectors e_p on this space, equal to 1 at surface point p ($1 \leq p \leq N_s$) and zero at every other point. For example, the x coordinate of point p is picked out of the vector r_x by projection onto the pth unit vector:

$$\mathsf{e}_p^T \mathsf{r}_x. \tag{3.14}$$

Each of the surface point spaces has an associated inner product that includes the surface length, e.g.,

$$\langle \mathsf{s}_1, \mathsf{s}_2 \rangle_{\mathcal{S}^{N_s}} = \sum_p \Delta S_p s_{1,p} s_{2,p}, \tag{3.15}$$

for any $\mathsf{s}_1, \mathsf{s}_2 \in \mathcal{S}^{N_s}$, so that the inner product approximates a surface integral. Another vector we will make substantial use of in this paper is $\mathit{1} \in \mathcal{S}^{N_s}$, a vector of ones on all surface points.

From any vector $\mathsf{s} \in \mathcal{S}^{N_s}$, we can also form a diagonal $N_s \times N_s$ operator D_s with the entries of the vector along the diagonal. When this operator acts upon another vector $\mathsf{u} \in \mathcal{S}^{N_s}$, it represents the Hadamard (i.e., element-by-element) product of the two vectors, $\mathsf{D}_\mathsf{s} \mathsf{u} = \mathsf{s} \circ \mathsf{u} \in \mathcal{S}^{N_s}$. Note that $\mathsf{D}_\mathsf{s} \mathsf{u} = \mathsf{D}_\mathsf{u} \mathsf{s}$, and that $\mathsf{D}_\mathsf{s} \mathit{1} = \mathsf{s}$.

Similar to (3.9), surface data are immersed into the grid with the *regularization operator* ${}^{\mathcal{N}}\mathsf{R}_{\mathcal{S}} : \mathcal{S}^{N_s} \mapsto \mathcal{N}$. Grid data are interpolated onto the surface points with the *interpolation operator* ${}^{\mathcal{S}}\mathsf{E}_{\mathcal{N}} : \mathcal{N} \mapsto \mathcal{S}^{N_s}$. ${}^{\mathcal{N}}\mathsf{R}_{\mathcal{S}}$ can be constructed (and we will assume it has) so that it is the transpose of the interpolation operator, ${}^{\mathcal{N}}\mathsf{R}_{\mathcal{S}} = {}^{\mathcal{S}}\mathsf{E}_{\mathcal{N}}{}^{T}$, with respect to the grid and surface inner products defined in this paper. Furthermore, note that ${}^{\mathcal{N}}\mathsf{R}_{\mathcal{S}}$ and ${}^{\mathcal{S}}\mathsf{E}_{\mathcal{N}}$ can be constructed with a different choice for the discrete Dirac delta function than the one used for the vortex regularization. In this work, we use the smoothed three-point function from Yang et al. [2009] (figure 3.1) and we use a uniform spacing between the surface points.

3.2.2 The immersed surface potential flow problem

The surface's motion is specified by a velocity distribution $\boldsymbol{u}_b$, represented discretely by components $\mathsf{v}_x, \mathsf{v}_y \in \mathcal{S}^{N_s}$. For rigid bodies, this surface motion can be described by a streamfunction. Specifically, translation at velocity (U, V) and rotation at angular velocity Ω would be described equivalently by velocity components

$$\mathsf{v}_x = U\mathsf{1} - \Omega \mathsf{r}_y, \qquad \mathsf{v}_y = V\mathsf{1} + \Omega \mathsf{r}_x \tag{3.16}$$

or by a surface streamfunction $\mathsf{s}_b \in \mathcal{S}^{N_s}$:

$$\mathsf{s}_b = U\mathsf{r}_y - V\mathsf{r}_x - \frac{1}{2}\Omega\left(\mathsf{D}_{\mathsf{r}_x}\mathsf{r}_x + \mathsf{D}_{\mathsf{r}_y}\mathsf{r}_y\right). \tag{3.17}$$

The no-penetration condition can be imposed by setting the discrete streamfunction equal to that of the surface, up to a uniform value, $\mathsf{s}_0 \in \mathcal{S}^{N_s}$:

$${}^{\mathcal{S}}\mathsf{E}_{\mathcal{N}}\mathsf{s} = \mathsf{s}_b - {}^{\mathcal{S}}\mathsf{E}_{\mathcal{N}}\mathsf{s}_\infty - \mathsf{s}_0. \tag{3.18}$$

For later shorthand, we will denote the difference between the body motion streamfunction and interpolated uniform flow streamfunction by $\mathsf{s}_b' \equiv \mathsf{s}_b - {}^{\mathcal{S}}\mathsf{E}_{\mathcal{N}}\mathsf{s}_\infty$. This modified streamfunction simply consists of subtracting the components (U_∞, V_∞) of the uniform flow from (U, V) in (3.17). The uniform value s_0 is left unspecified and will later serve the role of enforcing a constraint on circulation. For now, we will suppose that it can be set arbitrarily.

The no-penetration constraint is enforced in the basic potential flow problem (3.1) with the help of a vector of Lagrange multipliers, $\mathsf{f} \in \mathcal{S}^{N_s}$, on the surface points. The modified potential flow problem is thus

$$\mathsf{L}\mathsf{s} = -\left(\mathsf{w} + {}^{\mathcal{N}}\mathsf{R}_{\mathcal{S}}\mathsf{f}\right). \tag{3.19}$$

In fact, by simple comparison with the vorticity w, it is clear that the vector f represents the strength of the discrete bound vortex sheet on the surface and serves as another source term of the Poisson equation. Suppose we consider the bound vortex sheet $\gamma(\xi)$ that emerges from the analogous continuous problem on the undiscretized surface, where ξ is the arc-length parameter along the surface. At each point p, the discrete solution f is approximately equal to this continuous solution:

$$\mathsf{e}_p^T\mathsf{f} \simeq \gamma(\xi_p). \tag{3.20}$$

The vector of Lagrange multipliers f is initially unknown. Thus, the potential flow problem in the presence of the impenetrable surface is

$$\begin{bmatrix} \mathsf{L} & {}^{\mathcal{N}}\mathsf{R}_{\mathcal{S}} \\ {}^{\mathcal{S}}\mathsf{E}_{\mathcal{N}} & 0 \end{bmatrix} \begin{pmatrix} \mathsf{s} \\ \mathsf{f} \end{pmatrix} = \begin{pmatrix} -\mathsf{w} \\ \mathsf{s}_b' - \mathsf{s}_0 \end{pmatrix}. \tag{3.21}$$

This problem (3.21) has the structure of a generic *saddle-point problem* [Benzi et al., 2005]. We will encounter many such systems in this work, so in appendix A we summarize a solution approach, based on block-LU decomposition. The generated solution algorithm of (3.21) is

$$\mathsf{L}\mathsf{s}^* = -\mathsf{w} \tag{3.22}$$

$$\mathsf{S}\mathsf{f} = \mathsf{s}_b' - \mathsf{s}_0 - {}^{\mathcal{S}}\mathsf{E}_{\mathcal{N}}\mathsf{s}^* \tag{3.23}$$

$$\mathsf{s} = \mathsf{s}^* - \mathsf{L}^{-1}\,{}^{\mathcal{N}}\mathsf{R}_{\mathcal{S}}\mathsf{f}, \tag{3.24}$$

where the Schur complement S is

$$\mathsf{S} = -\,{}^{\mathcal{S}}\mathsf{E}_{\mathcal{N}}\mathsf{L}^{-1}\,{}^{\mathcal{N}}\mathsf{R}_{\mathcal{S}}. \tag{3.25}$$

Based on the properties of the matrices comprising S, this operator is symmetric and negative definite, and therefore invertible. Its inverse S^{-1}, also symmetric, maps a surface distribution of streamfunction to a corresponding bound vortex sheet strength. Note that the computation of f through S^{-1} is sensitive to both the ratio of discrete surface spacing to the grid spacing $\Delta S/\Delta x$ and the choice of discrete Dirac delta function. On the one hand, small values for $\Delta S/\Delta x$ and discrete Dirac delta functions with small support generally both lead to more high-frequency noise in f. This is because the underlying continuous problem for S^{-1} is a Fredholm integral equation of the first kind, which is ill-posed. As a result, the discrete analogue problem is poorly-conditioned [Goza et al., 2016]. On the other hand, values for $\Delta S/\Delta x$ that are too high can lead to streamlines penetrating a surface. We found that values between one to four can provide a good balance between the smoothness of f and the accuracy of the streamlines near a surface.

We can describe this algorithm in words: First, solve for the intermediate streamfunction field, associated with vorticity in the fluid, but without regard for the presence of the surface. Second, find the bound vortex sheet whose associated streamfunction cancels the difference between the specified streamfunction on the surface and the intermediate streamfunction evaluated on the surface. Finally, correct the intermediate streamfunction field for the influence of the bound vortex sheet. A version of the Julia code that implements this algorithm, as well as the algorithms in the following sections, is available in the authors' Github repository [Beckers and Eldredge, 2021].

We give two examples of the streamfunction with a body present and show the associated vortex sheet strength. Figure 3.3 shows a vortex near a circular cylinder and figure 3.4 shows a circular cylinder that translates horizontally. In both cases, the vortex sheet strength is in good agreement with the analytical solution from potential flow theory.

For later use, we note that the solution of (3.21) can also be written in inverse form using equation (A.7):

$$\begin{pmatrix} \mathsf{s} \\ f \end{pmatrix} = \begin{bmatrix} \mathsf{L}^{-1} + \mathsf{L}^{-1}\,{}^{\mathcal{N}}\mathsf{R}_{\mathcal{S}}\mathsf{S}^{-1}\,{}^{\mathcal{S}}\mathsf{E}_{\mathcal{N}}\mathsf{L}^{-1} & -\mathsf{L}^{-1}\,{}^{\mathcal{N}}\mathsf{R}_{\mathcal{S}}\mathsf{S}^{-1} \\ -\mathsf{S}^{-1}\,{}^{\mathcal{S}}\mathsf{E}_{\mathcal{N}}\mathsf{L}^{-1} & \mathsf{S}^{-1} \end{bmatrix} \begin{pmatrix} -\mathsf{w} \\ s_b' - s_0 \end{pmatrix}. \tag{3.26}$$

The matrix operator in (3.26) is the inverse of the basic saddle-point system.

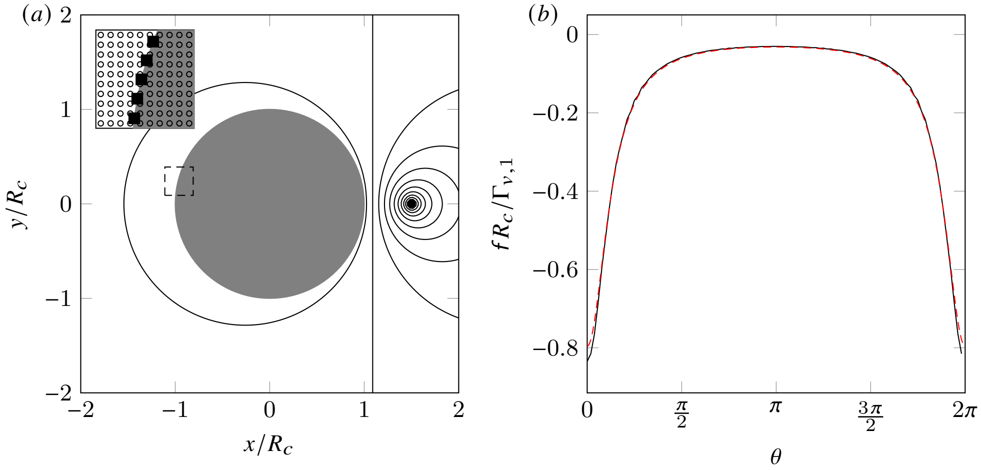


Figure 3.3: (a) Contours (——) of the discrete streamfunction for a point vortex (●) with strength $\Gamma_{v,1}$ at $(R_v, 0)$ near a circular cylinder of radius R_c and with a bound circulation $-\Gamma_{v,1}$. The inset figure shows a closeup of the nodes (○) and surface points (■) in the boxed area (- - - -). (b) The scaled discrete vortex sheet strength (——) as a function of the angle θ measured counter-clockwise from the positive x-axis. $k = 1$ corresponds to the right-most point on the surface and increases counterclockwise. Overlaid is the exact continuous solution (- - - -). The simulation is performed with $R_v/R_c = 3/2$, $\Delta x/R_c = 0.03$, and $\Delta S/\Delta x = 2$.

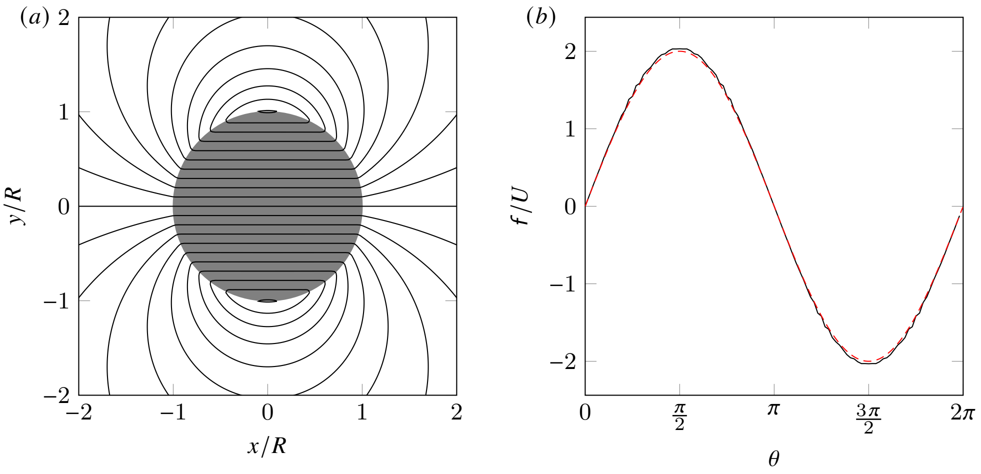


Figure 3.4: (a) Contours (——) of the discrete streamfunction for a horizontally translating circular cylinder with radius R, and (b) its scaled discrete vortex sheet strength (——). Overlaid is the exact continuous solution (- - - -). The simulation is performed with $\Delta x/R = 0.03$ and $\Delta S/\Delta x = 2$.

3.3 Non-uniqueness and discrete circulation

In two-dimensional potential flows, there is no unique solution to problem (3.21), since one can choose any value for the uniform value s_0 and still enforce the no-penetration condition. Equivalently, we can specify any *circulation* about the body and still enforce this condition. Let us determine the relationship between s_0 and circulation. For later use, let us write this uniform surface streamfunction as $\mathsf{s}_0 = s_0 \mathit{1}$, where s_0 is a single scalar value. The *discrete circulation* Γ_b about the body is given by the sum of the bound vortex sheet data and can be written compactly as

$$\Gamma_b = \langle \mathit{1}, \mathsf{f} \rangle_{\mathcal{S}^{N_s}}. \tag{3.27}$$

The discrete circulation of the vortex sheet in the solution (3.23) is

$$\Gamma_b = \left\langle \mathit{1}, \mathsf{S}^{-1} \left(\mathsf{s}_b' + {}^{\mathcal{S}}\mathsf{E}_{\mathcal{N}} \mathsf{L}^{-1} \mathsf{w} \right) \right\rangle_{\mathcal{S}^{N_s}} - s_0 \left\langle \mathit{1}, \mathsf{S}^{-1} \mathit{1} \right\rangle_{\mathcal{S}^{N_s}}. \tag{3.28}$$

Note that we can obtain the same expression if we would use s_0 as a Lagrange multiplier to enforce the constraint (3.27) as

$$\begin{bmatrix} \mathsf{L} & {}^{\mathcal{N}}\mathsf{R}_{\mathcal{S}} & 0 \\ {}^{\mathcal{S}}\mathsf{E}_{\mathcal{N}} & 0 & \mathit{1} \\ 0 & \mathit{1}^T\mathsf{D}_{\Delta\mathcal{S}} & 0 \end{bmatrix} \begin{pmatrix} \mathsf{s} \\ f \\ s_0 \end{pmatrix} = \begin{pmatrix} -\mathsf{w} \\ s_b' \\ \Gamma_b \end{pmatrix}, \tag{3.29}$$

where we use the fact that the inner product $\langle \mathit{1}, f \rangle_{\mathcal{S}^{N_s}}$ can, by its definition, be rewritten as $\mathit{1}^T\mathsf{D}_{\Delta\mathcal{S}}f$. Here, $\mathsf{D}_{\Delta\mathcal{S}}$ is a diagonal matrix containing the surface element arc lengths.We will use s_0 in a similar way in the next section to enforce the Kutta condition and demonstrate how to solve the associated saddle-point system.

The scalar factor $\langle \mathit{1}, \mathsf{S}^{-1}\mathit{1} \rangle_{\mathcal{S}^{N_s}}$ in expression (3.28) is a property of the set of points and their immersion into the Cartesian grid. Part of this factor, $\mathsf{S}^{-1}\mathit{1}$, represents the bound vortex sheet strength associated with a uniform, unit-strength streamfunction on the surface. This sheet has a particularly important role in some of the discussion to follow, so we will denote its strength by f_0:

$$f_0 \equiv \mathsf{S}^{-1}\mathit{1}. \tag{3.30}$$

The transpose of f_0, equal to $\mathit{1}^T\mathsf{S}^{-1}$, calculates the circulation of the associated bound vortex sheet when it acts upon a surface streamfunction. Thus, the factor $\langle \mathit{1}, \mathsf{S}^{-1}\mathit{1} \rangle_{\mathcal{S}^{N_s}}$ is the circulation associated with a uniform, unit-strength surface streamfunction. We will refer to this as Γ_0:

$$\Gamma_0 \equiv \langle \mathit{1}, \mathsf{S}^{-1}\mathit{1} \rangle_{\mathcal{S}^{N_s}} \equiv \langle \mathit{1}, f_0 \rangle_{\mathcal{S}^{N_s}} \equiv \langle f_0, \mathit{1} \rangle_{\mathcal{S}^{N_s}}. \tag{3.31}$$

We can rewrite the inner product $\langle f_0, \mathit{1} \rangle_{\mathcal{S}^{N_s}}$ as $f_0^T\mathsf{D}_{\Delta\mathcal{S}}\mathit{1}$, and we will define $\hat{f}_0 = \mathsf{D}_{\Delta\mathcal{S}}f_0$, for shorthand in what follows. By this notation, $\Gamma_0 = \hat{f}_0^{\,T}\mathit{1}$, and $\hat{f}_0^{\,T}$ applied to any surface streamfunction obtains the corresponding circulation.

Figure 3.5 shows the distribution of f_0 for elliptical cylinders with different aspect ratios. For a circular cylinder, the flow due to a uniform streamfunction on the body corresponds to the flow when the cylinder is replaced by a point vortex at its center. The streamlines are

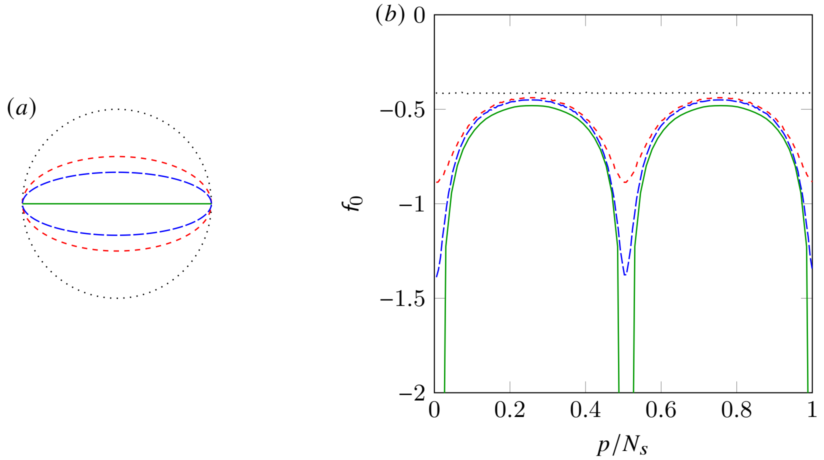


Figure 3.5: (*a*) Geometry and (*b*) the scaled bound vortex sheet strength f_0, associated with a uniform, unit-strength streamfunction, as a function of the surface point index p divided by the number of surface points N_s for elliptical cylinders with different aspect ratios (AR): ······, $AR = 1$; - - - -, $AR = 2$; — — —, $AR = 3$; ——, $AR = \infty$ (flat plate). The simulations are performed with $\Delta x/R = 0.004$ and $\Delta S/\Delta x = 3$.

concentric circles and the tangential velocity is constant at a given radius. The resulting vortex sheet strength f_0 therefore assumes a uniform distribution. The figure also demonstrates that when the aspect ratio increases, the distribution gradually shows stronger variations near the edges of the major axis, corresponding to an acceleration and deceleration of the flow when it passes those edges. The distribution eventually turns singular at the edges of a flat plate as the flow now has to navigate around a sharp corner, which will be discussed in more detail in the next section. To clearly illustrate the emergence of the singularities, the simulations in this figure are performed with an extremely fine grid ($\Delta x/R = 0.004$). Note that such fine grids are not needed in general, as is demonstrated by the other figures in this work.

The last term in (3.28) illustrates the direct relationship between the scalar value s_0 and the bound circulation Γ_b, and we identified s_0 as a means of setting the circulation. Before we use it in the next section to enforce the Kutta condition, we will use it here for an immediate purpose. The prescribed surface streamfunction s_b' (given by (3.17), with the uniform flow accounted for) may have some associated bound circulation, and it is desirable to adjust it by adding or subtracting a uniform value so that it has none. Equation (3.28) suggests that this circulation can be removed by setting s_0 to $\langle \mathit{1}, \mathsf{S}^{-1}\mathsf{s}_b' \rangle_{\mathcal{S}^{N_s}}/\Gamma_0 \equiv \hat{\mathsf{f}}_0^T \mathsf{s}_b'/\Gamma_0$ and then subtracting this value (multiplied by the uniform vector $\mathit{1}$) from s_b'. Overall, this process can be encapsulated in a *circulation removal* operator, P^Γ, that acts upon a surface streamfunction, $\mathsf{s} \in \mathcal{S}^{N_s}$,

$$\mathsf{P}^\Gamma \equiv \mathsf{I} - \frac{\mathit{1}\hat{\mathsf{f}}_0^T}{\Gamma_0}. \tag{3.32}$$

It is easy to verify that $\langle \mathsf{S}^{-1}\mathit{1}, \mathsf{P}^\Gamma \mathsf{s} \rangle_{\mathcal{S}^{N_s}} = \hat{\mathsf{f}}_0^T \mathsf{P}^\Gamma \mathsf{s} = 0$ for any $\mathsf{s} \in \mathcal{S}^{N_s}$, so that the circulation of any surface streamfunction acted upon by P^Γ is indeed zero. It is important to observe, also, that s_b' can be replaced by $\mathsf{P}^\Gamma \mathsf{s}_b'$ without affecting the nature of the no-penetration condition. We also note that the composite operator $\mathsf{S}^{-1}\mathsf{P}^\Gamma$ is symmetric, just as S^{-1} is, since

$$\mathsf{S}^{-1}\mathsf{P}^\Gamma = \mathsf{S}^{-1} - \hat{\mathsf{f}}_0 \hat{\mathsf{f}}_0^T. \tag{3.33}$$

3.4 The Kutta condition

For surfaces that contain convex edges, the vortex sheet strength assumes a singular behavior in the vicinity of these edges, with a strength that depends on the interior angle of the edge: sharper edges have more singular behavior. In the discrete representation of the surface, edges are only approximately represented by the sudden disruptions of positions in clusters of adjacent points, without any information about the surface normals. The behavior in this discrete form is not quite singular, but the solution of (3.23) nonetheless exhibits a large and rapid change of amplitude.

If we seek to eliminate this behavior, we must first have some means of exposing it. In fact, for any discretized surface, the essence of this nearly-singular behavior lies in the vector f_0, and all other bound vortex sheets associated with the same surface share the same nearly-singular behavior. Thus, we will use a multiplicative decomposition of the vortex sheet strength:

$$\mathsf{f} = \mathsf{f}_0 \circ \tilde{\mathsf{f}}, \tag{3.34}$$

where $\circ$ is the Hadamard product. This decomposed form isolates the singular behavior into f_0, and $\tilde{\mathsf{f}}$ is a relatively smoother vector of surface point data. In the regularization operation on f, we can absorb f_0 into ${}^{\mathcal{N}}\mathsf{R}_{\mathcal{S}}$, first noting that the Hadamard product can alternatively be written with the help of a diagonal matrix,

$$\mathsf{f} = \mathsf{f}_0 \circ \tilde{\mathsf{f}} = \mathsf{D}_{\mathsf{f}_0}\tilde{\mathsf{f}}. \tag{3.35}$$

Then, we can define a re-scaled regularization operator,

$${}^{\mathcal{N}}\mathsf{R}_{\mathcal{S}}\mathsf{f} = {}^{\mathcal{N}}\mathsf{R}_{\mathcal{S}}\mathsf{D}_{\mathsf{f}_0}\tilde{\mathsf{f}} = {}^{\mathcal{N}}\tilde{\mathsf{R}}_{\mathcal{S}}\tilde{\mathsf{f}}. \tag{3.36}$$

The re-scaled operator ${}^{\mathcal{N}}\tilde{\mathsf{R}}_{\mathcal{S}} = {}^{\mathcal{N}}\mathsf{R}_{\mathcal{S}}\mathsf{D}_{\mathsf{f}_0}$ can, in turn, be absorbed into the Schur complement, defining $\tilde{\mathsf{S}} = -{}^{\mathcal{S}}\mathsf{E}_{\mathcal{N}}\mathsf{L}^{-1}\,{}^{\mathcal{N}}\tilde{\mathsf{R}}_{\mathcal{S}} = \mathsf{S}\mathsf{D}_{\mathsf{f}_0}$. A useful property of $\tilde{\mathsf{S}}$ is that it preserves uniform vectors:

$$\tilde{\mathsf{S}}\mathit{1} = \mathsf{S}\mathsf{D}_{\mathsf{f}_0}\mathit{1} = \mathsf{S}\mathsf{f}_0 = \mathit{1}, \qquad \tilde{\mathsf{S}}^{-1}\mathit{1} = \mathit{1}. \tag{3.37}$$

The decomposition of the vortex sheet strength is demonstrated in figure 3.6 for a flat plate in a uniform flow. We plot the vortex sheet strength against the scaled coordinate ξ, which varies along the plate from -1 at the leading edge to 1 at the trailing edge. As expected, the vortex sheet strength f shows large amplitude variations at the sharp leading and trailing edges, corresponding to the singularities in the distribution of the continuous vortex sheet strength. By use of decomposition (3.34), these discrete singularities are retained in f_0 and we are left with a non-singular $\tilde{\mathsf{f}}$, which varies almost linearly with ξ.

The Kutta condition corresponds to annihilating the nearly-singular behavior at a surface point. At such points, we will set the corresponding value of $\tilde{\mathsf{f}}$ to zero. Suppose we wish to enforce the Kutta condition at an edge corresponding to surface point k. The condition is

$$\mathsf{e}_k^T\tilde{\mathsf{f}} = 0. \tag{3.38}$$

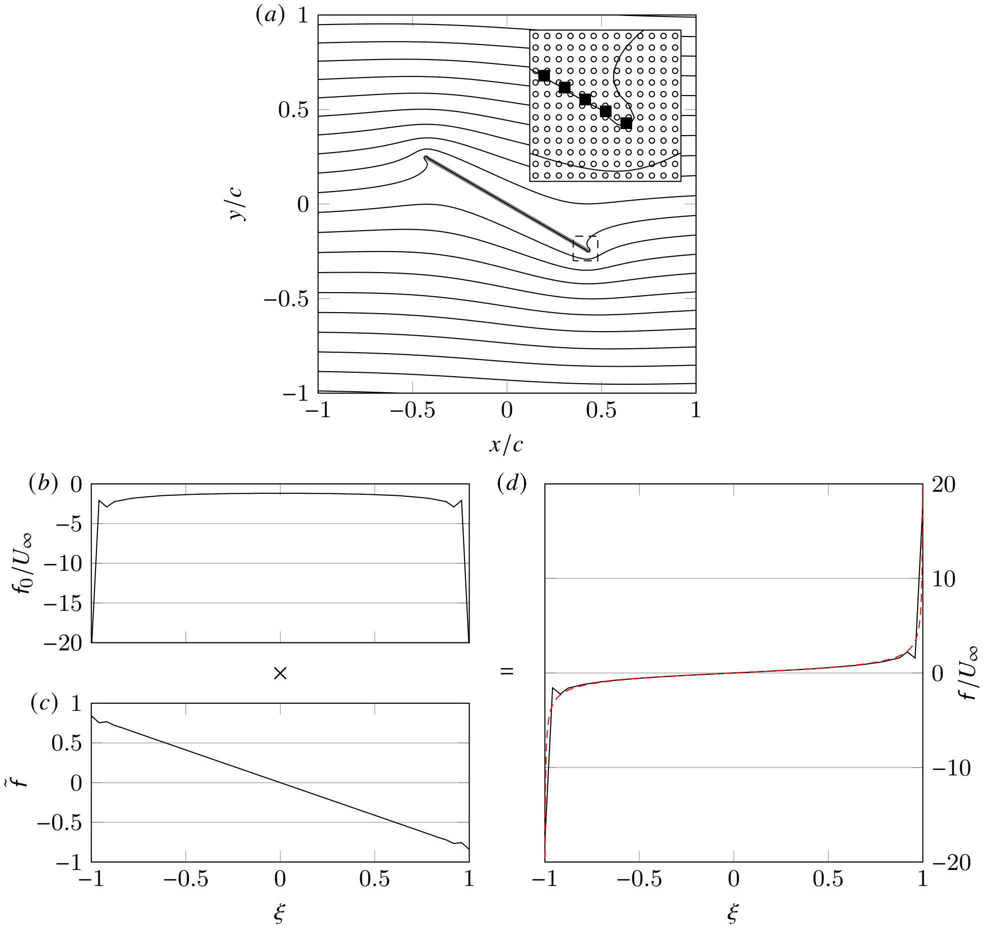


Figure 3.6: (*a*) Contours (——) of the discrete streamfunction for a flat plate of length c, at 30° in a uniform flow U_∞ without enforcement of the Kutta condition. The inset figure shows a closeup of the nodes (○) and surface points (■) in the boxed area (- - - -). The point-wise product of (*b*) the discrete vortex sheet strength associated with a uniform, unit-strength streamfunction on the body, and (*c*) the non-singular vector that results from re-scaling the regularization operator, composes (*d*) the discrete vortex sheet strength. Overlaid is the exact continuous solution (- - - -). The simulation is performed with $\Delta x/c = 0.01$ and $\Delta S/\Delta x = 2$.

3.4.1 Using the Kutta condition in a steady-state problem

We will first take the steady-state approach to enforce the Kutta condition: allow the bound circulation to be set appropriately, with the implicit understanding that there is a starting vortex of equal and opposite circulation at infinity that preserves the Kelvin circulation theorem. The Lagrange multiplier for this constraint will not be Γ_b, but s_0, similar to (3.29). We also use the circulation removal operator to adjust the imposed surface streamfunction:

$$\begin{bmatrix} \mathsf{L} & {}^{\mathcal{N}}\tilde{\mathsf{R}}_{\mathcal{S}} & 0 \\ {}^{\mathcal{S}}\mathsf{E}_{\mathcal{N}} & 0 & \mathit{1} \\ 0 & \mathsf{e}_k^T & 0 \end{bmatrix} \begin{pmatrix} \mathsf{s} \\ \tilde{\mathsf{f}} \\ s_0 \end{pmatrix} = \begin{pmatrix} -\mathsf{w} \\ \mathsf{P}^{\Gamma}\mathsf{s}_b' \\ 0 \end{pmatrix} \tag{3.39}$$

This block system, like the earlier one in (3.21), has a saddle point form, and we can reduce it by the same block-LU decomposition to develop a solution algorithm. We will interpret it in the general form (A.3), with the upper left 2 × 2 block taking the role of $\mathcal{A}$, the solution vector $\mathbf{x}$ and constraint force $\mathbf{y}$ set, respectively, to

$$\mathbf{x} = \begin{pmatrix} \mathsf{s} \\ \tilde{\mathsf{f}} \end{pmatrix}, \qquad \mathbf{y} = s_0, \tag{3.40}$$

the remaining operators set to

$$\mathcal{B}_2 = \begin{bmatrix} 0 & \mathsf{e}_k^T \end{bmatrix}, \qquad \mathcal{B}_1^T = \begin{bmatrix} 0 \\ \mathit{1} \end{bmatrix}, \qquad \mathcal{C} = 0, \tag{3.41}$$

and the right-hand side vectors set to

$$\mathbf{r_1} = \begin{pmatrix} -\mathsf{w} \\ \mathsf{P}^{\Gamma}\mathsf{s}_b' \end{pmatrix}, \qquad \mathbf{r_2} = 0. \tag{3.42}$$

We note that block $\mathcal{A}$ has the original form of the system before the Kutta constraint (3.21), though with the slight modification of a re-scaled regularization operator, and we already have the inverse of $\mathcal{A}$ available from (3.26). The solution of this original system forms the intermediate solution of the full system endowed with the Kutta condition:

$$\tilde{f}^* = \tilde{\mathsf{S}}^{-1}\left(\mathsf{P}^{\Gamma} s_b' + {}^{\mathcal{S}}\mathsf{E}_{\mathcal{N}}\mathsf{L}^{-1}\mathsf{w}\right), \qquad \mathsf{s}^* = -\mathsf{L}^{-1}\left(\mathsf{w} + {}^{\mathcal{N}}\tilde{\mathsf{R}}_{\mathcal{S}}\tilde{f}^*\right). \tag{3.43}$$

Then, using the general procedure outlined in appendix A, the solution of the full system (3.39) is easy to develop; its Schur complement is simply

$$\mathcal{S} = -1. \tag{3.44}$$

Applying the general solution equations, and using the property (3.37) to simplify the resulting operators, it can be shown that the solution is

$$s_0 = e_k^T \tilde{f}^* \tag{3.45}$$

$$\begin{pmatrix} \mathsf{s} \\ \tilde{f} \end{pmatrix} = \begin{pmatrix} \mathsf{s}^* \\ \tilde{f}^* \end{pmatrix} - \begin{bmatrix} -\mathsf{L}^{-1}\,{}^{\mathcal{N}}\tilde{\mathsf{R}}_{\mathcal{S}}\mathit{1} \\ \mathit{1} \end{bmatrix} e_k^T \tilde{f}^*. \tag{3.46}$$

The entire solution can be written more compactly as

$$\tilde{f} = \mathsf{P}_k^K \tilde{\mathsf{S}}^{-1}\left(\mathsf{P}^{\Gamma} s_b' + {}^{\mathcal{S}}\mathsf{E}_{\mathcal{N}}\mathsf{L}^{-1}\mathsf{w}\right), \tag{3.47}$$

$$\mathsf{s} = -\mathsf{L}^{-1}\left(\mathsf{w} + {}^{\mathcal{N}}\tilde{\mathsf{R}}_{\mathcal{S}}\tilde{f}\right), \tag{3.48}$$

where we have defined the *Kutta projection operator*,

$$\mathsf{P}_k^K \equiv \mathsf{I} - \mathit{1} e_k^T, \tag{3.49}$$

which acts upon the (non-singular part of the) bound vortex sheet vector, subtracting the value at point k from every point, including at k itself.

Note that the Lagrange multiplier for the Kutta condition takes the simple value given by equation (3.45), revealing that the additional streamfunction on the surface is exactly the value of the intermediate bound vortex sheet at the Kutta point k.

The application of the Kutta condition to a steady-state problem is demonstrated in Figure 3.7 on the flat plate problem that was introduced in the previous section. By constraining the trailing edge point of $\tilde{f}$, its whole distribution is shifted upward such that the trailing-edge value equals zero. The resulting streamfunction indicates that the flow then indeed leaves the trailing edge smoothly.

Figure 3.7 also shows a spatial grid refinement analysis of the non-singular part of the vortex sheet strength. For this analysis, we can use a multiplicative decomposition for the continuous vortex sheet strength, $\gamma(\xi) = \gamma_0(\xi)\tilde{\gamma}(\xi)$, analogous to (3.34). The continuous counterpart of f_0, can be found as the bound vortex sheet strength of a circular cylinder with a point vortex of strength Γ_0 at its center, conformally mapped to a flat plate. Due to the definition of f_0, the circulation Γ_0 depends on the grid spacing. Therefore, if we define γ_0 as the continuous counterpart of f_0/Γ_0, we obtain the grid-independent solution $\tilde{\gamma}(\xi) = -\pi c U_\infty \sin(\alpha)(1-\xi)$ as the continuous counterpart of $\Gamma_0\tilde{\mathsf{f}}$ for a flat plate with the Kutta condition enforced. We can then define the error for the non-singular part of the vortex sheet strength as $\epsilon_{\Gamma_0\tilde{\mathsf{f}}} = \|\Gamma_0\tilde{\mathsf{f}} - \tilde{\gamma}(\xi(\mathsf{r}_x,\mathsf{r}_y)\|_2/\|\tilde{\gamma}(\xi(\mathsf{r}_x,\mathsf{r}_y)\|_2$. The refinement analysis verifies that the immersed-boundary projection method using the vortex sheet strength decomposition is approximately first-order accurate in Δx. Furthermore, the figure shows the streamlines near the plate, the vortex sheet strength, and the error for four different values of $\Delta S/\Delta x$. These confirm that a lower value increases the noise in the vortex sheet strength, but a higher value can lead to the streamlines penetrating the surface, as discussed previously.

3.4.2 Using the Kutta condition to set a new vortex element

In the previous section, we used the Kutta condition to set the bound circulation but did not explicitly create a new vortex element. This vortex element was assumed to lie at infinity so that its effect was negligible except insofar as it left equal but opposite circulation about the body.

In this section, we will create a new vortex element in the vicinity of the edge at which we are applying the Kutta condition. We will thus seek to establish the strength of this new element and to do so in such a manner that the overall circulation of the flow is conserved. Once the element is created, it will be allowed to advect with the local fluid velocity.

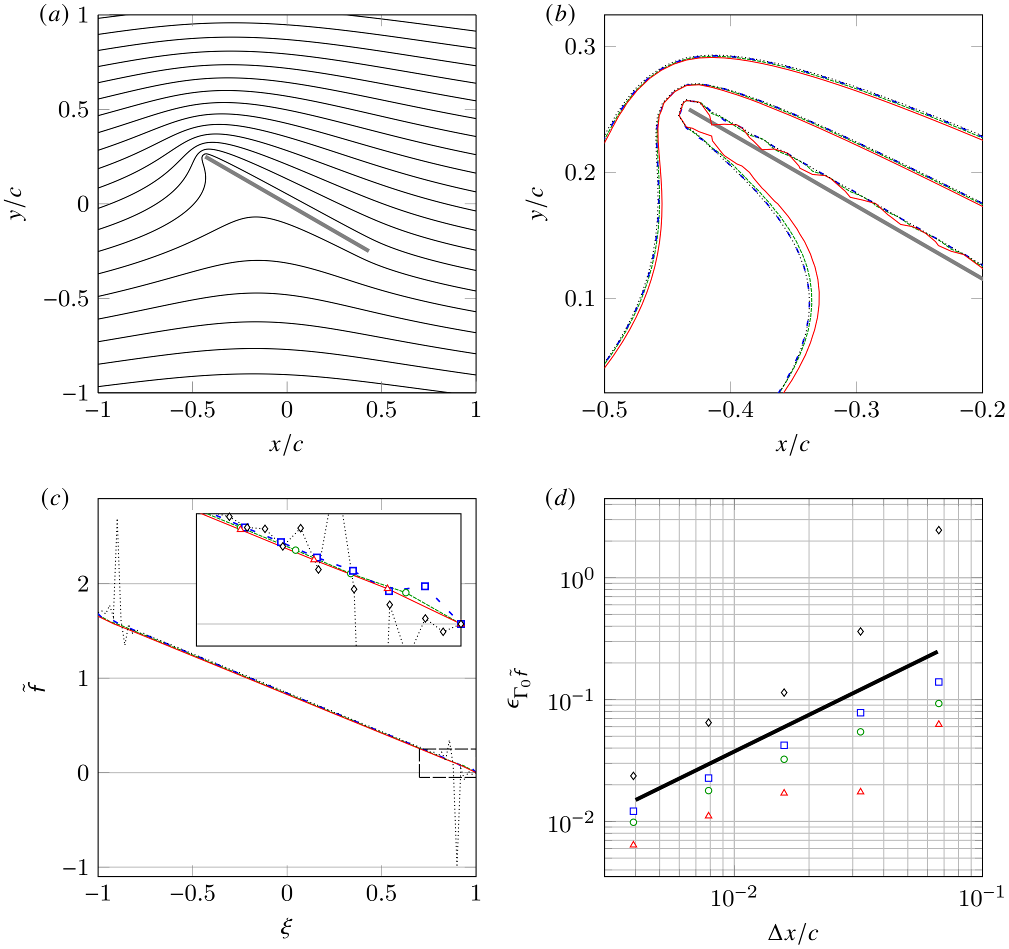


Figure 3.7: (a) Contours (——) of the steady, discrete streamfunction for a flat plate of length c at 30° in a uniform flow with enforcement of the Kutta condition at the trailing edge. (b) Zoom on the leading edge showing the streamlines for three values of the streamfunction, (c) the non-singular part of the associated discrete vortex sheet strength for $\Delta x/c = 0.01$ with an inset enlarging the boxed area (— — —), and (d) the variation of its error with grid spacing for different values for $\Delta S/\Delta x$: ······ and ◇, $\Delta S/\Delta x = 1$; – – – and □, $\Delta S/\Delta x = 2$; - - - - and ○, $\Delta S/\Delta x = 3$; —— and △, $\Delta S/\Delta x = 4$. Overlaid is an error (——) that scales as Δx.

Let us assume that the new vortex element (which we label with the subscript 1) is introduced at some point in physical space, and that its immersion into the Cartesian grid is described by d_1 and that its strength (i.e., its circulation) is $\Gamma_{v,1}$. Thus, the fluid vorticity after this new element's introduction can be written as

$$\mathsf{w} + \Gamma_{v,1}\mathsf{d}_1. \tag{3.50}$$

The Kutta condition (3.38) is still to be enforced. We also seek to ensure that the total circulation is zero to satisfy Kelvin's circulation theorem. (We are assuming that the flow has started from rest.) Let us denote the circulation of the existing fluid vorticity w by

$$\Gamma_{\mathsf{w}} = \langle \mathsf{1}, \mathsf{w} \rangle_{\mathcal{N}}. \tag{3.51}$$

Then, the circulation constraint is

$$\langle \mathit{1}, f \rangle_{\mathcal{S}^{N_s}} + \Gamma_{v,1} + \Gamma_{\mathsf{w}} = 0. \tag{3.52}$$

The circulation of the bound vortex sheet f can be re-written in terms of the non-singular part of the sheet as $\langle \mathit{1}, f \rangle_{\mathcal{S}^{N_s}} = \langle f_0, \tilde{f} \rangle_{\mathcal{S}^{N_s}} = \hat{f}_0^{\,T} \tilde{f}$.

With these two constraints, the overall saddle point system of equations is

$$\begin{bmatrix} \mathsf{L} & {}^{\mathcal{N}}\tilde{\mathsf{R}}_{\mathcal{S}} & 0 & \mathsf{d}_1 \\ {}^{\mathcal{S}}\mathsf{E}_{\mathcal{N}} & 0 & \mathit{1} & 0 \\ 0 & e_k^T & 0 & 0 \\ 0 & \hat{f}_0^{\,T} & 0 & 1 \end{bmatrix} \begin{pmatrix} \mathsf{s} \\ \tilde{f} \\ s_0 \\ \Gamma_{v,1} \end{pmatrix} = \begin{pmatrix} -\mathsf{w} \\ \mathsf{P}^{\Gamma} s_b' \\ 0 \\ -\Gamma_{\mathsf{w}} \end{pmatrix}. \tag{3.53}$$

Again, the basic saddle-point matrix constitutes the upper left 2×2 block $\mathcal{A}$ and the solution vector $\mathbf{x}$ is as before. The constraint force vector is

$$\mathbf{y} = \begin{pmatrix} s_0 \\ \Gamma_{v,1} \end{pmatrix}, \tag{3.54}$$

and the remaining vectors and operators are now

$$\mathbf{r}_2 = \begin{pmatrix} 0 \\ -\Gamma_{\mathsf{w}} \end{pmatrix}, \qquad \mathcal{B}_2 = \begin{bmatrix} 0 & e_k^T \\ 0 & \hat{f}_0^{\,T} \end{bmatrix}, \qquad \mathcal{B}_1^T = \begin{bmatrix} 0 & \mathsf{d}_1 \\ \mathit{1} & 0 \end{bmatrix}, \qquad \mathcal{C} = -\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}. \tag{3.55}$$

The solution algorithm follows, once again, from the equations in appendix A. After carrying out the block matrix multiplications, it can be shown that the Schur complement (A.2) is the 2 × 2 matrix

$$\mathcal{S} = \begin{bmatrix} -1 & \mathsf{e}_k^T \tilde{\mathsf{f}}_1 \\ -\hat{\mathsf{f}}_0^T \mathit{1} & 1 + \hat{\mathsf{f}}_0^T \tilde{\mathsf{f}}_1, \end{bmatrix} \tag{3.56}$$

where, for convenience, we have defined

$$\tilde{\mathsf{f}}_1 = \tilde{\mathsf{S}}^{-1}\, {}^{\mathcal{S}}\mathsf{E}_{\mathcal{N}} \mathsf{L}^{-1} \mathsf{d}_1, \tag{3.57}$$

which represents the (non-singular part of the) strength of the vortex sheet that “reacts” to the presence of a unit-strength vortex d_1 immersed into the grid, canceling that vortex's induced velocity on the surface. The term $\hat{\mathsf{f}}_0^T \tilde{\mathsf{f}}_1$ represents this sheet's bound circulation and $\mathsf{e}_k^T \tilde{\mathsf{f}}_1$ is its contribution to the Kutta condition at point k. The problem (A.6) for the constraint forces s_0 and $\Gamma_{v,1}$ is then

$$\begin{bmatrix} -1 & \mathsf{e}_k^T \tilde{\mathsf{f}}_1 \\ -\hat{\mathsf{f}}_0^T \mathit{1} & 1 + \hat{\mathsf{f}}_0^T \tilde{\mathsf{f}}_1, \end{bmatrix} \begin{pmatrix} s_0 \\ \Gamma_{v,1} \end{pmatrix} = \begin{pmatrix} -\mathsf{e}_k^T \tilde{\mathsf{f}}^* \\ -\Gamma_{\mathsf{w}} - \hat{\mathsf{f}}_0^T \tilde{\mathsf{f}}^* \end{pmatrix}, \tag{3.58}$$

where the intermediate solution $\tilde{\mathsf{f}}^*$ is available from (3.43).

The determinant of this Schur complement matrix is $-1 - \hat{\mathsf{f}}_0^T \mathsf{P}_k^K \tilde{\mathsf{f}}_1$, which represents the negative of the circulation of the unit vortex and its associated vortex sheet, after the Kutta condition has been enforced on this sheet. It is straightforward then to calculate the strength of the new vortex $\Gamma_{v,1}$ and the additional uniform surface streamfunction, s_0:

$$\Gamma_{v,1} = \frac{-\Gamma_{\mathsf{w}} - \hat{\mathsf{f}}_0^T \mathsf{P}_k^K \tilde{\mathsf{f}}^*}{1 + \hat{\mathsf{f}}_0^T \mathsf{P}_k^K \tilde{\mathsf{f}}_1}, \qquad s_0 = \mathsf{e}_k^T \left(\tilde{\mathsf{f}}^* + \Gamma_{v,1} \tilde{\mathsf{f}}_1 \right). \tag{3.59}$$

From these, we can then obtain the vortex sheet strength and the fluid streamfunction,

$$\tilde{\mathsf{f}} = \mathsf{P}_k^K \left(\tilde{\mathsf{f}}^* + \Gamma_{v,1} \tilde{\mathsf{f}}_1 \right), \qquad \mathsf{s} = -\mathsf{L}^{-1} \left(\mathsf{w} + \Gamma_{v,1} \mathsf{d}_1 + {}^{\mathcal{N}}\tilde{\mathsf{R}}_{\mathcal{S}} \tilde{\mathsf{f}} \right). \tag{3.60}$$

The intermediate solution, which corresponds to the flow associated with existing vorticity, is corrected here with the new vortex 1 to enforce the Kutta condition at the point k.

We now first demonstrate the enforcement of the Kutta condition in unsteady flow on the flat plate problem with a point vortex near the trailing edge to enforce the Kutta condition at that edge. We position the point vortex at a distance $10\Delta t U_\infty$ from the edge in the direction of the free stream, perpendicular to the plate. Figure 3.8 demonstrates that, because of the proximity of the point vortex to the flat plate, $\tilde{\mathsf{f}}$ exhibits a quick variation at the surface points that lie closest to the point vortex. The value at the trailing edge point itself is still constrained to zero and the flow again leaves the edge smoothly. This situation corresponds to the flow right after impulsively starting a uniform flow around a flat plate and the point vortex now represents the starting vortex. The second demonstration of the method (figure 3.9) is the repeated enforcement of the Kutta condition on a NACA0012 airfoil while advancing the positions of the point vortices in time after impulsively starting the flow. This simulation, and all following simulations, use forward Euler time-stepping, unless noted otherwise, and new point vortices are inserted at one-third of the way from the edge to the last released vortex from that edge. The streamlines in the figure show that a strong initial vortex and weaker subsequent vortices were created and convected downstream. In turn, at each time step, the airfoil obtained a circulation that enforces the Kutta condition.

3.4.3 Applying more than one Kutta condition on a body

Suppose we wish to enforce the Kutta condition at two edges of the body—at points k_1 and k_2—instead of one. Each such point has a constraint,

$$\mathsf{e}_{k_j}^T \tilde{\mathsf{f}} = 0, \quad j = 1, 2. \tag{3.61}$$

For two such constraints, we need two Lagrange multipliers: the strengths of two new vortices, $\Gamma_{v,1}$ and $\Gamma_{v,2}$, immersed into the grid with d_1 and d_2, respectively; and we still need the Lagrange multiplier s_0 to ensure that Kelvin's circulation theorem is also enforced. The

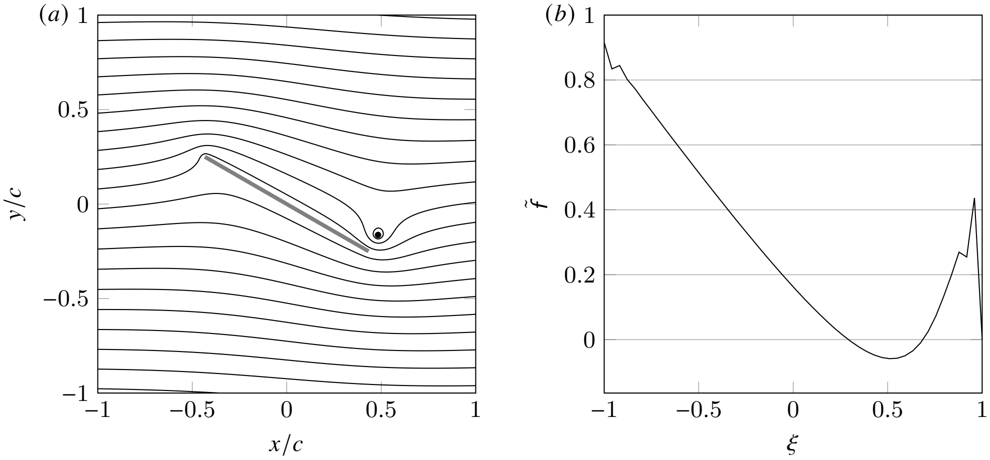


Figure 3.8: (*a*) Contours (——) of the unsteady, discrete streamfunction for a flat plate of length c at 30° in a uniform flow with release of vorticity into a point vortex (●) for enforcement of the Kutta condition at the trailing edge. (*b*) The non-singular part of the associated discrete vortex sheet strength. The simulation is performed with $\Delta x/c = 0.01$ and $\Delta S/\Delta x = 2$.

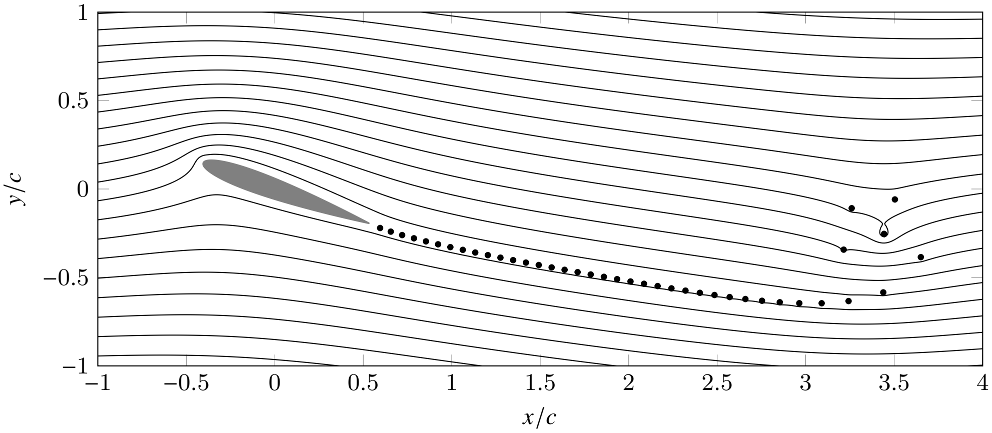


Figure 3.9: Contours (——) of the unsteady, discrete streamfunction for a NACA0012 airfoil with chord length c at 20°, 3 convective times after impulsively starting a uniform flow U_∞. Vorticity is released into a stream of point vortices (●) for enforcement of the Kutta condition at the trailing edge. The simulation is performed with $\Delta x/c = 0.01$, $\Delta S/\Delta x = 1.5$, and $\Delta t U_\infty/c = 0.075$.

system in the previous section is thus easily generalized to the following:

$$\begin{bmatrix} \mathsf{L} & {}^{\mathcal{N}}\tilde{\mathsf{R}}_{\mathcal{S}} & 0 & \mathsf{d}_1 & \mathsf{d}_2 \\ {}^{\mathcal{S}}\mathsf{E}_{\mathcal{N}} & 0 & \mathit{1} & 0 & 0 \\ 0 & \mathsf{e}_{k_1}^T & 0 & 0 & 0 \\ 0 & \mathsf{e}_{k_2}^T & 0 & 0 & 0 \\ 0 & \hat{\mathsf{f}}_0^T & 0 & 1 & 1 \end{bmatrix} \begin{pmatrix} \mathsf{s} \\ \tilde{\mathsf{f}} \\ s_0 \\ \Gamma_{v,1} \\ \Gamma_{v,2} \end{pmatrix} = \begin{pmatrix} -\mathsf{w} \\ \mathsf{P}^{\Gamma}\mathsf{s}_b' \\ 0 \\ 0 \\ -\Gamma_{\mathsf{w}} \end{pmatrix}. \tag{3.62}$$

The system is reduced in the same manner as before, with the same intermediate solution obtained from the basic system (3.21). Now, the Schur complement problem for the constraint forces takes the form

$$\begin{bmatrix} -1 & \mathsf{e}_{k_1}^T\tilde{\mathsf{f}}_1 & \mathsf{e}_{k_1}^T\tilde{\mathsf{f}}_2 \\ -1 & \mathsf{e}_{k_2}^T\tilde{\mathsf{f}}_1 & \mathsf{e}_{k_2}^T\tilde{\mathsf{f}}_2 \\ -\hat{\mathsf{f}}_0^T\mathit{1} & 1+\hat{\mathsf{f}}_0^T\tilde{\mathsf{f}}_1 & 1+\hat{\mathsf{f}}_0^T\tilde{\mathsf{f}}_2, \end{bmatrix} \begin{pmatrix} s_0 \\ \Gamma_{v,1} \\ \Gamma_{v,2} \end{pmatrix} = \begin{pmatrix} -\mathsf{e}_{k_1}^T\tilde{\mathsf{f}}^* \\ -\mathsf{e}_{k_2}^T\tilde{\mathsf{f}}^* \\ -\Gamma_{\mathsf{w}} - \hat{\mathsf{f}}_0^T\tilde{\mathsf{f}}^* \end{pmatrix}, \tag{3.63}$$

where we have now defined bound vortex sheets associated with each of the two new vortices (with unit strengths):

$$\tilde{\mathsf{f}}_j = \tilde{\mathsf{S}}^{-1}\,{}^{\mathcal{S}}\mathsf{E}_{\mathcal{N}}\mathsf{L}^{-1}\mathsf{d}_j, \tag{3.64}$$

for $j = 1, 2$. It is interesting to note that, if we take the difference between the two Kutta constraints, we obtain

$$\left(\mathsf{e}_{k_1}^T - \mathsf{e}_{k_2}^T\right)\left(\tilde{\mathsf{f}}^* + \Gamma_{v,1}\tilde{\mathsf{f}}_1 + \Gamma_{v,2}\tilde{\mathsf{f}}_2\right) = 0. \tag{3.65}$$

It can be shown that this Schur complement problem can be split into

$$\begin{bmatrix} 1+\hat{\mathsf{f}}_0^T\mathsf{P}_{k_1}^K\tilde{\mathsf{f}}_1 & 1+\hat{\mathsf{f}}_0^T\mathsf{P}_{k_1}^K\tilde{\mathsf{f}}_2 \\ 1+\hat{\mathsf{f}}_0^T\mathsf{P}_{k_2}^K\tilde{\mathsf{f}}_1 & 1+\hat{\mathsf{f}}_0^T\mathsf{P}_{k_2}^K\tilde{\mathsf{f}}_2 \end{bmatrix} \begin{pmatrix} \Gamma_{v,1} \\ \Gamma_{v,2} \end{pmatrix} = -\begin{pmatrix} \Gamma_{\mathsf{w}} + \hat{\mathsf{f}}_0^T\mathsf{P}_{k_1}^K\tilde{\mathsf{f}}^* \\ \Gamma_{\mathsf{w}} + \hat{\mathsf{f}}_0^T\mathsf{P}_{k_2}^K\tilde{\mathsf{f}}^* \end{pmatrix} \tag{3.66}$$

and

$$s_0 = \frac{1}{2}\left(\mathsf{e}_{k_1}^T + \mathsf{e}_{k_2}^T\right)\left(\tilde{\mathsf{f}}^* + \Gamma_{v,1}\tilde{\mathsf{f}}_1 + \Gamma_{v,2}\tilde{\mathsf{f}}_2\right). \tag{3.67}$$

The latter equation, when combined with (3.65), reveals that the value of the vortex sheet strength $\tilde{\mathsf{f}}^* + \Gamma_{v,1}\tilde{\mathsf{f}}_1 + \Gamma_{v,2}\tilde{\mathsf{f}}_2$ is the same at both Kutta points and equal to s_0.

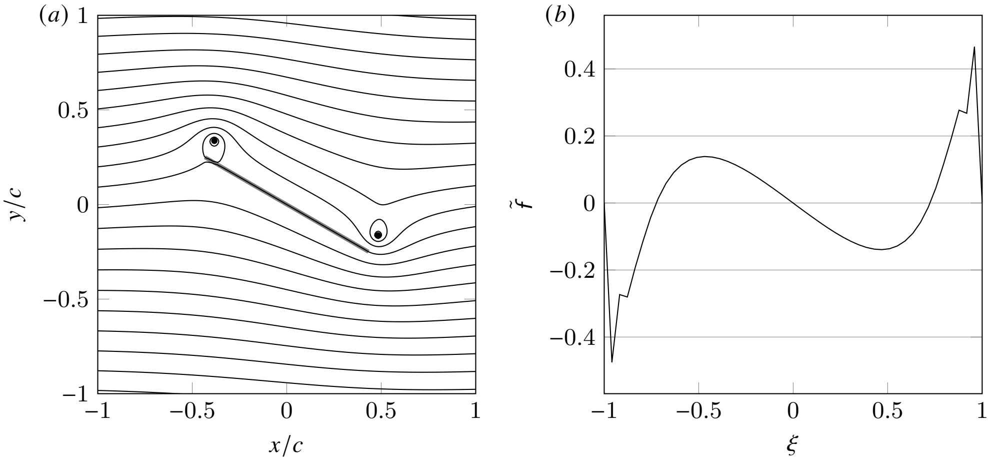


Figure 3.10: (*a*) Contours (——) of the unsteady, discrete streamfunction for a flat plate of length c at 30° in a uniform flow with release of vorticity into two point vortices (●) for enforcement of the Kutta condition at both edges. (*b*) The non-singular part of the associated discrete vortex sheet strength. The simulation is performed with $\Delta x/c = 0.01$ and $\Delta S/\Delta x = 2$.

Equation (3.66) can be solved easily for the strengths of the two new point vortices. Then, the solution for the vortex sheet strength and streamfunction are

$$\tilde{\mathsf{f}} = \frac{1}{2}\left(\mathsf{P}^{K}_{k_1} + \mathsf{P}^{K}_{k_2}\right)\left(\tilde{\mathsf{f}}^{*} + \Gamma_{v,1}\tilde{\mathsf{f}}_1 + \Gamma_{v,2}\tilde{\mathsf{f}}_2\right), \qquad \mathsf{s} = -\mathsf{L}^{-1}\left(\mathsf{w} + \Gamma_{v,1}\mathsf{d}_1 + \Gamma_{v,2}\mathsf{d}_2 + {}^{\mathcal{N}}\tilde{\mathsf{R}}_{\mathcal{S}}\tilde{\mathsf{f}}\right) \tag{3.68}$$

We now apply this method in figure 3.10 to enforce the Kutta condition at the leading and trailing edge of our flat plate problem. We position a point vortex close to each edge and observe again that $\tilde{\mathsf{f}}$ shows strong variation at the surface points closest to the two point vortices. The contours of the streamfunction indicate that the flow indeed leaves the edges smoothly. Like the previous case, this solution corresponds to the flow right after impulsively starting a uniform flow around a flat plate, but unlike the previous case, the flow now separates at the leading edge.

It should be observed that these solutions are posed in a manner easily extensible to an arbitrary number of edges.

3.5 Generalized edge condition

In the previous section, we demonstrated the means of annihilating the (nearly) singular behavior at edges on a discretized surface. In some cases, our desire is not to annihilate this behavior, but simply to keep it within some bounds. In the analytical treatment of potential flow problems, this objective is served by placing an inequality constraint on the *edge suction parameter* [Ramesh et al., 2014, Darakananda and Eldredge, 2019, Eldredge, 2019]. That parameter is proportional to the coefficient on the bound vortex sheet strength's singularity [Eldredge, 2019], so in this discrete setting, in which we have extracted the singular part of f in the form of f_0, we expect the suction parameter to be related to the value of $\tilde{\mathsf{f}}$ at the edge. In fact, by simple comparison, it can be shown that

$$\mathsf{e}_k^T \tilde{\mathsf{f}} = -\frac{2\pi c}{\Gamma_0} \sigma_k \tag{3.69}$$

for a flat plate of length c, where σ_k is the suction parameter at the edge corresponding to point k.

Let $\sigma_k^{\min}$ and $\sigma_k^{\max}$ denote the minimum and maximum tolerable values of σ_k at edge k. We then seek to confine the suction parameter to the range $\sigma_k^{\min} \leq \sigma_k \leq \sigma_k^{\max}$. This generalized edge constraint is placed on the suction parameter of the intermediate sheet $\tilde{\mathsf{f}}^*$. To avoid confusion, we will redefine the bounds based on this non-singular part of the vortex sheet rather than σ_k itself; for this, we define $\tilde{f}_k^{\min} = -2\pi c \sigma_k^{\max}/\Gamma_0$ and $\tilde{f}_k^{\max} = -2\pi c \sigma_k^{\min}/\Gamma_0$ if Γ_0 is positive or $\tilde{f}_k^{\max} = -2\pi c \sigma_k^{\max}/\Gamma_0$ and $\tilde{f}_k^{\min} = -2\pi c \sigma_k^{\min}/\Gamma_0$ if Γ_0 is negative. Thus, we inspect whether the value $\mathsf{e}_k^T \tilde{\mathsf{f}}^*$ lies in the range

$$\tilde{f}_k^{\min} \leq \mathsf{e}_k^T \tilde{\mathsf{f}}^* \leq \tilde{f}_k^{\max}. \tag{3.70}$$

If $\mathsf{e}_k^T \tilde{\mathsf{f}}^*$ lies within this range, then no new vortex is created near the edge (or equivalently, a new vortex of zero strength is created); if $\mathsf{e}_k^T \tilde{\mathsf{f}}^* > \tilde{f}_k^{\max}$, then we create a new vortex so that

$\mathsf{e}_k^T \tilde{\mathsf{f}} = \tilde{f}_k^{\max}$; and if $\mathsf{e}_k^T \tilde{\mathsf{f}}^* < \tilde{f}_k^{\min}$, then we do the same, but now so that $\mathsf{e}_k^T \tilde{\mathsf{f}} = \tilde{f}_k^{\min}$. Note that the Kutta condition simply corresponds to setting $\tilde{f}_k^{\min} = \tilde{f}_k^{\max} = 0$.

We can easily accommodate these constraints into our solution approach for enforcing the Kutta condition from the previous section: in the case of two edges, by modifying the right-hand side vector of (3.66) (if the edge suction lies outside of its bounds) or setting the vortex strength corresponding to that edge to zero. For example, suppose that $\mathsf{e}_{k_1}^T \tilde{\mathsf{f}}^* < \tilde{f}_{k_1}^{\min}$ and $\mathsf{e}_{k_2}^T \tilde{\mathsf{f}}^* > \tilde{f}_{k_2}^{\max}$; then we solve the system

$$\begin{bmatrix} 1 + \hat{\mathsf{f}}_0^T \mathsf{P}_{k_1}^K \tilde{\mathsf{f}}_1 & 1 + \hat{\mathsf{f}}_0^T \mathsf{P}_{k_1}^K \tilde{\mathsf{f}}_2 \\ 1 + \hat{\mathsf{f}}_0^T \mathsf{P}_{k_2}^K \tilde{\mathsf{f}}_1 & 1 + \hat{\mathsf{f}}_0^T \mathsf{P}_{k_2}^K \tilde{\mathsf{f}}_2 \end{bmatrix} \begin{pmatrix} \Gamma_{v,1} \\ \Gamma_{v,2} \end{pmatrix} = - \begin{pmatrix} \Gamma_{\mathsf{w}} + \hat{\mathsf{f}}_0^T \mathsf{P}_{k_1}^K \tilde{\mathsf{f}}^* + \Gamma_0 \tilde{f}_{k_1}^{\min} \\ \Gamma_{\mathsf{w}} + \hat{\mathsf{f}}_0^T \mathsf{P}_{k_2}^K \tilde{\mathsf{f}}^* + \Gamma_0 \tilde{f}_{k_2}^{\max} \end{pmatrix}. \tag{3.71}$$

But if, say, $\tilde{f}_{k_2}^{\min} \leq \mathsf{e}_{k_2}^T \tilde{\mathsf{f}}^* \leq \tilde{f}_{k_2}^{\max}$, then we set $\Gamma_{v,2} = 0$ and this system reduces to

$$\Gamma_{v,1} = -\frac{\Gamma_{\mathsf{w}} + \hat{\mathsf{f}}_0^T \mathsf{P}_{k_1}^K \tilde{\mathsf{f}}^* + \Gamma_0 \tilde{f}_{k_1}^{\min}}{1 + \hat{\mathsf{f}}_0^T \mathsf{P}_{k_1}^K \tilde{\mathsf{f}}_1}. \tag{3.72}$$

The effect of applying these generalized edge conditions to the leading edge of a flat plate is shown in figure 3.11 for the first instants after impulsively starting a uniform flow. The positions of the point vortices emanating from the leading edge in the figure indicate that as $\sigma_k^{\max}$ increases, the stream of point vortices is swept back from the edge. At the trailing edge, the Kutta condition is enforced in each case and the positions of the point vortices overlap, as they are not yet influenced by the different situations at the leading edge in these first instants.

3.6 Pressure, force, and the added mass

In this section, we present the means for calculating pressure and force (and moment) in the grid-based treatment.

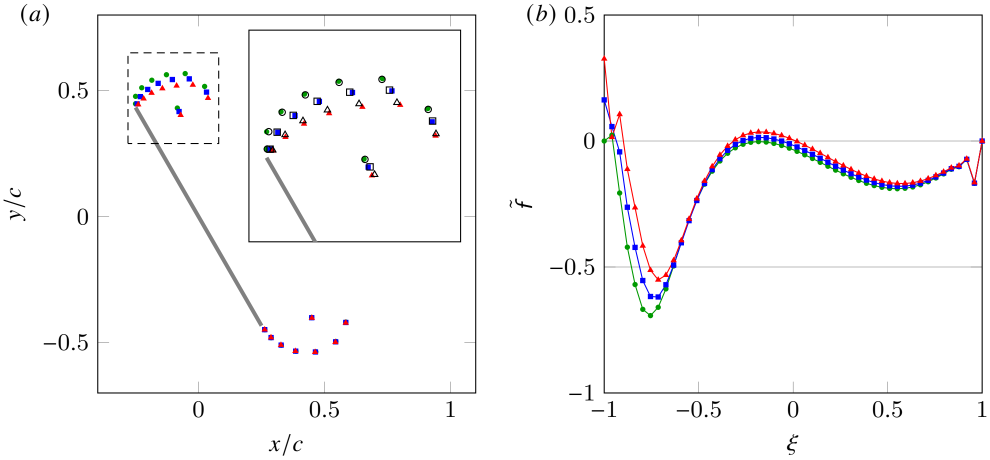


Figure 3.11: Effect of increasing $\sigma_{\text{LE}}^{\text{max}}/U_\infty$ from 0 (●) to 0.05 (■) and 0.1 (▲) on (*a*) the positions of shedded point vortices and (*b*) the non-singular part of the associated discrete vortex sheet strength for a flat plate of length c at 60°, 0.15 convective times after impulsively starting a uniform flow U_∞. The inset enlarges the boxed area (— — —) and overlays the positions of the vortices (○, □, and △) obtained using the Biot-Savart method of Darakananda and Eldredge [2019]. At the trailing edge, the Kutta condition is enforced. The simulation is performed with $\Delta x/c = 0.01$, $\Delta S/\Delta x = 2$, and $\Delta t U_\infty/c = 0.025$.

3.6.1 Pressure

Here we devise a means of computing the pressure, both throughout the flow field, $\mathsf{p} \in \mathcal{C}$, as well as on the surface of a body. As we will show below, our immersed boundary treatment naturally gives rise to the jump in pressure across this surface, $\Delta p = p^+ - p^- \in \mathcal{S}^{N_s}$ (with + in the direction of the surface normal, $n \in \mathcal{V}^{N_s}$). Thus, to distinguish the pressures on either side of the surface from one another, we use the fact that the interpolation of p onto the surface produces the average of surface values, ${}^{\mathcal{S}}\mathsf{E}_{\mathcal{C}}\mathsf{p} = (p^+ + p^-)/2 \in \mathcal{S}$, where ${}^{\mathcal{S}}\mathsf{E}_{\mathcal{C}}$ interpolates data from cell centers to the surface. It is thus easy to see that the pressure on either side is

$$p^{\pm} = {}^{\mathcal{S}}\mathsf{E}_{\mathcal{C}}\mathsf{p} \pm \tfrac{1}{2}\Delta p. \tag{3.73}$$

Thus, we seek Δp and p. It should not be a surprise that our starting point for these quantities is the Euler equations. However, our approach exploits the fact that the methodology we have presented thus far already solves the Euler equations in the fluid—in vorticity form, via transport of vortex elements—and provides us with the instantaneous velocity field v and strength of the bound vortex sheet f on any bodies. This approach, which satisfies the incompressibility constraint by expressing velocity in the null space of the divergence operator (i.e., as curl of a streamfunction), obviates the need for computing pressure, the Lagrange multiplier for this constraint. However, now that we seek pressure, we use the Euler equations in their velocity form to provide it.

It is important to note that the immersed boundary treatment enriches the Euler equations' standard form with surface terms [Eldredge, 2022]. Written in their spatially-discrete form, these immersed boundary Euler equations are

$$\begin{aligned} \rho\frac{\mathrm{d}\mathsf{v}}{\mathrm{d}t} + \rho\left(\mathsf{w} + {}^{\mathcal{N}}\mathsf{R}_{\mathcal{S}}f\right) \times \mathsf{v} = -\mathsf{G}\left(\mathsf{p} + \tfrac{1}{2}\rho|\mathsf{v}|^2\right) + \\ {}^{\mathcal{F}}\mathsf{R}_{\mathcal{V}}\left[n \circ \Delta p + \rho\Delta\mathsf{v}\left({}^{\mathcal{V}}\mathsf{E}_{\mathcal{F}}\mathsf{v} - v\right) \cdot n\right], \end{aligned} \tag{3.74}$$

where ρ is the fluid density and ${}^{\mathcal{V}}\mathsf{E}_{\mathcal{F}}$ interpolates grid data from $\mathcal{F}$ to the space of vector-valued surface data $\mathcal{V}^{N_s}$. Note the appearance of a few terms that are not typically seen in the Euler equations. First, we note the final bracketed pair of surface terms on the right-hand side, containing the jumps in pressure and velocity ($\Delta\mathsf{v}$) across the surface, as well as the difference between the normal components of the fluid velocity (interpolated onto the surface) and the surface velocity itself, $v \in \mathcal{V}^{N_s}$. (This difference is zero by virtue of the no-flow-through condition, but we keep it here since it combines with other terms in later manipulations.) These surface terms are immersed into the grid by the operator ${}^{\mathcal{F}}\mathsf{R}_{\mathcal{V}}$. In fact, if we had chosen to solve the Euler equations in velocity—instead of by streamfunction-vorticity—form, then we would have used the pressure jump in this term as a Lagrange multiplier for enforcing the no-flow-through condition on v. The second new term is the bound vortex sheet strength, appearing alongside the fluid vorticity on the left-hand side. This term emerges because the curl of the velocity field in the immersed boundary method generates both of these: $\mathsf{C}^T\mathsf{v} = \mathsf{w} + {}^{\mathcal{N}}\mathsf{R}_{\mathcal{S}} f$, where $\mathsf{C}^T : \mathcal{F} \mapsto \mathcal{N}$. (Our notation for this operator is consistent with that of previous authors, such as Colonius and Taira [2008].)

These quantities, Δp and p, can be solved for simultaneously from the Schur complement system that arises from solving the Euler equations (3.74) and the associated constraints of divergence-free velocity and no-flow-through condition. Indeed, the approach we outline here is the natural outcome of that system. However, rather than present a detailed derivation, we present the equations with an intuitive explanation.

First, we develop an equation for Δp by taking the discrete curl C^T of (3.74) to eliminate the gradient term—since $\mathsf{C}^T\mathsf{G}$ is identically zero—and obtain a vorticity form of the immersed-boundary Euler equations. With some manipulation to account for the motion of the surface (embodied in the time variation of ${}^{\mathcal{N}}\mathsf{R}_{\mathcal{S}}$), these equations can be written as

$$\rho\left(\frac{\mathrm{d}\mathsf{w}}{\mathrm{d}t} - \mathsf{C}^T(\mathsf{v}\times\mathsf{w})\right) + \rho\,{}^{\mathcal{N}}\mathsf{R}_{\mathcal{S}}\frac{\mathrm{d}f}{\mathrm{d}t} = \mathsf{C}^T\,{}^{\mathcal{F}}\mathsf{R}_{\mathcal{V}}\, n \circ \left[\Delta p + \rho\left({}^{\mathcal{V}}\mathsf{E}_{\mathcal{F}}\mathsf{v} - v\right)\cdot(f\times n)\right], \qquad (3.75)$$

where $\mathrm{d}f/\mathrm{d}t$ represents the time derivative of each element of f while following a point moving with velocity v. We assume that each immersed point moves with this local surface velocity, so this is simply the time derivative of the vector f. We have written the equation in this intermediate form on purpose in order to make a few key points. First, it is important to note that we have already satisfied the vorticity equation in the fluid—the first terms in parentheses on the left-hand side—by advecting the point vortices. Thus, we can set these terms to zero, leaving only those terms associated with the surface. Aside from $\Delta \mathsf{p}$, these remaining terms involve only known quantities, and we could solve them in the current form for $\Delta \mathsf{p}$ using similar techniques to the ones we will describe below. However, we will first write the equation in a more familiar form, and define some helpful operators and quantities to enable this.

In equation (3.75) we see a composite of the curl of the regularization operator; let us write this more compactly as $\mathsf{C}_{\mathcal{S}}^{T} : \mathcal{S}^{N_s} \mapsto \mathcal{N}$,

$$\mathsf{C}_{\mathcal{S}}^{T}\sigma = \mathsf{C}^{T\, \mathcal{F}}\mathsf{R}_{\mathcal{V}} \left(\mathsf{n} \circ \sigma \right), \tag{3.76}$$

for some surface scalar data, $\sigma \in \mathcal{S}^{N_s}$. We refer to this as a *surface curl* operator. Its transpose, $\mathsf{C}_{\mathcal{S}} : \mathcal{N} \mapsto \mathcal{S}^{N_s}$, also arises in what follows, and can also be described as a surface curl operator. It is defined as

$$\mathsf{C}_{\mathcal{S}}\mathsf{s} = \mathsf{n} \cdot {}^{\mathcal{V}}\mathsf{E}_{\mathcal{F}}\mathsf{C}\mathsf{s}, \tag{3.77}$$

for $\mathsf{s} \in \mathcal{N}$. The operator $\mathsf{C}_{\mathcal{S}}$ obtains the normal component of velocity on an immersed surface for a given streamfunction s. Before we explain the role of its transpose, it is useful to remember that any potential flow generated by (or about) an impenetrable surface can be equivalently described by either a distribution of vortices (a vortex sheet, with strength f) or a distribution of dipoles (a double layer) on the surface. In the latter case, the strength of the double layer is given by the negative of the jump in scalar potential across the surface, $-\Delta\phi \in \mathcal{S}^{N_s}$. In fact, the two distributions' strengths can be related to each other, either by using Stokes' theorem or by the properties of the generalized functions that underpin the immersed boundary method [Eldredge, 2022], leading to

$${}^{\mathcal{N}}\mathsf{R}_{\mathcal{S}}\mathsf{f} = -\mathsf{C}_{\mathcal{S}}^{T}\Delta\phi. \tag{3.78}$$

Thus, $\mathsf{C}_{\mathcal{S}}^{T}$ produces the equivalent bound vorticity distribution (immersed into the grid) associated with a given jump in scalar potential on the surface. To calculate $\Delta\phi$ in terms of f, we apply $\mathsf{C}_{\mathcal{S}}\mathsf{L}^{-1}$ to both sides of (3.78) to equate the normal velocity induced on the surface by each distribution. The composite operator $-\mathsf{C}_{\mathcal{S}}\mathsf{L}^{-1}\mathsf{C}_{\mathcal{S}}^{T}$ is positive semi-definite, and its null space can be shown to consist only of uniform values on the surface (i.e., the null space has an equivalent bound vorticity equal to zero). Thus, the jump in scalar potential associated with a vortex sheet of strength f is

$$\Delta\phi = -\left(\mathsf{C}_{\mathcal{S}}\mathsf{L}^{-1}\mathsf{C}_{\mathcal{S}}^{T}\right)^{-1}\mathsf{C}_{\mathcal{S}}\mathsf{L}^{-1\,\mathcal{N}}\mathsf{R}_{\mathcal{S}}\mathsf{f}, \tag{3.79}$$

to which we can add any constant value without affecting the result. Now, armed with this insight, we can return to equation (3.75), and rewrite the surface terms on the left-hand side in terms of $\Delta\phi$:

$$\mathsf{C}_{\mathcal{S}}^{T}\left[\Delta\mathsf{p} + \rho\left({}^{\mathcal{V}}\mathsf{E}_{\mathcal{F}}\mathsf{v} - \mathsf{v}\right)\cdot(\mathsf{f}\times\mathsf{n}) + \rho\frac{\mathrm{d}\Delta\phi}{\mathrm{d}t}\right] = 0, \tag{3.80}$$

where $\mathrm{d}\Delta\phi/\mathrm{d}t$ denotes the time derivative of $\Delta\phi$ associated with a particular immersed point, assumed to be moving with velocity v. This equation implies that the expression in brackets must be equal to a uniform value, which we can take to be zero without loss of generality. We can immediately write an immersed boundary Bernoulli equation,

$$\Delta\mathsf{p} + \rho\left({}^{\mathcal{V}}\mathsf{E}_{\mathcal{F}}\mathsf{v} - \mathsf{v}\right)\cdot(\mathsf{f}\times\mathsf{n}) + \rho\frac{\mathrm{d}\Delta\phi}{\mathrm{d}t} = 0. \tag{3.81}$$

This equation is the discrete equivalent of a continuous version that appears in previous works, e.g. Jones [2003], Eldredge [2019]. At each time step, we use (3.80) to compute the instantaneous jump in scalar potential associated with the vortex sheet f, and then use (3.81) to find $\Delta\mathsf{p}$.

Now we can substitute $\Delta\mathsf{p}$ from equation (3.81) into the Euler equations (3.74) and operate on these equations with the discrete divergence operator, so that $\mathsf{L} = \mathsf{D}\mathsf{G}$ acts on p. We solve the resulting equation, obtaining

$$\mathsf{p} = -\frac{1}{2}\rho|\mathsf{v}|^2 - \rho\mathsf{L}^{-1}\left[\mathsf{D}(\mathsf{w}\times\mathsf{v}) + {}^{\mathcal{F}}\mathsf{R}_{\mathcal{V}}\left(\mathsf{n}\circ\frac{\mathrm{d}\Delta\phi}{\mathrm{d}t} - \mathsf{v}\times\mathsf{f}\right)\right]. \tag{3.82}$$

It should be noted that any uniform value can be arbitrarily added to this expression. Also, we note in passing that the final set of terms (with the inverse Laplacian acting on the term in brackets) is equivalent to the time derivative of the scalar potential field, rendering the overall equation equivalent to a Bernoulli equation on the grid. However, there is no particular advantage in writing the equation in that form. In the current form, the first term in brackets (the divergence of the Lamb vector) represents the direct force exerted on the fluid by the fluid vorticity. The remaining terms in brackets collectively constitute the effects of surface motion and of the surface's modification of the flow induced by fluid elements (e.g., vorticity, free stream).

We now demonstrate the pressure calculations with two examples that have an analytical solution for the pressure distribution on the surface. Figure 3.12 and 3.13 depict the pressure field and surface distribution for a vortex near a cylinder and a plate, respectively. The pressure inside the cylinder is close to the exact solution of a uniform value, except for some noise at the side near the vortex, which is visible on the interior surface distribution of the pressure. The exterior surface pressure distribution on the cylinder shows good agreement with the analytical solution. In the example of the flat plate, shown in figure 3.13, the bottom and top surface pressure distributions again show good agreement with the analytical solution except for at the edges, which is expected. Note that in these examples, we used a smaller value for $\Delta S/\Delta x$ than before, which was necessary to prevent the low pressure from the vortex from leaking through the surface. As a result, high-frequency components of the surface pressure are incorrectly amplified through the regularization and interpolation operators Goza et al. [2016], especially at the edges.

3.6.2 Impulse-based calculations of force and moment

The integral of the pressure distribution over the surface (plus any edge-suction parameters in the case of sharp edges) will be equal to the force on the surface. However, in this section we provide an alternative means of calculating the force and moment on the body through the negative rate of change of impulse in the fluid. The continuous expressions for linear and

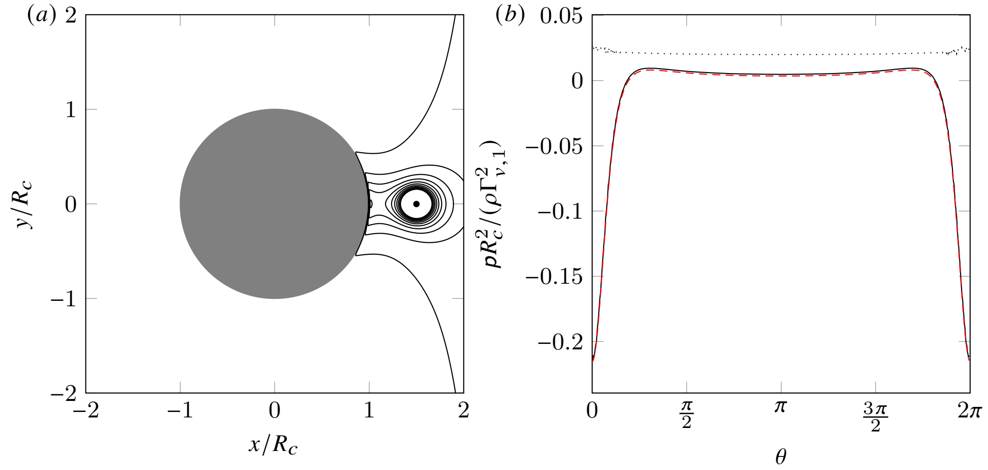


Figure 3.12: (a) Contours (——) of the discrete pressure for a point vortex (●) with strength $\Gamma_{v,1}$ at $(R_v, 0)$ near a cylinder consisting of N_s points with radius R_c and a bound circulation $-\Gamma_{v,1}$. (b) The scaled discrete pressure at the exterior (——) and interior (······) of the cylinder. Overlaid is the exact continuous solution (- - - -) for the exterior pressure. The simulation is performed with $R_v/R_c = 3/2$, $\Delta x/R_c = 0.05$, $\Delta S/\Delta x = 1.4$, and $\Delta t\Gamma_{v,1}/R_c^2 = 0.01$.

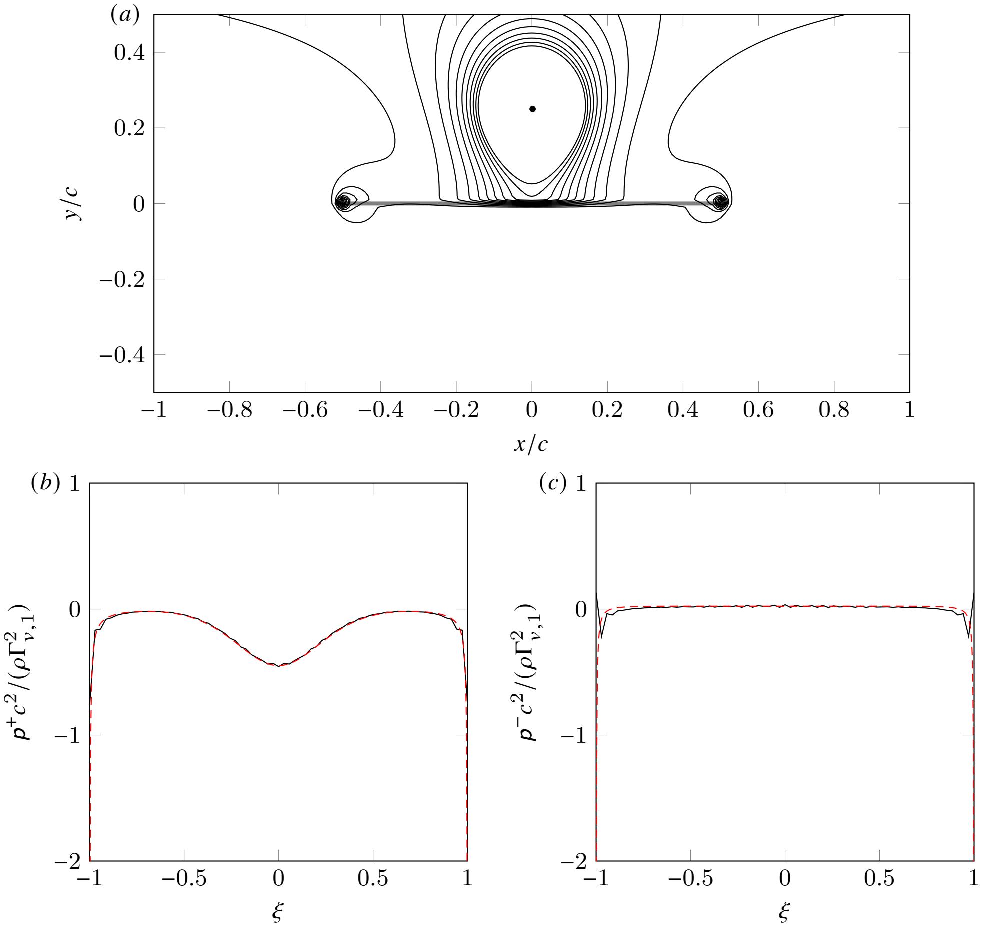


Figure 3.13: (a) Contours (——) of the discrete pressure for a point vortex (●) with strength $\Gamma_{v,1}$ at $(0, R_v)$ near a flat plate consisting of N_s points with chord length c and a bound circulation $-\Gamma_{v,1}$. The scaled discrete pressure (——) at (b) the top side and (c) the bottom side of the plate. Overlaid is the exact continuous solution (- - - -). The simulation is performed with $R_v/c = 0.25$, $\Delta x/c = 0.01$, $\Delta S/\Delta x = 1.4$, and $\Delta t\Gamma_{v,1}/c^2 = 0.005$.

angular impulse (about the origin) are, in two dimensions [Saffman, 1993, Eldredge, 2019],

$$\boldsymbol{P} = \int_{\boldsymbol{\Omega}} \boldsymbol{x} \times \boldsymbol{\omega} \, \mathrm{d}V + \int_{\mathcal{S}} \boldsymbol{x} \times (\boldsymbol{n} \times \boldsymbol{u}) \, \mathrm{d}S \tag{3.83}$$

$$\boldsymbol{\Pi}[\mathrm{O}] = \frac{1}{2} \int_{\boldsymbol{\Omega}} \boldsymbol{x} \times (\boldsymbol{x} \times \boldsymbol{\omega}) \, \mathrm{d}V + \frac{1}{2} \int_{\mathcal{S}} \boldsymbol{x} \times [\boldsymbol{x} \times (\boldsymbol{n} \times \boldsymbol{u})] \, \mathrm{d}S \tag{3.84}$$

If there is only a single body, then the force and moment (about the origin) exerted by the fluid on that body are given by

$$\boldsymbol{F} = -\rho \frac{\mathrm{d}\boldsymbol{P}}{\mathrm{d}t}, \qquad \boldsymbol{M}[\mathrm{O}] = -\rho \frac{\mathrm{d}\boldsymbol{\Pi}_{\mathrm{O}}}{\mathrm{d}t}, \tag{3.85}$$

where ρ is the fluid density. In the two-dimensional applications of this paper, the angular impulse and the moment have only a single component, e.g., $\boldsymbol{\Pi}_{\mathrm{O}} = \Pi_{\mathrm{O}} \boldsymbol{e}_z$, where $\boldsymbol{e}_z$ is the unit vector out of the plane.

It should be observed that, by definition, the bound vortex sheet strength γ is equal to the jump in tangential velocity between the fluid and the surface, $\boldsymbol{n} \times \boldsymbol{u} = \gamma \boldsymbol{e}_z + \boldsymbol{n} \times \boldsymbol{u}_b$, where $\boldsymbol{n}$ is the unit surface normal vector directed into the fluid, $\boldsymbol{u}$ is the fluid velocity, and $\boldsymbol{u}_b$ is the velocity of the surface. Thus, the surface integrals in (3.83) and (3.84) can be re-written in terms of the vortex sheet strength and the body motion.

We can easily develop discrete forms of the integrals (3.83) and (3.84) with the solutions and notation described in this paper. For the volume integrals, let us denote diagonal matrices containing the coordinates of the grid nodes by $\mathsf{D_x}$ and $\mathsf{D_y}$. Thus, the expressions in (3.83) and (3.84) can be written in discrete form as

$$P_x = \langle \mathsf{y}, \mathsf{w} \rangle_{\mathcal{N}} + \langle \mathsf{r}_y, \mathsf{f} + \mathsf{D}_{\mathsf{n}_x} \mathsf{v}_y - \mathsf{D}_{\mathsf{n}_y} \mathsf{v}_x \rangle_{\mathcal{S}^{N_s}}, \tag{3.86}$$

$$P_y = -\langle \mathsf{x}, \mathsf{w} \rangle_{\mathcal{N}} - \langle \mathsf{r}_x, \mathsf{f} + \mathsf{D}_{\mathsf{n}_x} \mathsf{v}_y - \mathsf{D}_{\mathsf{n}_y} \mathsf{v}_x \rangle_{\mathcal{S}^{N_s}}, \tag{3.87}$$

and

$$\Pi_{\mathrm{O}} = -\frac{1}{2} \langle \mathsf{D_x x} + \mathsf{D_y y}, \mathsf{w} \rangle_{\mathcal{N}} - \frac{1}{2} \langle \mathsf{D}_{\mathsf{r}_x} \mathsf{r}_x + \mathsf{D}_{\mathsf{r}_y} \mathsf{r}_y, \mathsf{f} + \mathsf{D}_{\mathsf{n}_x} \mathsf{v}_y - \mathsf{D}_{\mathsf{n}_y} \mathsf{v}_x \rangle_{\mathcal{S}^{N_s}}. \tag{3.88}$$

The overall force and moment exerted on the body are obtained from calculating these impulses and computing their rates of change in (3.85). Part of this force and moment is attributable to the dynamics of vorticity in the fluid. The remaining part is due to surface motion relative to the fluid, and we will discuss this in the next section.

To illustrate the accuracy of the impulse-based calculation of force, we apply the method to two examples. In the first example, we simulate the trajectories of two point vortices of opposite strength, in which case each vortex is convected past a cylinder due to the presence of the other vortex. The time stepping in this example is carried out using a fourth-order Runge-Kutta scheme. Figure 3.14 shows the trajectories and the x component of the impulse together with their exact solutions, which show good agreement with the exact solution. In the second example, we compare our simulation of the first instants of the unsteady, fully separated flow around a flat plate after impulsively starting a uniform flow with the Biot-Savart method from Darakananda and Eldredge [2019], using the same positioning rules to insert point vortices and the same time step. The vortex positions and the corresponding impulse and lift are compared in figure 3.15 and show good agreement as well.

3.6.3 Added mass

The added mass tensor provides a measure of the inertial influence of the fluid on the body in response to changes in the body's translational or rotational motion. The coefficients of the added mass tensor of a body are obtained by computing the impulse components associated with a unit-valued component of motion. The motion's influence is both direct, via the surface velocity, and indirect, in the bound vortex sheet that develops on the surface.

For example, suppose that we consider translation at unit velocity in the x direction, for which the motion is described by $\mathsf{v}_x = \mathit{1}$, $\mathsf{v}_y = 0$, and $\mathsf{s}_b' = \mathsf{r}_y$, and the associated bound vortex sheet—obtained without the Kutta condition by solving the basic problem (3.21)—is $\mathsf{f} = \mathsf{S}^{-1}\mathsf{P}^{\Gamma}\mathsf{s}_b' = \mathsf{S}^{-1}\mathsf{P}^{\Gamma}\mathsf{r}_y$. The added mass coefficients corresponding to this motion are

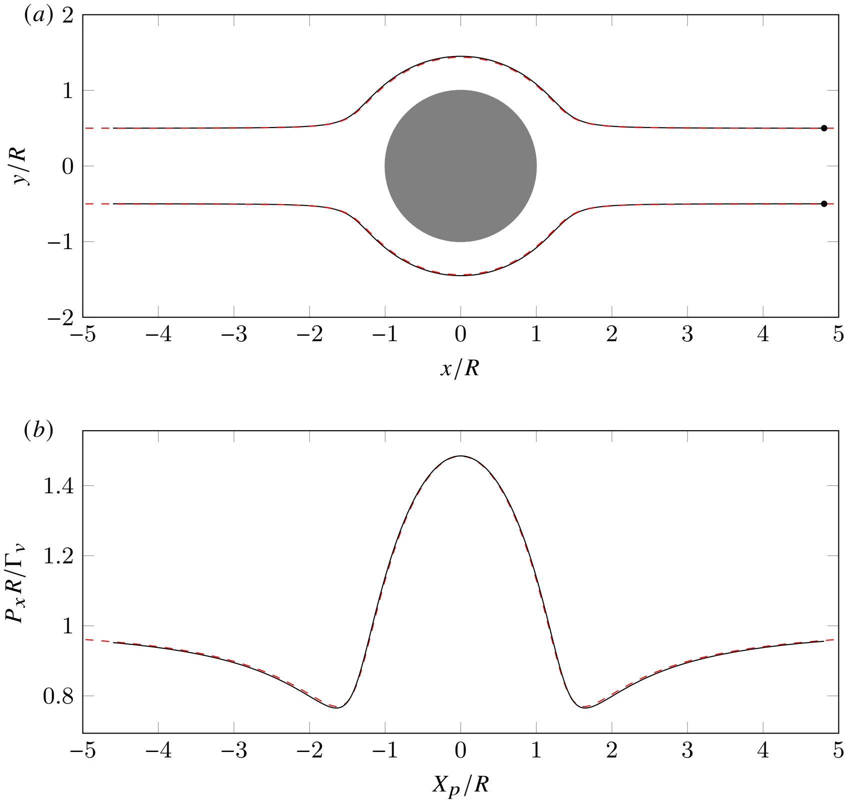


Figure 3.14: (*a*) Numerically simulated trajectories (——) of two point vortices (●) of opposite strengths $\Gamma_{v,1}$ and $\Gamma_{v,1} = -\Gamma_{v,[}2]$ being convected past a circular cylinder with radius R, and (*b*) the x component of the associated, numerically simulated impulse (——) in the fluid. Overlaid are the exact continuous trajectories and impulse (- - - -). The simulation is performed with $\Delta x/R = 0.04$, $\Delta S/\Delta x = 2$, and $\Delta t\Gamma_{v,1}/R^2 = 0.1$.

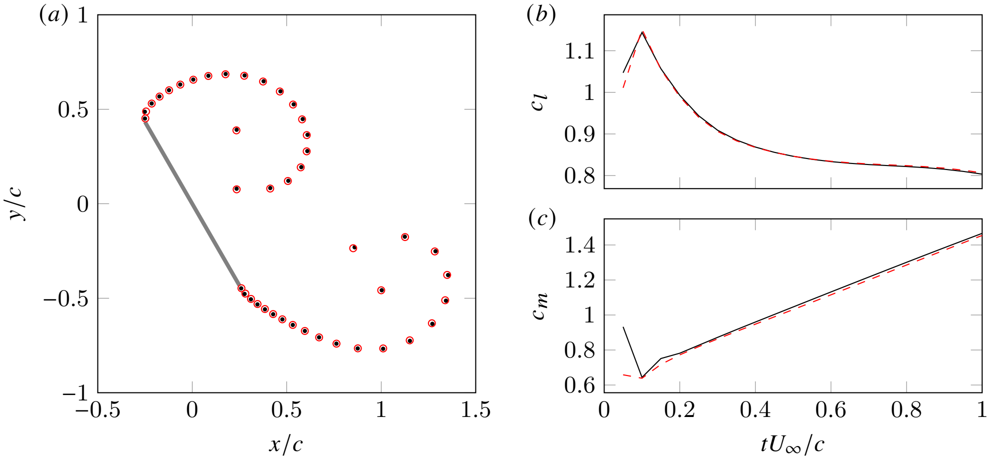


Figure 3.15: Comparison of the simulated vortex shedding behind a flat plate of length c at 60° in a uniform flow using the method in this paper (● and ——) and using the Biot-Savart method of Darakananda and Eldredge [2019] (○ and - - - -). (*a*) The positions of the shedded point vortices, (*b*) the lift coefficient, and (*c*) the moment coefficient, one convective time after impulsively starting the uniform flow. At both edges, the Kutta condition is enforced. The simulation is performed with $\Delta x/c = 0.01$, $\Delta S/\Delta x = 2$, and $\Delta t U_\infty/c = 0.05$.

derived by substituting these into the impulse formulas (3.86)–(3.88):

$$P_x^{(x)} = \langle \mathsf{r}_y, \mathsf{S}^{-1}\mathsf{P}^{\Gamma}\mathsf{r}_y - \mathsf{D}_{\mathsf{n}_y}\mathit{1}\rangle_{\mathcal{S}^{N_s}} \tag{3.89}$$

$$P_y^{(x)} = -\langle \mathsf{r}_x, \mathsf{S}^{-1}\mathsf{P}^{\Gamma}\mathsf{r}_y - \mathsf{D}_{\mathsf{n}_y}\mathit{1}\rangle_{\mathcal{S}^{N_s}}, \tag{3.90}$$

$$\Pi_{\mathrm{O}}^{(x)} = -\frac{1}{2}\langle \mathsf{D}_{\mathsf{r}_x}\mathsf{r}_x + \mathsf{D}_{\mathsf{r}_y}\mathsf{r}_y, \mathsf{S}^{-1}\mathsf{P}^{\Gamma}\mathsf{r}_y - \mathsf{D}_{\mathsf{n}_y}\mathit{1}\rangle_{\mathcal{S}^{N_s}}. \tag{3.91}$$

Thus, the components of the added mass coefficients tensor associated with translation in the x direction are

$$m_{xx}^F = \rho P_x^{(x)} \tag{3.92}$$

$$m_{xy}^F = \rho P_y^{(x)}, \tag{3.93}$$

$$m_x^M = \rho\left(\Pi_{\mathrm{O}}^{(x)} - \boldsymbol{X}_c \times \boldsymbol{P}^{(x)}\right), \tag{3.94}$$

where $\boldsymbol{X}_c$ is the centroid of the body, which can be calculated using (B.8), and the superscript F and M are used to denote the coefficient for the force and moment, respectively.

A similar approach can be used to obtain the added mass coefficients due to unit translation in the y direction, for which $\mathsf{v}_x = 0$, $\mathsf{v}_y = \mathit{1}$, and $\mathsf{s}_b' = -\mathsf{r}_x$. The coefficients due to unit rotation follow from taking $\mathsf{v}_x = -\mathsf{r}_y$, $\mathsf{v}_y = \mathsf{r}_x$, and $\mathsf{s}_b' = -\frac{1}{2}(\mathsf{D}_{\mathsf{r}_x}\mathsf{r}_x + \mathsf{D}_{\mathsf{r}_y}\mathsf{r}_y)$.

3.7 Multiple bodies

The previous sections provided the formulations for potential flow with the presence of a body. The extension of these expressions to multiple bodies is straightforward and consists of allocating partitions of f to the different bodies. The surface streamfunction has to be partitioned accordingly, with the body motion streamfunction s_b containing the values for the discrete surface points from all the bodies and $\mathsf{s}_0 = \sum_j \mathit{1}_j s_{0,j}$ allocating a uniform value $s_{0,j}$ to the jth body, where $\mathit{1}_j \in \mathcal{S}^{N_s}$ is a vector whose ith component is one if it belongs to the jth body and zero otherwise. The system (3.21) can then be solved for the streamfunction field without modification.

As in the single-body case, if we want to enforce an edge condition on the jth body, we treat its uniform streamfunction value $s_{0,j}$ as a Lagrange multiplier and add a constraint on f to the saddle point system. In the unsteady case, we add a circulation constraint for each body and consider each new point vortex to be released from a specified body. For example, let us consider two bodies, with each body having one sharp edge. We assume $\Gamma_{v,1}$ and $\Gamma_{v,2}$ are the strengths from the vortices that were released from the first and second body, respectively. We can compute these strengths by enforcing the Kutta condition for both bodies using the saddle point system

$$\begin{bmatrix} \mathsf{L} & {}^{\mathcal{N}}\tilde{\mathsf{R}}_{\mathcal{S}} & 0 & 0 & \mathsf{d}_1 & \mathsf{d}_2 \\ {}^{\mathcal{S}}\mathsf{E}_{\mathcal{N}} & 0 & \mathit{1}_1 & \mathit{1}_2 & 0 & 0 \\ 0 & \mathit{e}_{k_1}^T & 0 & 0 & 0 & 0 \\ 0 & \mathit{e}_{k_2}^T & 0 & 0 & 0 & 0 \\ 0 & \hat{\mathit{f}}_{0,1}^T & 0 & 0 & 1 & 0 \\ 0 & \hat{\mathit{f}}_{0,2}^T & 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} \mathsf{s} \\ \tilde{\mathit{f}} \\ s_{0,1} \\ s_{0,2} \\ \Gamma_{v,1} \\ \Gamma_{v,2} \end{pmatrix} = \begin{pmatrix} -\mathsf{w} \\ \mathsf{P}^{\Gamma}\mathit{s}_b' \\ 0 \\ 0 \\ -\Gamma_{\mathsf{w},1} \\ -\Gamma_{\mathsf{w},2} \end{pmatrix}, \tag{3.95}$$

where we defined $\Gamma_{\mathsf{w},j}$ as the circulation of the existing vorticity in the flow that has previously been released from the jth body and $\mathit{f}_{0,j} = \mathsf{D}_{\mathit{f}_0}\mathit{1}_j$. Again, these solutions are easily extensible to an arbitrary number of edges per body. For example, Figure 3.16 demonstrates the method for two flat plates in a uniform flow where the LE and TE are regularized for both plates by releasing four point vortices during each time step.

The formulas for pressure, force and added mass also generalize to systems with multiple bodies. For example, figure 3.17 demonstrates an example of a potential flow model with an array of nine circular cylinders and compares the ratio of the largest eigenvalue of added mass coefficient tensor and the largest self-added mass coefficient of the system with the results of Chen [1975].

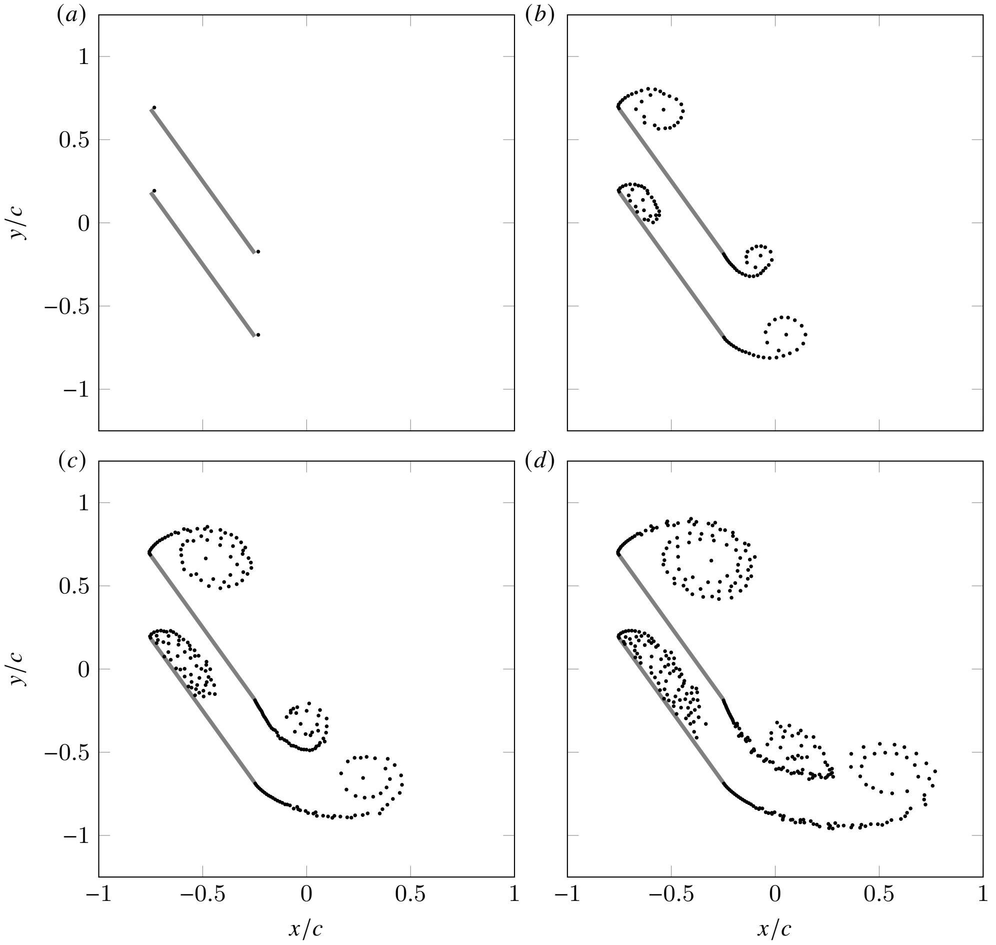


Figure 3.16: Evolution of the flow around two flat plates of length c at 60°, vertically separated by a distance of $c/2$, at tU_∞/c equal to (*a*) 0, (*b*) 0.3, (*c*) 0.6, and (*d*) 0.9 after impulsively starting a uniform flow. The simulation is performed with $\Delta x/c = 0.01$, $\Delta S/\Delta x = 2$, and $\Delta t U_\infty/c = 0.01$.

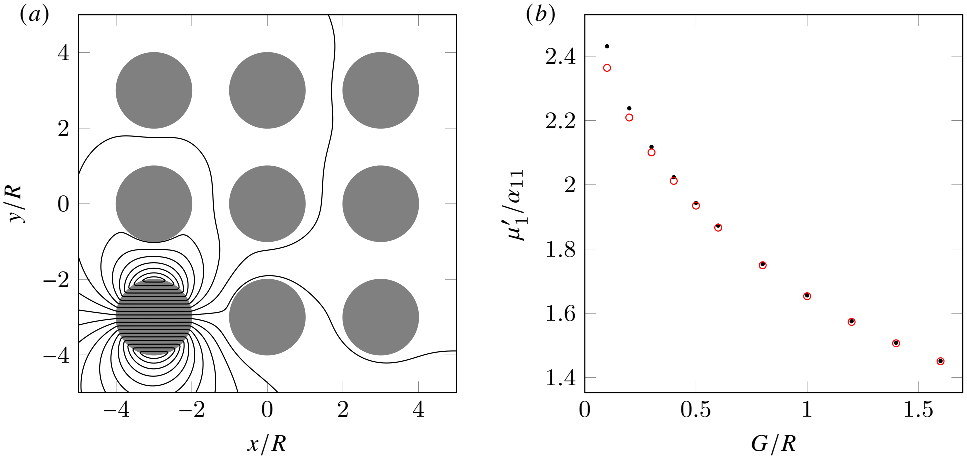


Figure 3.17: (a) Contours of the discrete streamfunction for an array of nine circular cylinders with radius R, spaced with a gap distance G between each cylinder, of which the bottom left cylinder translates horizontally. (b) Numerically simulated variation (●) of the ratio between the largest eigenvalue of the added mass coefficient tensor and the largest self-added mass coefficient with the gap-to-radius ratio G/R. Overlaid are the values (○) obtained by Chen [1975] from solving a system of truncated analytical expressions. The simulations are performed with $\Delta x/R = 0.05$ and $\Delta S/\Delta x = 2$.

CHAPTER 4

Modeling of wind tunnel walls

Experimental investigations of gust responses in wind tunnels provide information about the true flow physics at a wide range of Reynolds numbers that are representative of flight regimes, but they can suffer from drawbacks such as wind tunnel installation effects or limited flow-field information due to sensor shortcomings. Computational predictions of gust responses, on the other hand, provide data for the entire flow field and can mimic free flight but suffer from their own limitations. These include model uncertainties and constraints on spatial and temporal resolution, which result in limited capabilities to accurately model turbulent or separated flow at flight Reynolds numbers Malik and Bushnell [2012].

Recent computational efforts that are based on wind tunnel gust experiments have mostly been directed at validating numerical models for gust response predictions Badrya et al. [2019], Moushegian and Smith [2019], Grubb et al. [2020], performing a computational parameter exploration Badrya et al. [2021], or investigating the gust-interaction physics solely from the computational flow field Moriche et al. [2021]. In these studies, the similarity between the configurations of experiments and computations is usually compromised in order to make the numerical computation tractable, for example by ignoring the wind tunnel walls, using turbulence modeling instead of direct numerical simulations, or incomplete modeling of the gust. However, the effect of some of these simplifications on the flow features can be unclear.

This section investigates wind tunnel effects on a simulated, discrete gust response. To this end, we develop a computational model of a wind tunnel and compare its prediction of the gust lift response with the responses predicted by a free-space numerical model and classical unsteady aerodynamics. The numerical examples are two-dimensional at a low Reynolds number, but their purpose is to describe wind tunnel effects that generalize to higher Reynolds numbers. Furthermore, as a virtual counterpart (or digital twin) of an actual wind tunnel, the wind tunnel model can be combined with the experimental data using data assimilation to increase the accuracy of the gust prediction or estimate uncertain parameters.

The computational wind tunnel model is described for a wind tunnel design that is inspired by the low-speed Andrew Fejer Unsteady Wind Tunnel (AFUWT) at the Illinois Institute of Technology He et al. [2022], which generates irrotational transverse gusts by creating suction at the top wall of the wind tunnel. However, this is a computational framework that applies to any wind tunnel design that can generate irrotational gusts. We provide three two-dimensional examples of an airfoil's gust response simulated using the wind tunnel model. In the first example, we investigate the responses of a flat plate and NACA0009 airfoil to a Gaussian suction gust. In the second example, we model the response to the same gust but with higher strength and for a flat plate at an angle of attack to the inlet flow to investigate the non-linear effects of the gust. In the last example, we use synthetic velocity measurements that represent an actual opening and closing of the suction duct.

4.1 Approximation of the wind tunnel flow field

The velocity field $\boldsymbol{u}$ in a wind tunnel is governed by the Navier-Stokes equations. The true boundary conditions at every solid surface are no slip and no flow through, and the conditions at the inlets and outlets depend on the entire wind tunnel system, including the flow resistance of each wind tunnel component and the fan operating conditions Rennie et al. [2017]. Furthermore, most wind tunnels rely on a flow straightener (such as a honeycomb screen) to reduce the turbulence intensity at the inlet of the test section, but the turbulence can usually not be removed entirely.

Ideally, our computational model of the wind tunnel flow preserves these same flow qualities and boundary conditions. However, in this work, we limit ourselves to the computation of low Reynolds number flows and model the wind tunnel walls using inviscid boundary conditions, which account for the leading-order effect of the wind tunnel walls on the flow around the airfoil. This results in the following computational modeling framework for approximating wind tunnel flows that, through simplifications, can use highly-efficient computational methods from external flow solvers and lends itself well to the modeling of flows of higher Reynolds numbers.

Consider the two-dimensional geometry in Figure 4.1. We are interested in obtaining the aerodynamic forces on the stationary test subject $\partial\boldsymbol{\Omega}_{\mathrm{b}}$ and a detailed description of the flow near it. The overall velocity field $\boldsymbol{u}$ satisfies the incompressible Navier-Stokes equations

$$\begin{gathered}\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{\omega} \times \boldsymbol{u} = -\nabla\left(p + \frac{1}{2}|\boldsymbol{u}|^2\right) + \frac{1}{\mathrm{Re}}\nabla^2\boldsymbol{u}, \\ \nabla \cdot \boldsymbol{u} = 0, \end{gathered} \tag{4.1}$$

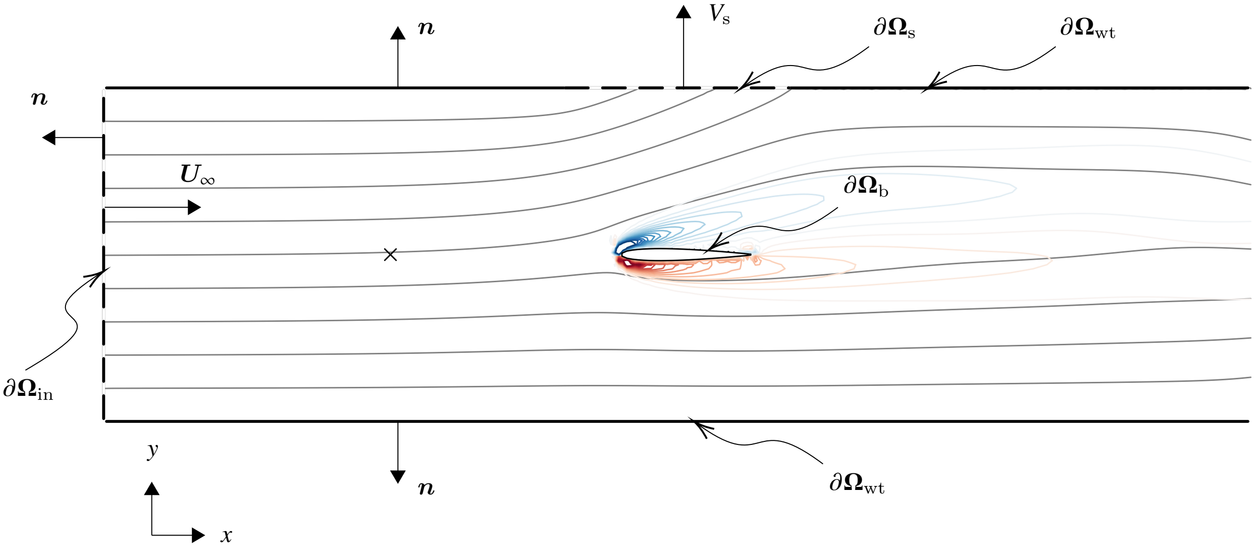
with p the pressure and Re the Reynolds number. We specify the viscous boundary condition on the surface of the test subject

$$\boldsymbol{u} = \boldsymbol{0} \text{ on } \partial\boldsymbol{\Omega}_{\mathrm{b}}. \tag{4.2}$$

We want to model the influence of the wind tunnel walls on the flow around the test subject, but we want to avoid spending computational resources on accurately modeling the viscous effects at the walls. Because the boundary layers are thin compared to the test section height and width at the high-Reynolds flows we are interested in, we can approximate them as infinitesimally thin and treat the walls as inviscid boundaries:

$$\boldsymbol{n} \cdot \boldsymbol{u} = 0 \text{ on } \partial\boldsymbol{\Omega}_{\mathrm{wt}}. \tag{4.3}$$

Similar normal-component kinematic boundary conditions can be formulated to specify the



$\boldsymbol{n}$
V_s
$\partial\boldsymbol{\Omega}_\mathrm{s}$
$\partial\boldsymbol{\Omega}_\mathrm{wt}$
$\boldsymbol{n}$
$\boldsymbol{U}_\infty$
$\partial\boldsymbol{\Omega}_\mathrm{b}$
$\partial\boldsymbol{\Omega}_\mathrm{in}$
y
$\boldsymbol{n}$
$\partial\boldsymbol{\Omega}_\mathrm{wt}$
x

Figure 4.1: Diagram of the wind tunnel with suction showing the streamlines (grey) and contours of vorticity (colored). The position of a crosswire probe is indicated with the x marker.

suction at $\partial\boldsymbol{\Omega}_\mathrm{s}$ (which we will discuss later) and prescribe a uniform flow at $\partial\boldsymbol{\Omega}_\mathrm{in}$,

$$\boldsymbol{n}\cdot\boldsymbol{u} = -U_\infty \text{ on } \partial\boldsymbol{\Omega}_\mathrm{in}. \tag{4.4}$$

Furthermore, we idealize the test section as a semi-infinite domain of constant cross section, and far downstream we prescribe the velocity to return to uniform flow

$$\boldsymbol{n}\cdot\boldsymbol{u} \to U, \text{ as } x\to\infty, \tag{4.5}$$

where U must be set so that the net volume flow rate due to inflow, suction, and outflow is zero.

If we further assume that the inlet and initial flow are irrotational, the vorticity will be confined to a region near the test subject and in its wake. The flow $\boldsymbol{u}$ that satisfies the conditions (4.2) and (4.3), as well as any conditions similar to (4.3), can then be constructed according to the Helmholtz decomposition from the sum of an external viscous flow $\boldsymbol{u}_{\boldsymbol{\omega}}$ with non-zero vorticity around the test subject, a freestream $\boldsymbol{U}_\infty = (U_\infty, 0)$, and an additional potential flow $\boldsymbol{u}_\phi = \nabla\phi$:

$$\boldsymbol{u} = \boldsymbol{u}_{\boldsymbol{\omega}} + \boldsymbol{u}_\phi + \boldsymbol{U}_\infty. \tag{4.6}$$

The purpose of the potential flow $\boldsymbol{u}_\phi$ is to introduce the influence of the wind tunnel into the problem: to correct the flow-through at the wind tunnel walls, prescribe the mass flow rate at the test section inlet, and to create a suction flow along a portion of the wind tunnel wall for gust generation. We will now focus on formulating the individual viscous and potential flow problems.

4.1.1 Viscous flow problem

The viscous part of our modeling framework is defined to be an unbounded external flow. If we replace the velocity in (4.1) by its Helmholtz decomposition, the viscous flow problem in our incompressible model becomes the problem of finding the velocity $\boldsymbol{u}_{\boldsymbol{\omega}}$ that satisfies

$$\begin{gathered} \frac{\partial \boldsymbol{u}_{\boldsymbol{\omega}}}{\partial t} + \boldsymbol{\omega} \times \boldsymbol{u} = -\nabla \tilde{P} + \frac{1}{\mathrm{Re}} \nabla^2 \boldsymbol{u}_{\boldsymbol{\omega}}, \\ \nabla \cdot \boldsymbol{u}_{\boldsymbol{\omega}} = 0, \end{gathered} \tag{4.7}$$

subject to

$$\begin{gathered} \boldsymbol{u}_{\boldsymbol{\omega}} = -\boldsymbol{U}_\infty - \boldsymbol{u}_\phi \text{ on } \partial\boldsymbol{\Omega}_\mathrm{b}, \\ \boldsymbol{u}_{\boldsymbol{\omega}}(\boldsymbol{x}, t) \to \boldsymbol{0},\ \tilde{P}(\boldsymbol{x}, t) \to p_\infty \text{ as } |\boldsymbol{x}| \to \infty, \end{gathered} \tag{4.8}$$

with $\tilde{P} = p + \frac{1}{2}|\boldsymbol{u}|^2 + \partial\phi/\partial t$.

We choose to solve this problem on a nominally infinite Cartesian grid using a fast Fourier transform implementation of the lattice Green's function (LGF) Liska and Colonius [2014b], combined with an immersed boundary Liska and Colonius [2016] or immersed layers Eldredge [2022] method to immerse the body in the grid. Alternatively, one can solve the problem on a truncated domain using artificial boundary conditions Colonius and Taira [2008]. However, one must be able to evaluate $\boldsymbol{u}_{\boldsymbol{\omega}}$ on the wind tunnel inlet and walls, as we will discuss in the next section. The benefit of using the LGF in the wind tunnel model is that the computational domain for the viscous flow problem only has to be large enough to encompass the vorticity Yu [2021], while the solved flow field can easily be evaluated outside of the solver's computational domain, for example, to obtain $\boldsymbol{u}_{\boldsymbol{\omega}}$ on the wind tunnel walls. Figure 4.2 shows the contours of the vorticity field $\boldsymbol{\omega}$ and the streamlines of the $\boldsymbol{u}_{\boldsymbol{\omega}}$ field, which generally pass through the wind tunnel walls, indicating the need to correct the velocity at those walls.

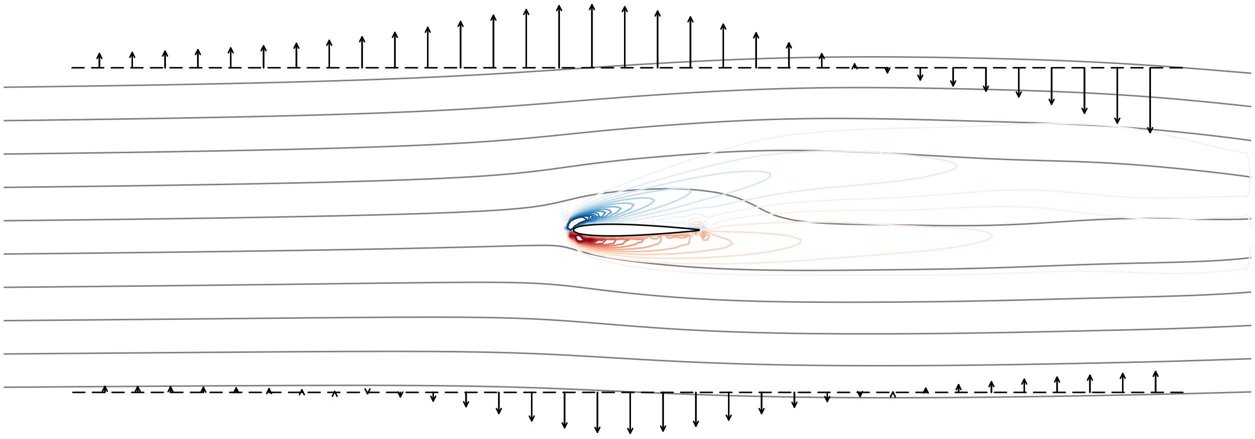


Figure 4.2: Streamlines of $\boldsymbol{u}_{\boldsymbol{\omega}}$ (grey), contours of vorticity (colored), and the location of the wind tunnel walls (----). The arrows indicate the relative magnitude and direction of the normal component of $\boldsymbol{u}_{\boldsymbol{\omega}}$ at the locations of those walls.

4.1.2 Potential flow problem

The role of the potential flow in (4.6) is to correct $\boldsymbol{u}_{\boldsymbol{\omega}}$ at the walls and inlet and introduce the suction into the flow. Due to the linear nature of potential flow, we can view $\boldsymbol{u}_{\phi}$ as a superposition of both effects:

$$\boldsymbol{u}_{\phi} = \boldsymbol{u}_{\phi,\mathrm{c}} + \boldsymbol{u}_{\phi,\mathrm{s}}, \tag{4.9}$$

where $\boldsymbol{u}_{\phi,\mathrm{c}}$ is the correction flow at the walls and inlet and $\boldsymbol{u}_{\phi,\mathrm{s}}$ is the suction flow, and each flow is a potential flow:

$$\begin{aligned} \nabla^2\phi_{\mathrm{c}} &= 0, & \qquad \nabla^2\phi_{\mathrm{s}} &= 0, \\ \boldsymbol{u}_{\phi,\mathrm{c}} &= \nabla\phi_{\mathrm{c}}, & \qquad \boldsymbol{u}_{\phi,\mathrm{s}} &= \nabla\phi_{\mathrm{s}}. \end{aligned}$$

We don't constrain the flow on $\partial\boldsymbol{\Omega}_{\mathrm{b}}$ in either potential flow problem, but each flow has different constraints on the test-section boundaries.

To ensure that $\boldsymbol{u}$ satisfies the no-flow-through condition at the wind tunnel walls (4.3), we specify the Neumann condition

$$\boldsymbol{n}\cdot\boldsymbol{u}_{\phi,\mathrm{c}} = -\boldsymbol{n}\cdot\boldsymbol{u}_{\boldsymbol{\omega}} \text{ on } \partial\boldsymbol{\Omega}_{\mathrm{wt}} \cup \partial\boldsymbol{\Omega}_{\mathrm{s}} \tag{4.10}$$

as a boundary condition for the potential flow problem for $\boldsymbol{u}_{\phi,\mathrm{c}}$.

So far, our only means of controlling the test section mass flow rate is choosing the freestream velocity $\boldsymbol{U}_\infty$. However, the normal velocity at the inlet and outlet is potentially modified by $\boldsymbol{u}_{\boldsymbol{\omega}}$. The potential flow correction provides a way to fix the flow rate to a prescribed value because we can set the boundary condition for this correction flow so that the overall flow satisfies the normal-velocity requirement Eq. (4.4). Thus, the correction potential is constrained to satisfy the Neumann condition at the inlet

$$\boldsymbol{n}\cdot\boldsymbol{u}_{\phi,\mathrm{c}} = -\boldsymbol{n}\cdot\boldsymbol{u}_{\boldsymbol{\omega}} - U_\infty \text{ on } \partial\boldsymbol{\Omega}_{\mathrm{in}}. \tag{4.11}$$

At the idealized outlet of the test section at $x \to \infty$, we have a similar Neumann condition,

$$\boldsymbol{n}\cdot\boldsymbol{u}_{\phi,\mathrm{c}} \to U_\infty + U_{\boldsymbol{\omega},\infty} \text{ as } x \to \infty, \tag{4.12}$$

where $U_{\boldsymbol{\omega},\infty}$ is equal to the total volume flow rate of $\boldsymbol{u}_{\boldsymbol{\omega}}$ through the other boundaries divided by the height of the outlet. This ensures that the total volume flow rate through all boundaries is zero.

This potential flow problem does not require the fine grid resolution that the viscous problem generally needs, since the length scales of the correction flow are generally much coarser. It can be solved by a traditional finite-difference or finite-volume method on a finite domain, in which the truncated outlet boundary is set reasonably far downstream of the airfoil Wang [1999]. However, we instead choose to solve the problem with the LGF on a Cartesian grid slightly larger than the test section, immersing the inlet and wind tunnel walls into the grid with the immersed layers method Beckers and Eldredge [2022]. Because the LGF automatically satisfies the condition $\boldsymbol{u}_{\phi,\mathrm{c}} \rightarrow \mathbf{0}$ at infinity, no explicit outlet boundary condition (4.12) is needed. The immersed wind tunnel walls are truncated around five chord lengths downstream of the airfoil.

Figure 4.3 shows the streamlines of the potential flow correction for the wind tunnel walls with a fixed inlet mass flow rate and without suction, together with the normal velocities used as boundary conditions. The streamlines indicate that the flow accelerates at the position of the airfoil toward the outlet, which is the confinement effect of the walls and which will thus be present in the combined flow $\boldsymbol{u}$ as well.

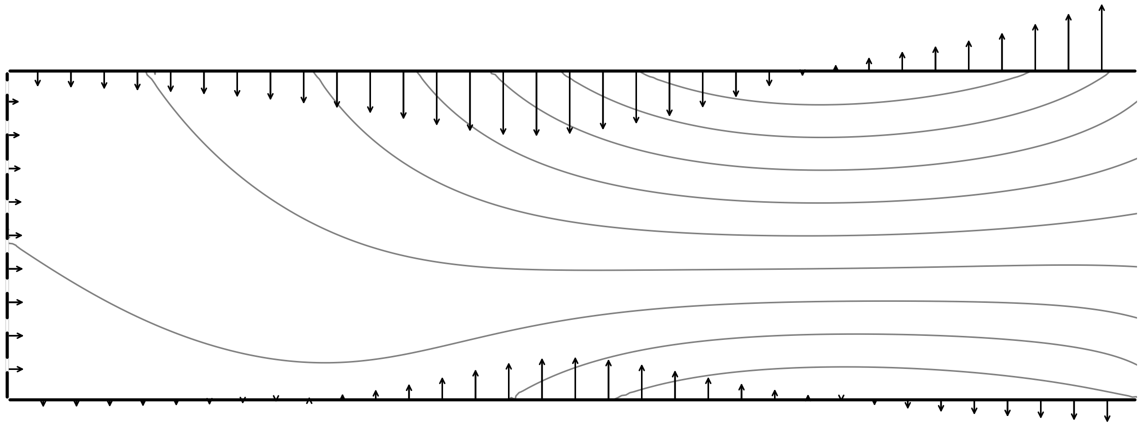


Figure 4.3: Streamlines of $\boldsymbol{u}_{\phi,\mathrm{c}}$ (grey). The arrows indicate the relative magnitude and direction of the normal component of $\boldsymbol{u}_{\phi,\mathrm{c}}$ at the walls and inlet.

To model the suction at the top wall in our framework, we can treat the entrance to the suction duct as a distribution of sinks on the wall. Due to our inviscid treatment of the walls, the resulting suction flow without the presence of the airfoil is then entirely irrotational. The airfoil's response to the suction flow emerges in $\boldsymbol{u}_{\boldsymbol{\omega}}$ through its boundary condition (4.8) and in the convective acceleration in (4.7). We can thus include the irrotational suction flow in $\boldsymbol{u}_{\phi,\mathrm{s}}$ through the boundary condition

$$\boldsymbol{n} \cdot \boldsymbol{u}_{\phi,\mathrm{s}} = V_{\mathrm{s}} \text{ on } \partial\boldsymbol{\Omega}_{\mathrm{s}}, \tag{4.13}$$

where V_{s} is a prescribed suction velocity. The suction flow field furthermore satisfies no-flow-through on the other sections,

$$\boldsymbol{n} \cdot \boldsymbol{u}_{\phi,\mathrm{s}} = 0 \text{ on } \partial\boldsymbol{\Omega}_{\mathrm{wt}} \cup \partial\boldsymbol{\Omega}_{\mathrm{in}}, \tag{4.14}$$

and a similar boundary condition downstream of the test section,

$$\boldsymbol{n} \cdot \boldsymbol{u}_{\phi,\mathrm{s}} \to -U_{\mathrm{s},\infty} \text{ as } x \to \infty, \tag{4.15}$$

where $U_{\mathrm{s},\infty}$ is equal to the suction volume flow rate divided by the height of the outlet. We could solve this problem on a finite domain with a truncated downstream boundary. As with the correction flow, the grid resolution requirement is similarly coarse. However, as we did for $\boldsymbol{u}_{\phi,\mathrm{c}}$, we solve this problem with the LGF/immersed layers method with no explicit outlet boundary.

Figure 4.4 demonstrates that, when this suction flow is active, the overall mass flow rate through the test-section outlet is reduced for a fixed inlet mass flow rate, and, depending on the location of the suction, the horizontal velocity at the test subject is also potentially reduced.

Note that the viscous flow depends on the potential flow and vice versa. In the presented framework, one can implement the coupled problem monolithically or in a weakly-coupled way. In the examples in this work, we choose the latter and exchange information between the viscous flow solver and the potential flow solver at every stage of the time-stepping algorithm.

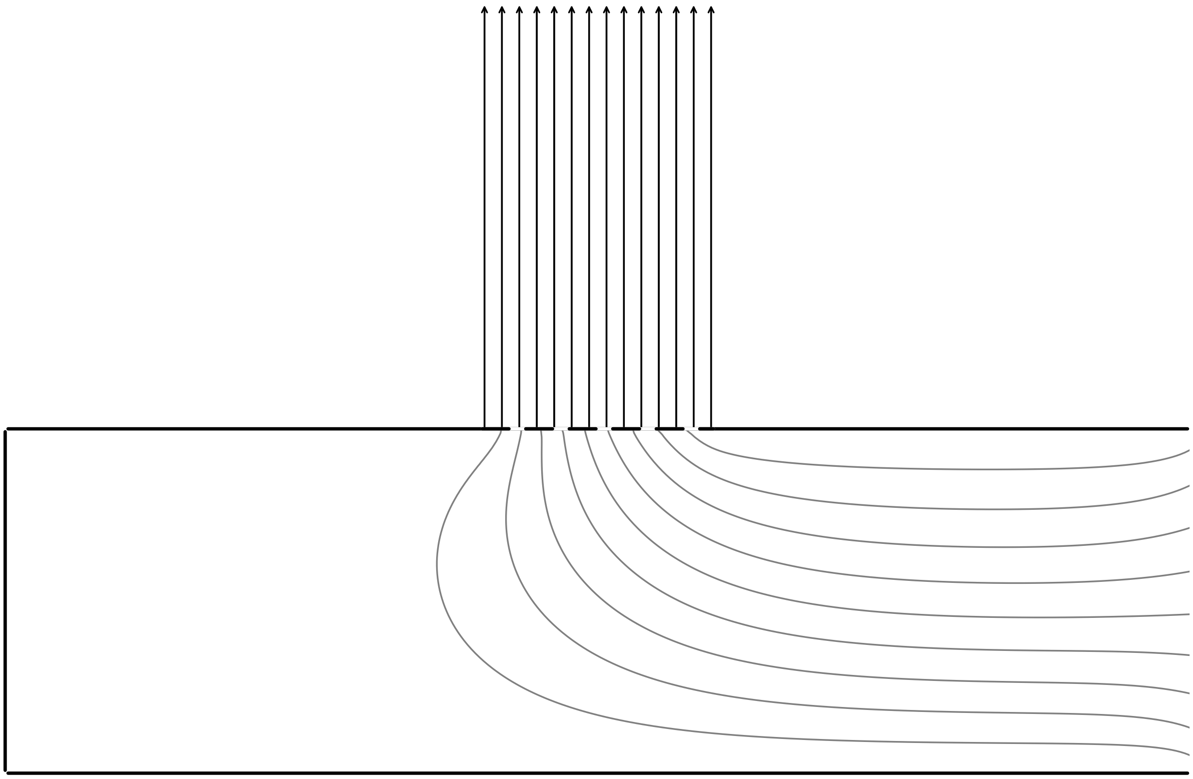


Figure 4.4: Streamlines of $\boldsymbol{u}_{\phi,\mathrm{s}}$ (grey). The arrows indicate the relative magnitude and direction of the normal component of $\boldsymbol{u}_{\phi,\mathrm{s}}$ at the suction boundary.

4.1.3 Gust model in free space and the modified Wagner response

Irrotational gusts can straightforwardly be modeled with a potential flow, just as we did by using potential-flow boundary conditions in our viscous flow solver. If we consider a flat plate at zero angle of attack in our previously introduced wind tunnel model, the setup is similar to that of the Wagner configuration and differs only in the manner in which unsteady forcing is introduced: in the computational wind tunnel, it is created through the suction of flow around a stationary airfoil; in the Wagner response, it is introduced through the airfoil's motion. To compare the simulated wind tunnel gust response to the Wagner response, we need to interpret the gust as a vertical motion.

Figure 4.5 shows the velocity $\boldsymbol{u}_\phi$ and its components during suction for a wind tunnel

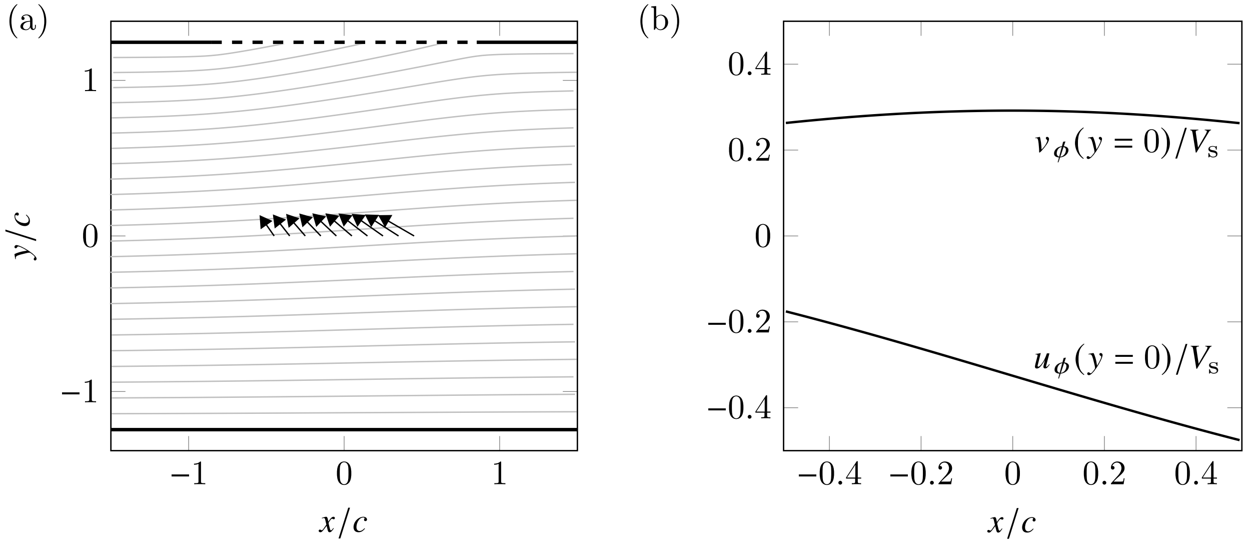


Figure 4.5: Distribution of $\boldsymbol{u}_\phi$ along the airfoil's location (with $\boldsymbol{u}_{\boldsymbol{\omega}} = 0$) during suction, represented by (a) its vectors and (b) the magnitude of its components scaled by the suction velocity.

without an airfoil at the location where the airfoil would be. The vertical component is almost constant over the airfoil length for the given wind tunnel geometry. The horizontal component varies slightly more, but we can reasonably approximate both components of the suction flow in our wind tunnel reference frame with the components of a time-varying uniform flow $\boldsymbol{U}_\mathrm{O}$ equal to $\boldsymbol{u}_\phi(\boldsymbol{x}_\mathrm{O}, t) + \boldsymbol{U}_\infty$ evaluated in the airfoil-free wind tunnel, with $\boldsymbol{x}_\mathrm{O}$ the center of the airfoil. This approximation provides us with an easy method to simulate the same gust without wind tunnel walls or, equivalently, in free space.

In an inertial reference frame, where the fluid would be at rest at infinity, the airfoil experiences the same approximate flow if it would move at a velocity $\boldsymbol{u}_b = (-U_b, V_b)$ that is equal and opposite to $\boldsymbol{U}_\mathrm{O}$, as illustrated in Figure 4.6. To compare the force and moment responses in both reference frames, one needs to account for a buoyant force $\rho \mathcal{V}_b \dot{\boldsymbol{U}}_\mathrm{O}$ on a body with volume $\mathcal{V}_b$ due to the accelerating freestream.

The Wagner formulation (2.17) is expressed in the inertial reference frame but does not

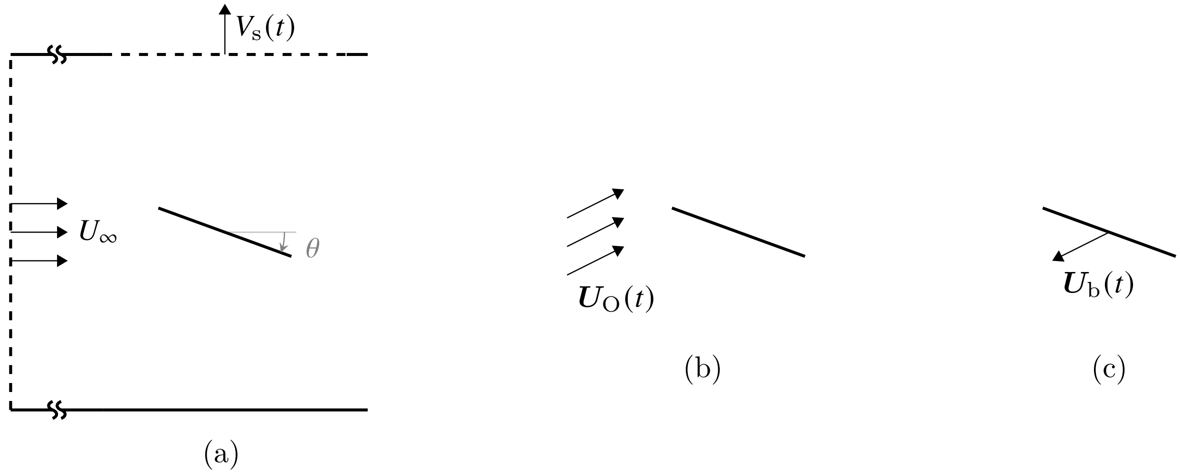


Figure 4.6: Illustration of (a) the wind tunnel gust setup and the approximating setups used in this work: (b) a stationary airfoil in an unbounded, time-varying uniform flow and (c) an accelerating airfoil in an unbounded fluid that is at rest at infinity.

account for the variation in streamwise velocity during the gust, which is of the same order as the vertical velocity variation. Furthermore, we would like to account for an airfoil at high angle of attack. For this purpose, we make three modifications to the previous formulation. Firstly, we consider the Wagner response as a perturbation on top of the steady lift $f_{\mathrm{ss},y}$ of the viscous flow we want to compare it to, which usually differs from the steady lift $\pi\rho U_b c \sin\alpha$ predicted by potential flow theory. Secondly, we account for the angle of attack in the added mass force by computing the translational added mass matrix M^{FV} for the force on a flat plate in free space as

$$\begin{aligned} M^{FV} &= \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & \frac{\pi}{4}\rho c^2 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \\ &= \frac{\pi}{4}\rho c^2 \begin{bmatrix} \sin^2\theta & -\sin\theta\cos\theta \\ -\sin\theta\cos\theta & \cos^2\theta \end{bmatrix}. \end{aligned} \tag{4.16}$$

Finally, to account for streamwise velocity variations, we include the time-varying $U_b(t) = U_\mathrm{O}(t)$ in $\Gamma^{\mathrm{qs}} := \pi c[V_b(t) + U_b(t)\theta]$ and we compute a convected time t^+ based on $U_b(t)$ instead of U_∞ as

$$t^+ = \int_0^t \frac{U_b(\tau)}{c} \mathrm{d}\tau, \tag{4.17}$$

which is equivalent to the thin-airfoil approach of van der Wall and Leishman van der Wall and Leishman [1994] for arbitrary free-stream velocity fluctuations. The expression for the modified Wagner equation in the wind-tunnel problem then becomes:

$$f_y(t^+) = f_{\mathrm{ss},y} - M^{FV}\dot{\boldsymbol{U}}_b(t^+) - \rho U_b(t^+) \int_0^{t^+} \dot{\Gamma}^{\mathrm{qs}}(\tau^+)\Phi(t^+ - \tau^+)\mathrm{d}\tau^+. \tag{4.18}$$

In the next section, we numerically compute the lift using this formulation and use Jones' Jones [1938] approximation (2.18) for the indicial response function.

4.2 Investigation of wind tunnel effects

In this section, we demonstrate the wind tunnel framework by simulating two types of suction gusts in a two-dimensional setting. First, we examine the response of several airfoil configurations to a suction gust that has a Gaussian time profile. Afterward, we do the same exercise for a suction gust that arises from impulsively opening and closing the suction duct, but where we infer the time profile of V_s from synthetic measurements that mimic real experimental data. The two-dimensional wind tunnel geometry in these examples is chosen to match the longitudinal section of the AFUWT test section (2.1 m × 0.61 m), scaled by a representative chord length $c = 0.245\,\mathrm{m}$ of an airfoil He and Williams [2020], which is positioned at the center of the test section. For each simulation, the Reynolds number $\mathrm{Re} = U_\infty c/\nu$ is equal to 400, the grid cell size is $\Delta x = 2.5c/\mathrm{Re}$, and the time step size is $\Delta t = 0.5\Delta x/U_\infty$.

4.2.1 Gaussian gust response at zero angle of attack

In our first example, we compare the responses to a known gust of a flat plate and a NACA0009 airfoil at a zero-degree angle of attack. Consider a suction velocity with a

Gaussian profile in time that is uniformly distributed over a width of $1.714c$ on the top wall centrally above the airfoil. As in equation (4.11), we fix the inlet velocity to be equal to the free stream velocity U_∞, and we formulate the gust strength in terms of a suction ratio $\dot{\mathcal{V}}_{\mathrm{s}}/\dot{\mathcal{V}}_{\mathrm{in}} = 0.2\exp[-\left(t^* - t_0^*\right)^2/b^2]$, where $\dot{\mathcal{V}} = \int_{\partial\Omega} \boldsymbol{u} \cdot \boldsymbol{n} \mathrm{d}A$, $b = 0.15$, and $t^* = U_\infty t/c$. In the following examples, we first allow the impulsively-started simulation to reach a steady state before starting the gust, which is centered at $t_0^* = 10$.

Figure 4.7 shows the streamlines and Q contours for the flow near the airfoil, where $Q = \frac{1}{2}(||\boldsymbol{\Omega}||^2 - ||\boldsymbol{S}||^2)$ and $\boldsymbol{S}$ and $\boldsymbol{\Omega}$ are the symmetric and anti-symmetric components of $\nabla\boldsymbol{u}$. The contours indicate that the features of the flow during and after the gust for both the flat plate and NACA airfoil are comparable.

The lift response of the flat plate and NACA0009 airfoil in our wind tunnel can be broken down into four components: the added mass force, the circulatory force, the viscous force, and the buoyancy force due to the time-varying free stream, which is zero for the infinitely-thin flat plate. The viscous lift force is typically negligible, and the added mass force depends on the entire wind tunnel geometry but can be approximated by that of the modified Wagner response (4.18) (which is the same as that for the flat plate in the equivalent uniform flow). The circulatory lift force can then be approximated for each case by subtracting the other contributors from the total lift. The lift response and its contributors are shown in Figure 4.8 in terms of the lift coefficient $c_L = 2f_y/\left(\rho U_\infty^2 c\right)$ for both airfoils in the wind tunnel, a flat plate in an equivalent uniform flow (without the wind tunnel walls), and the Wagner response. The lift response for the flat plate predicted by the wind tunnel model and equivalent uniform-flow model (the two viscous models) is nearly identical, indicating that the wind tunnel effects are negligible for small angles of attack and low suction strength. Interestingly, the Wagner response predicts the same peak lift as the viscous models. After the initial increase in suction, however, the Wagner response overpredicts the circulatory lift (provided that the added mass forces are indeed similar for each case). Part of this difference in behavior is because classical unsteady aerodynamics only models linear behavior, while the vortex-shedding mechanism that drives the circulatory force is typically non-linear. Furthermore, the difference between the flat plate and NACA0009 airfoil responses is mainly due to the buoyancy force on the latter airfoil.

4.2.2 Gaussian gust response with flow separation

Suppose we increase the effective angle of attack by positioning the airfoil at an angle or creating a stronger gust. In that case, one can expect more substantial effects from flow non-linearities and the wind tunnel confinement, resulting in increasingly different behavior between the different models.

To illustrate these effects, we change the angle of attack α to 10° and increase the maximum $\dot{\mathcal{V}}_\mathrm{s}/\dot{\mathcal{V}}_\mathrm{in}$ to 0.4. The angle of the airfoil with the positive x-axis is then $\theta = -\alpha$, which is positive for counterclockwise rotations. Figure 4.9 compares the flow response of a flat plate in a wind tunnel with that of a flat plate in an equivalent, time-varying uniform flow. The snapshots show the same flow features as in the previous example, with the exception that the flow is now fully separated during the gust. The flows with and without the wind tunnel look almost identical. We look at the lift response to observe their differences.

Looking at the lift response in Figure 4.10 and the circulatory contribution in Figure 4.10, it is clear that the main difference between the responses with and without the wind tunnel lies in the steady-state value of the circulatory force. This force is higher for the case with the wind tunnel, which can be attributed to the confinement effect of the wind tunnel walls, considering that the wind tunnel inlet velocity is U_∞. The higher angle of attack partly obstructs the wind tunnel flow, which significantly increases the flow velocity around the airfoil, leading to a higher lift. Aside from the steady-state lift difference, the lift responses of the viscous models are again almost identical. The modified Wagner model is applied here as a perturbation to the steady-state lift of the wind tunnel model. Its steady-state lift $f_{\mathrm{ss},y}$ is set to the total lift of the flat plate at $\alpha = 10°$ in the wind tunnel model right before the gust is introduced. However, it overpredicts the lift response, most likely as a result of the flow completely separating during the gust, which is a highly non-linear effect that the Wagner model cannot account for.

4.2.3 Realistic step opening and closing of the suction duct

In the previous examples, we assumed the gust profile and strength were known. Now we look at a more realistic example where the time profile of the gust is estimated from measurements. Since no experimental data have been published yet for an airfoil's gust response in the AFUWT setup using the suction duct, we will generate synthetic measurements that are loosely based on the experiments by He et al. He et al. [2022] of impulsively opening and closing the suction duct louvers without an airfoil. They measure the vertical velocity at the center of the test section, but because our airfoil is positioned at the center, we consider measurements upstream of the airfoil. Specifically, we create a synthetic, noisy signal that simulates the vertical velocity measurements $z = V(\boldsymbol{x}_{\text{probe}}) + \nu$ of a crosswire probe positioned at a quarter of the length of the test section ($2.14c$ upstream from the center of the airfoil), with the noise ν normally distributed with variance σ_ν^2 and zero mean. A random-walk state-space model in combination with a Kalman filter is used to estimate the parameter $\hat{\theta}_{k|k} = V_{\text{s}}/U_\infty$ and its associated variance σ_ω^2 at every time t_k that the measurement signal was sampled. For illustrative purposes, we arbitrarily choose a $\sigma_\nu^2 = 1 \times 10^{-7}$ and $\sigma_\omega^2 = 1 \times 10^{-7}$. For more details on the Kalman filter equations and this data assimilation approach, we refer the reader to appendix C.

The measurements and the resulting estimated signal are presented graphically in Figure 4.11. The figure also shows the estimated flow acceleration at the center of the wind tunnel $\dot{V}_{\text{O}}$ which amplifies the fluctuations and uncertainty in V_{s} through numerical differentiation, which is performed at the time steps of the flow solver using linear interpolation to find the velocity values between successive measurements. This acceleration appears in c_L, scaled by $\pi/4$, through the added mass contribution.

The flow-field snapshots in Figure 4.12 show that, for this gust, the flow separates by the time the suction reaches its steady-state value if we model the wind tunnel environment. This does not happen if we only model the airfoil in the equivalent uniform flow, indicating that the wind tunnel walls (or concentrated suction) can alter the features of the flow. The associated lift response is shown in Figure 4.13 for these two cases and the Wagner case. The effect of the flow separation on the lift response is partly obscured by the added-mass fluctuations, but it causes the lift response in the wind tunnel to decrease to below that of the unseparated uniform-flow case as the suction approaches a steady state.

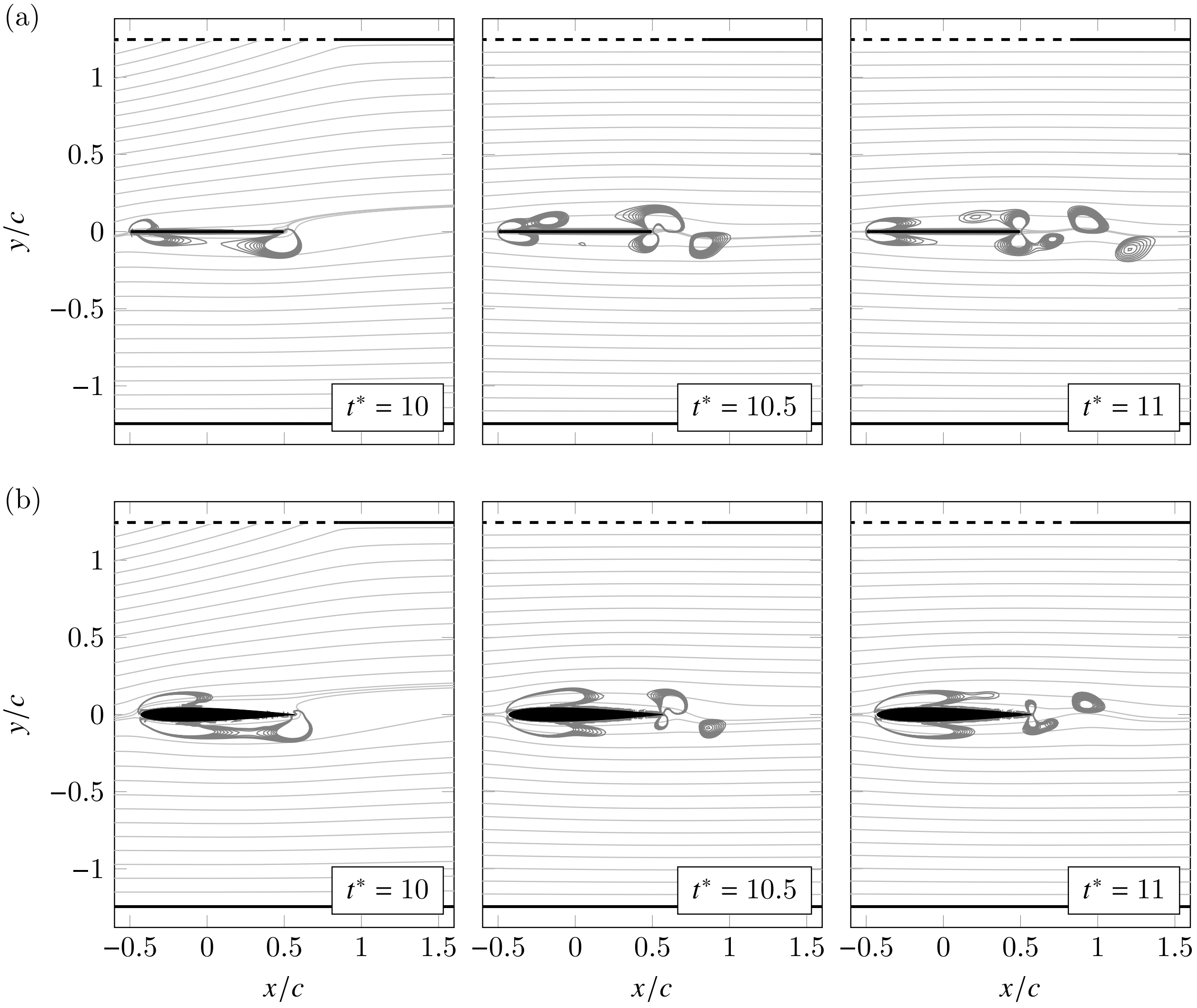


Figure 4.7: Streamlines (light grey) and Q contours (dark grey) for (a) a flat plate and (b) a NACA0009 airfoil at a zero-degree angle of attack experiencing a Gaussian gust created in a wind tunnel with suction.

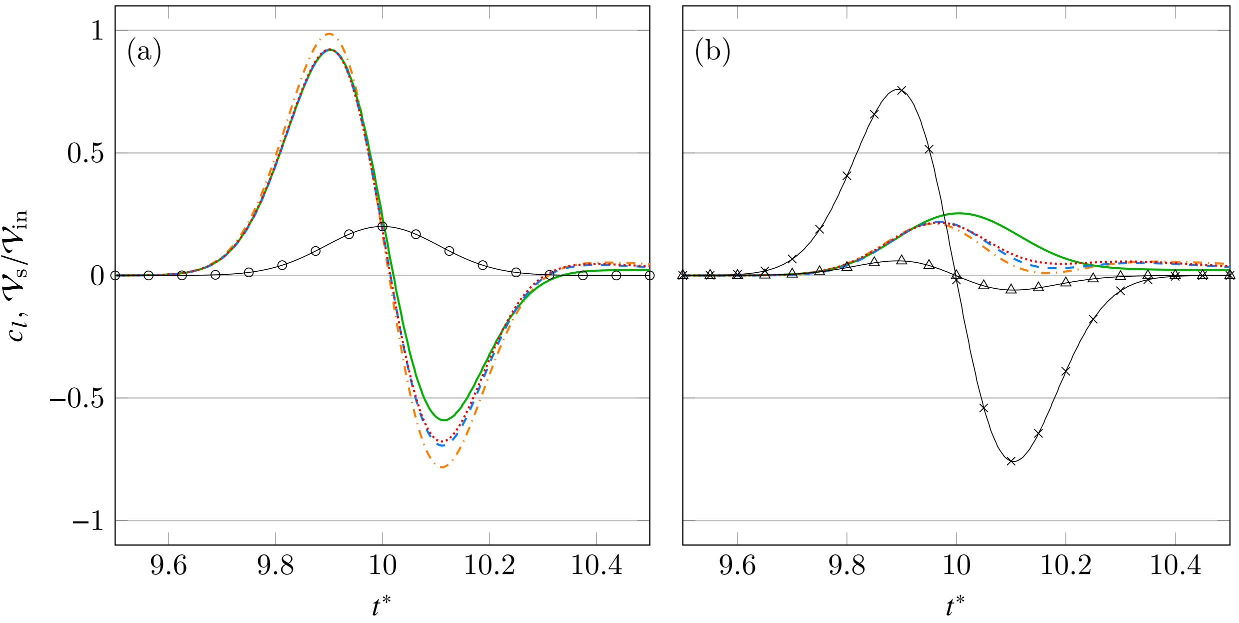


Figure 4.8: Zero-degree angle of attack lift response to a Gaussian gust with a maximum suction ratio of 0.2 for the following models: flat plate in the wind tunnel (⋯⋯), NACA0009 in the wind tunnel (-·-·-), flat plate in an equivalent uniform flow (- - -), and the modified Wagner model (——). Panel (a) shows the total lift responses (colored lines without markers) and the suction ratio (—○—), and panel (b) shows the contributors: circulatory lift (colored lines without markers), added mass lift (—×—), and the buoyancy lift for the NACA0009 airfoil (—△—).

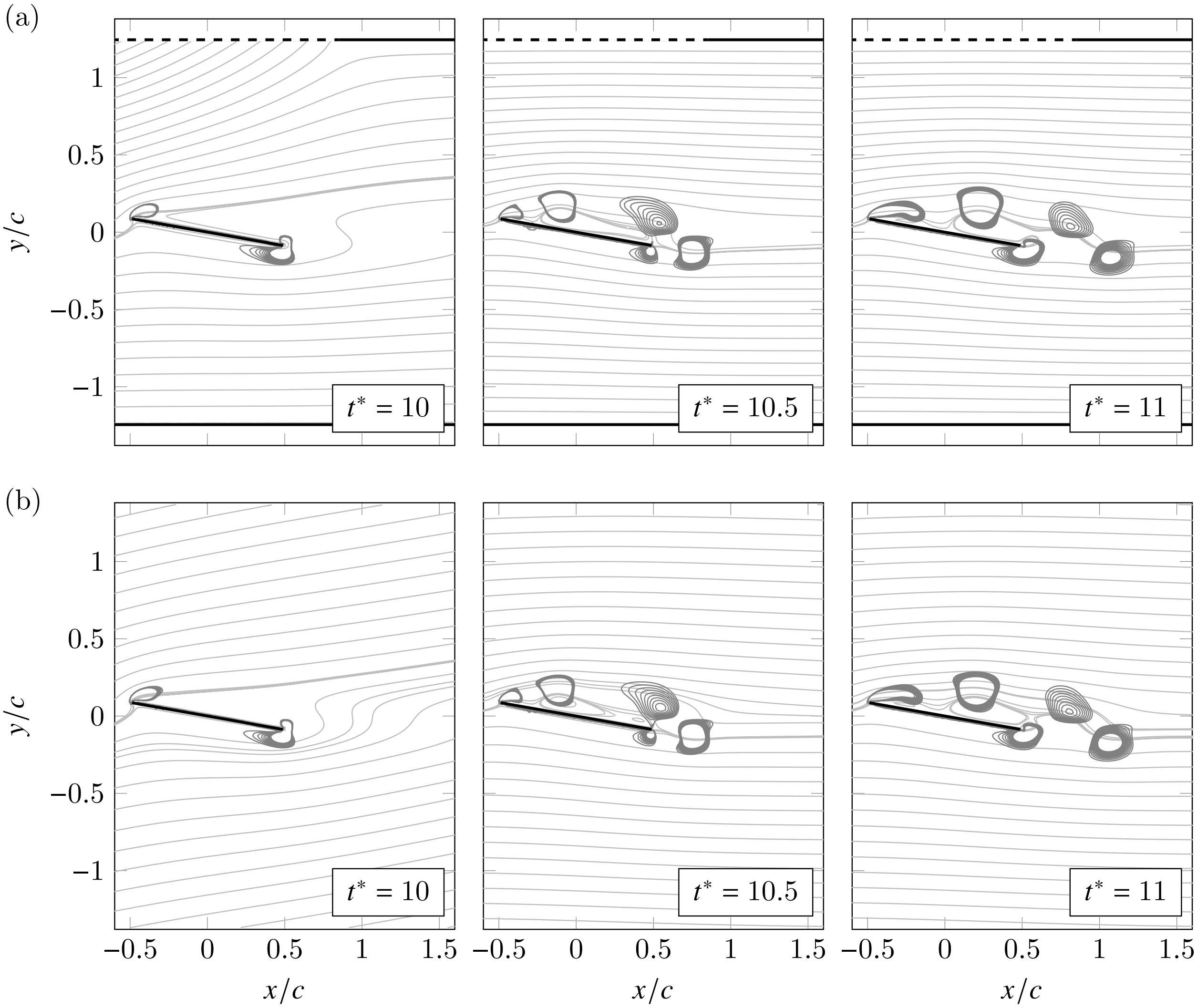


Figure 4.9: Streamlines (light grey) and Q contours (dark grey) for a flat plate at a ten-degree angle of attack experiencing a Gaussian gust created (a) in a wind tunnel through suction and (b) in an equivalent time-varying uniform flow.

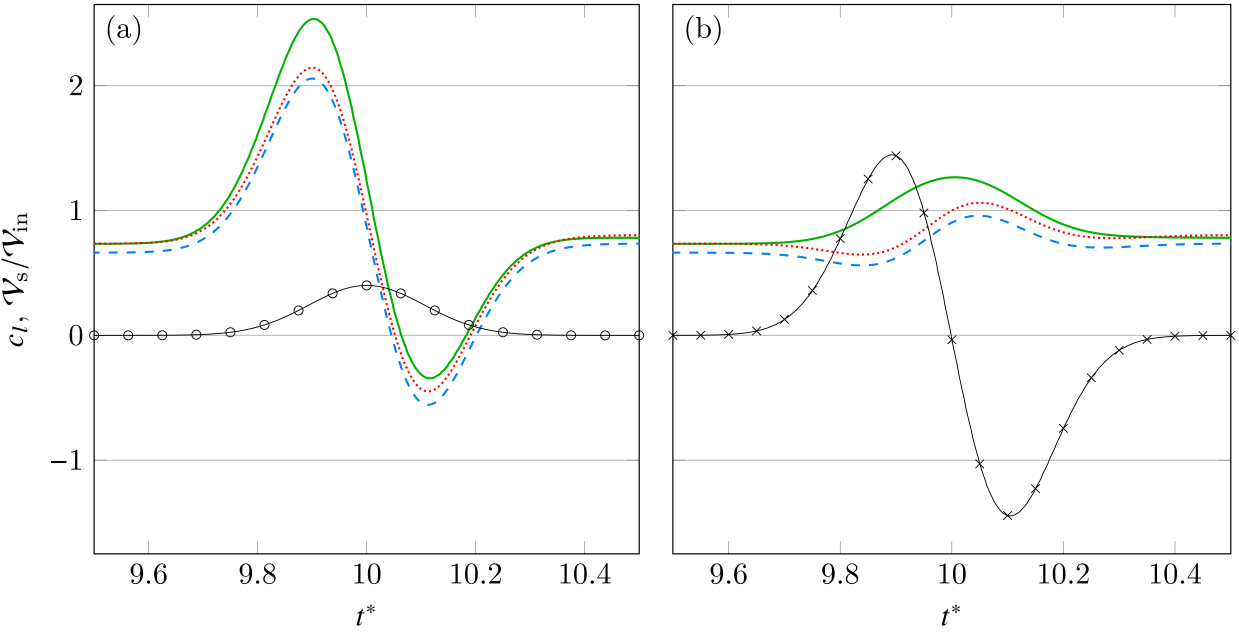


Figure 4.10: Ten-degree angle of attack lift response to a Gaussian gust with a maximum suction ratio of 0.4 for the following models: flat plate in the wind tunnel (⋯⋯), flat plate in an equivalent uniform flow (- - -), and the modified Wagner model (——). Panel (a) shows the total lift responses (colored lines without markers) and the suction ratio (—○—), and panel (b) shows the contributors: circulatory lift (colored lines without markers) and the added mass lift (—×—).

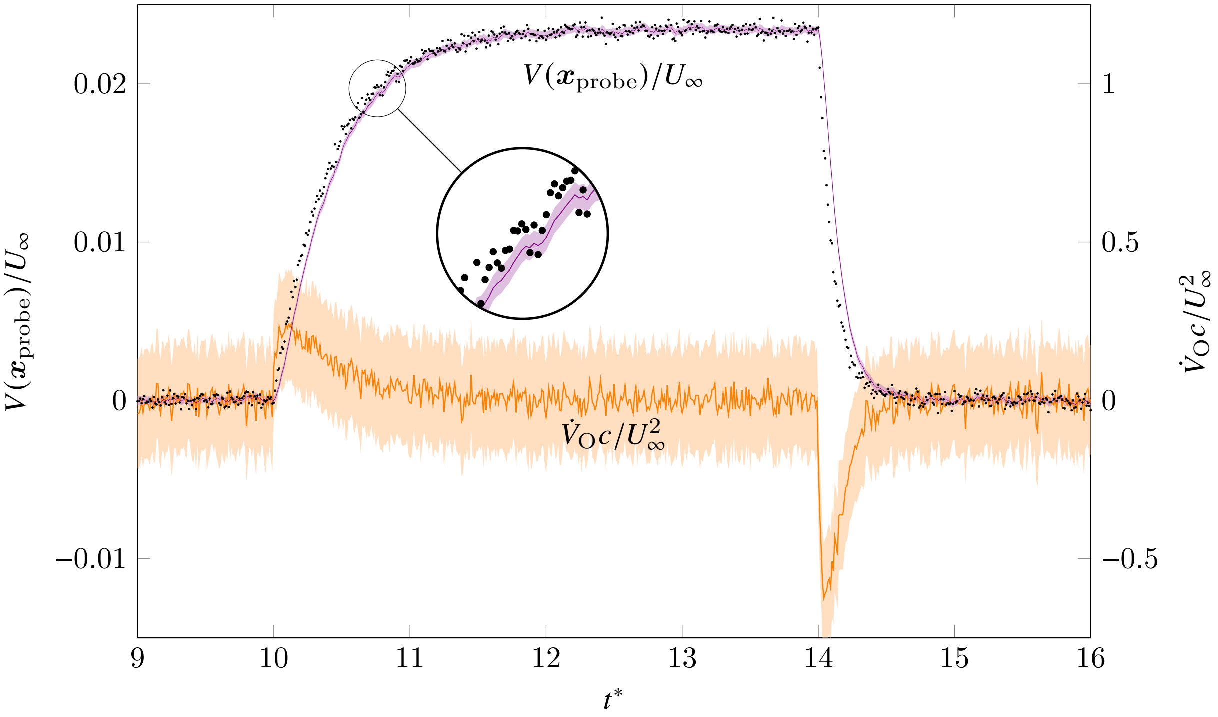


Figure 4.11: Vertical velocity measurement data (dots) and the filtered measurement signal (——) at the location of the probe, and the derived vertical acceleration (——) at the center of the wind tunnel, including their 95 % confidence intervals.

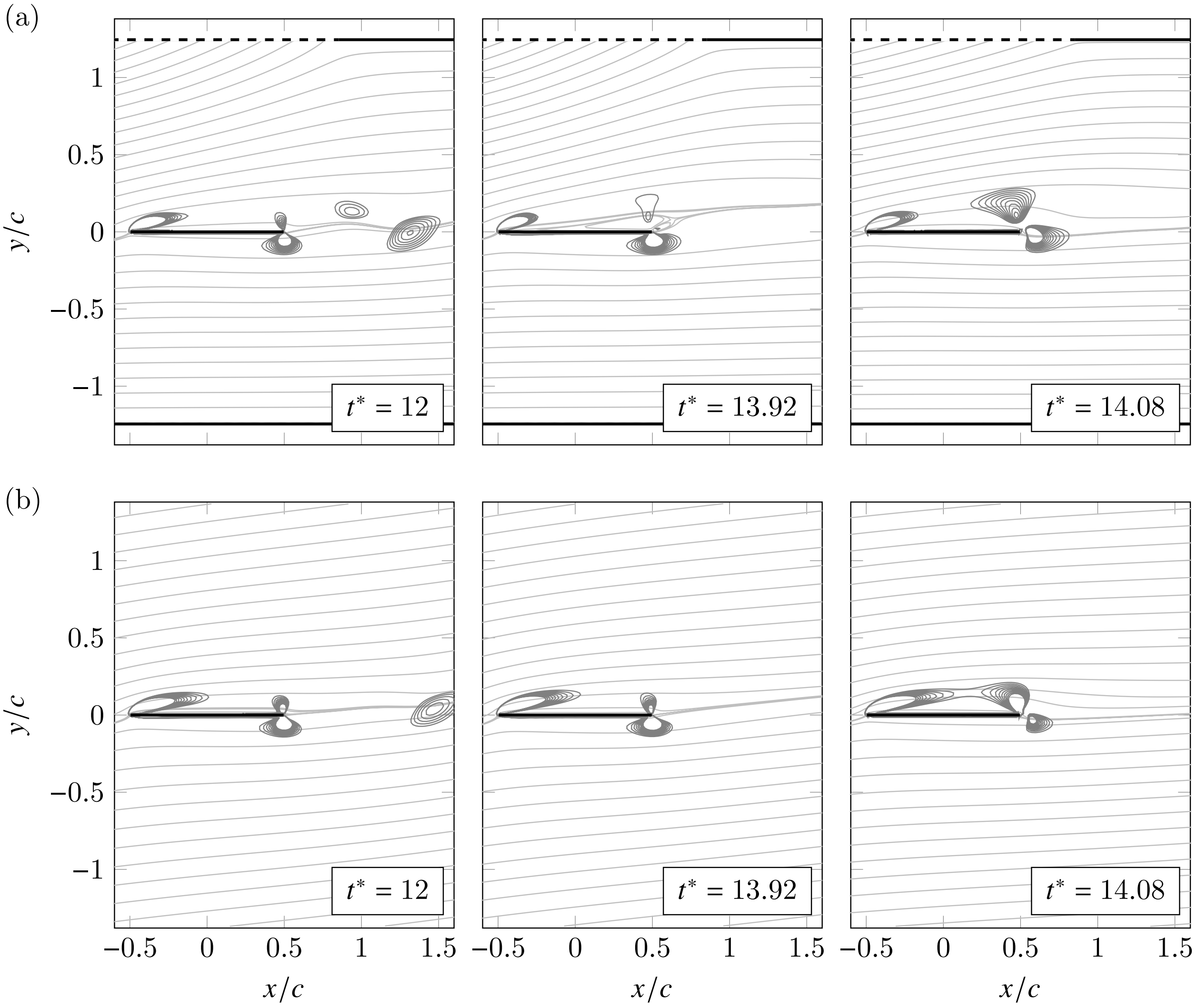


Figure 4.12: Streamlines (light grey) and Q contours (dark grey) for a flat plate at a zero-degree angle of attack experiencing a gust created (a) in a wind tunnel through impulsively opening and closing the suction duct and (b) in an unbounded fluid through an equivalent, time-varying uniform flow.

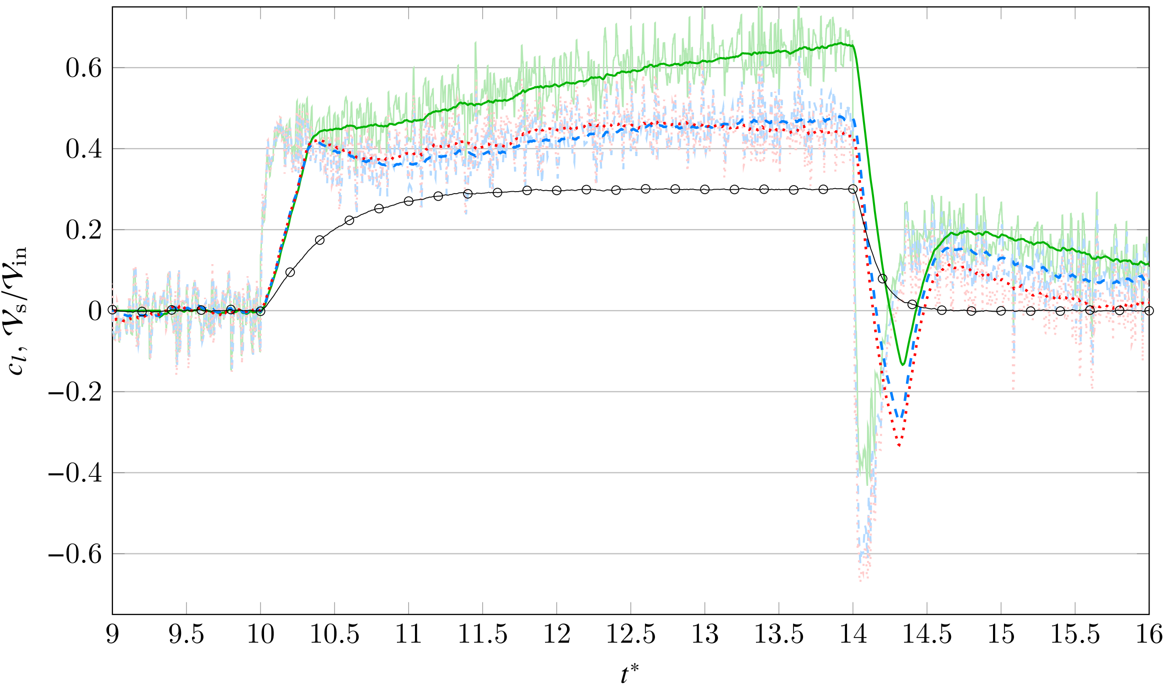


Figure 4.13: Zero-degree angle of attack lift response (light color) and its 50-timestep-window moving average (dark color) to the indicated suction ratio (—○—) for the following models: a flat plate in the wind tunnel (⋯⋯), a flat plate in an equivalent uniform flow (- - -), and the Wagner model (——).

CHAPTER 5

Deep reinforcement learning of airfoil pitch control

In this chapter, we apply reinforcement learning to learn a control strategy for an airfoil's pitch angle while disturbances or a change in the reference trajectory are introduced, resulting in an unsteady flow. When the disturbances or pitch variations are strong enough, the flow will exhibit nonlinear behavior, complicating the control design. Deep reinforcement learning is a powerful tool that can leverage neural networks to provide nonlinear control in response to nonlinear flow variations. The focus here is to explore its use in designing a control strategy, or policy, for a viscous flow system at low Reynolds number, but first, we will introduce the reinforcement learning setup and turn our attention to the equivalent problem in the classical unsteady aerodynamics context introduced in Section 2.3.1. In this setup, the problem becomes much simpler to train and analyze since it consists of a linear system with only two states. Afterward, we apply the same framework to a viscous flow, modeled with the body-fixed reference frame immersed-boundary method described in Section 2.4, and analyze its performance for a test case using a point force pulse as disturbance.

5.1 Flow disturbances and the control problem

Consider an inertial reference frame with coordinate system (x, y) and a flat plate airfoil oriented at an angle α with the negative x-axis (positive in the clockwise direction) and with its centroid positioned at $\mathbf{X}_C$ in the inertial reference frame. We describe the kinematics of the airfoil in the inertial reference frame in terms of the velocity $\dot{\mathbf{X}}_P$ of a pivot point, which we will place at the mid-chord point, and the angular velocity $\dot{\alpha}$ about this pivot point.

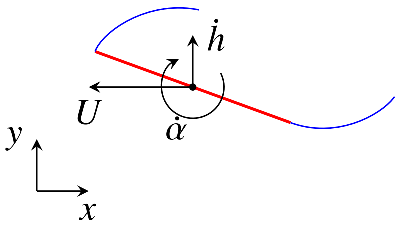


Figure 5.1: Diagram of the airfoil and wake.

The airfoil moves forward in the negative x-direction with a constant flight speed U, while it possibly encounters flow disturbances. The disturbances we consider in this work can consist of random vertical freestream disturbances $V_\infty(t)$ or local flow disturbances that are inserted through a point force $\mathbf{F}(\boldsymbol{x}, t)$ with random parameters. Note that for a flat plate airfoil, a variation in the vertical freestream has the same effect on the relative flow as an equal but opposite vertical motion $\dot{h} = -V_\infty$ of the airfoil. Thus, in this work, we will assume the airfoil's pivot point can undergo random variations of its vertical motion and its velocity is thus $\dot{\mathbf{X}}_P = (-U, \dot{h}(t))$, as shown in Figure 5.1.

The control problem is to design the partial-state feedback controller $\mathbf{K}$ in Figure 5.2 that manipulates the flow around the airfoil, represented by the system $\mathbf{G}$, through pitching, by controlling the angular acceleration $\ddot{\alpha}$, such that the lift force tracks a prescribed reference lift (the servo problem) and that lift variations due to the disturbances are minimized (the regulator problem), while limiting the control signal to a prescribed range. In this work we will only focus on the regulator problem, but the servo problem is a straightforward extension. The feedback consists of the observation vector O, which contains only specific parts of the state vector S of the system. Note that in a real application, this control problem should be extended to account for model uncertainties and measurement noise. However, in this work, we will assume that we have a perfect model of the system dynamics and observe the states or output without noise. In other words, we look at the nominal stability and performance

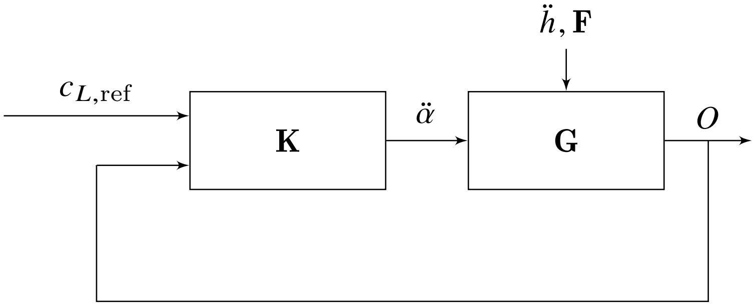


Figure 5.2: Closed-loop control diagram.

of the system. In this case, we can afford to use a feedback controller with high bandwidth, leading to fast rise times but also high sensitivity to noise and parameter variations. Such a controller should, therefore, only be viewed as an upper limit for performance (i.e. without any guarantees of robustness). Lastly, we will not directly measure the disturbances. If we would, combined feedback and feedforward controllers (two-degree-of-freedom controllers) could perfectly eliminate the disturbance in the case of a linear aerodynamic model [Åström and Murray, 2021].

Three main approaches exist for the design of a controller [Skogestad and Postlethwaite, 2005]: the signal-based approach (or time-domain approach), loop-shaping (or frequency-domain approach), and numerical optimization. Most controller designs follow one of these approaches, but the distinction is not always clear and sometimes a combination of approaches is used.

In the signal-based approach, we assume we have some information available about the disturbance. In the context of unsteady aerodynamics, this could mean that we know the shape or time profile (but perhaps not the parameters) of the flow disturbance (e.g. sinusoidal or one of the discrete canonical gust shapes), or we know that the disturbance is a stochastic signal for which we know the statistical measures. With the availability of such information, the disturbance can be modeled and any unknown states can be estimated by an observer. An example of this approach is the use of a linear quadratic regulator to control the vortex shedding behind an airfoil at large angle of attack [Ahuja and Rowley, 2010]. Because of the wide variety of canonical gust models and the nonlinear response of flows to large disturbances, which prevents the superposition of modeled responses to a rapid sequence of those, we will avoid the use of signal-based approaches in this work. In other words, to keep our approach generalizable, we assume that we have no information available on the disturbance.

In the loop-shaping, or frequency-based, approach, one aims first to shape the open-loop or closed-loop transfer functions to achieve specific robustness or performance objectives and then design a controller whose transfer functions approximate those shapes. This approach includes $\mathcal{H}_\infty$-synthesis, which has been applied by Brunton and Rowley [2013] to the same control problem we look in this work. Several experimental examples of this approach in the context of aerodynamic flow control have been summarized in Williams and King [2018].

The numerical optimization approach encompasses a wide array of controller design techniques. Such techniques employs computational methods to optimize a metric that combines control objectives and constraints, which otherwise would be intractable to solve. The evolving landscape of these techniques is witnessing rapid advancement, with notable contributions emerging from domains like machine learning and data science [Brunton and Kutz, 2022]. As a result, these methods offer effective means to address the complexities inherent in nonlinear systems, making them ideal for use in flow control. Some examples include model predictive control of cylinder rotation to control the wake [Bieker et al., 2020], iterative maneuver optimization for pitch control to regulate the lift during gust encounters Xu et al. [2023], and the use of reinforcement learning for active flow control [Vignon et al., 2023].

Here we look at the use of deep reinforcement learning of the pitch control in two types of environments: a classical unsteady aerodynamics environment and a viscous flow environment.

5.2 Reinforcement learning framework

We apply deep reinforcement learning to the previously introduced control problem using discrete-time, finite-horizon episodes with randomly-generated disturbances. The reinforcement learning framework in this work consists of an environment with a single *agent*, which uses a learning algorithm to learn a control strategy, or policy. For simplicity, the agent learns its policy only through interactions with the environment's flow model and does not rely on a reduced- or low-order model for model-based training. Interacting with the viscous flow environment is expensive to simulate and warrants using an off-policy learning algorithm designed to make the most efficient use of each sampled interaction (for a model-free training algorithm). For this reason, and because we desire a control law for a continuous-valued action, the training uses the twin delayed deep deterministic policy gradient (TD3) algorithm [Fujimoto et al., 2018], which we found to perform better in this problem than alternative state-of-the-art, off-policy, continuous-action algorithms, such as soft actor-critic [Haarnoja et al., 2018].

The training algorithm stores interactions in a standard first-in-first-out replay buffer, which can store up to 200000 interactions. The training starts after 10000 interactions have been generated. During training, the algorithm randomly samples 256 interactions from the replay buffer for stochastic gradient descent updates of the agent's six neural networks using the Adam optimization algorithm with a learning rate of 0.001. These networks are two policy, or actor, networks and their target networks and one state-action value (Q-function) approximation, or critic, network and its target network. These networks all have the same feedforward architecture: an input layer whose size depends on the observations, two fully-connected hidden layers with 400 and 300 rectified linear units (ReLU), respectively, and an output layer representing the action in the case of an actor network or the state-action value in case of a critic network. The training uses a discount factor of 0.98 for future rewards and applies target policy smoothing by adding a zero-mean Gaussian noise with a standard

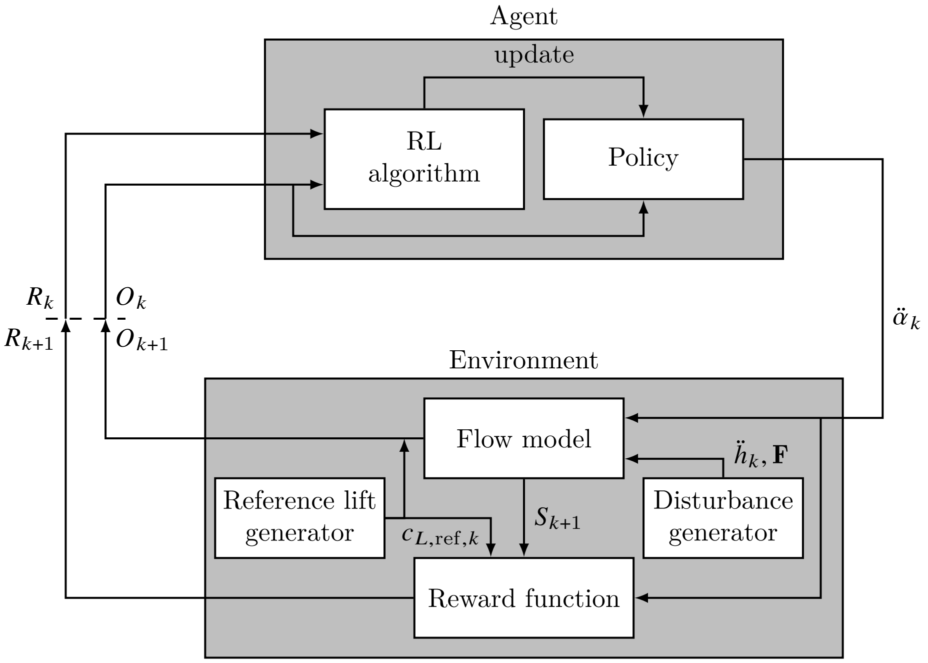


Figure 5.3: Diagram of the reinforcement learning framework.

deviation of 0.1 to the action generated by the target critic network. Lastly, the algorithm adds zero-mean Gaussian noise with a standard deviation of 0.1 to the actions of the policy when collecting training data to promote the exploration of the state and action spaces.

The environment contains all setup elements that the agent cannot control. These are the flow model, the reward function, the reference lift, and the disturbances, as shown in the diagram in Figure 5.3.

The flow model can be of any fidelity, and we will look at a classical unsteady aerodynamics model and a viscous, low-Reynolds number flow model. Its only requirements are that it should accept the angular acceleration $\ddot{\alpha}$, the vertical pivot acceleration $\ddot{h}$, and any other disturbances as its input and that we have access to the state vector S describing the flow and airfoil. If the agent observes this state vector, the control problem is a Markov decision process (MDP). However, in fluid dynamics, it is often unfeasible to observe the entire state of the wake around the airfoil, and sensor data could be limited to, e.g., only pressure, force, and airfoil kinematic info. In that case, the agent has to make control decisions based on observations O that don't contain all the state information of the environment, and the problem is called a partially observable Markov decision process. This could result in less optimal solutions than if the agent observed the entire state, but it often makes the problem computationally more tractable.

The reward function maps the state of the flow and airfoil, the control input, and the reference lift value to a scalar reward for the agent. The formulation of this reward has a significant impact on the training, and we will discuss different implementations in the following sections. Lastly, the environment contains the reference lift and disturbance generators. These can generate a reference lift value and a vertical acceleration or point force that can be randomized for each episode.

5.3 Training an agent in a classical unsteady aerodynamics environment

In our first training case, the environment flow model in the environment is the classical unsteady aerodynamics model introduced in Section 2.3.1, and the goal is to maintain a zero lift while random variations of $\ddot{h}$ are applied. One can easily solve this linear control problem with classical or robust control theory [Brunton and Rowley, 2013], but its purpose here is to introduce the reinforcement learning setup.

For the training data, we simulate episodes that last 20 convective times with the discrete-time state-space sytem (2.28)-(2.29) (with $d = 0$) using a timestep size $\Delta t = 0.1$. The $\ddot{h}$ variations consist of a random number (uniformly sampled from $[0, 20]$) of steps and impulses of random amplitude (uniformly sampled from $[0, \ddot{h}_{\max}]$). Because of the linearity, the variables in this system can be scaled arbitrarily (by the same constant) without changing the system's behavior. However, we will choose for our example values that match the assumptions for the model. Specifically, the $\ddot{h}$ variations will have a maximum value of $\ddot{h}_{\max} = 0.01U^2/c$, the control input $\ddot{\alpha}$ is constrained to $[-\ddot{\alpha}_{\max}, \ddot{\alpha}_{\max}]$, where $\ddot{\alpha}_{\max} = 0.1U^2/c^2$, and the lift f_y is thresholded at $f_{y,\text{thres}} = 0.01\rho c U^2$. Once the absolute value of the lift exceeds this value, the episode is terminated. Note that the action space we choose is large enough to avoid saturation of the control input. However, if the action space were smaller, the nonlinear nature of deep reinforcement learning is expected to deal adequately with the nonlinear saturation, which might not always be possible with linear control theory.

To train the agent to minimize the lift variations, the reward after every timestep consists of a positive reward for maintaining the lift within the allowed threshold and a negative reward for a lift that is not zero, which increases as the lift magnitude increases:

$$R_{k+1} = 1 - \frac{|f_{y,k}|}{f_{y,\text{thres}}} \tag{5.1}$$

Even though giving the positive part of the reward is not required to learn to minimize the lift variations, we found that the training improved if we did.

As discussed earlier, in a real flow environment, it is often intractable to have the agent observe the entire flow state and train with this data because of the high dimensionality. Furthermore, in a practical application, only limited information is usually available through, e.g., pressure sensors on the surface. The classical unsteady aerodynamics environment does not suffer from this same limitation because it only has two states representing the wake. It, therefore, allows us to try out and compare the reinforcement learning framework with three types of agents: one that observes no info about the wake, one that observes pressure, and one that observes all the system states. All the agents observe α, $\dot{\alpha}$, and f_y. Note that $\ddot{h}$ is never directly observed to mimic a realistic application where the disturbance cannot be measured. These cases are summarized in table 5.1. The pressure-observing agent observes the pressure at two points on the plate that are asymmetrically positioned w.r.t. the mid-chord point. It allows the agent to distinguish the circulatory pressure from the added-mass pressure, according to (2.15)-(2.16). Any additional pressure observations do not introduce additional independent information.

Agent type	Description	Observation vector for one timestep
No wake info	Only the lift and angular states are observed.	$[\alpha_k, \dot{\alpha}_k, f_{y,k-1}]^\top$
Full wake info	The full state of the wake is observed in addition to the lift and angular states.	$[\alpha_k, \dot{\alpha}_k, f_{y,k-1}, \alpha_{\text{eff},k-1}, \hat{\mathbf{x}}_{k-1}]^\top$
Pressure	The pressure jump at $\tilde{x}_1 = 0$ and $\tilde{x}_2 = c/4$ is observed in addition to the lift and angular states.	$[\alpha_k, \dot{\alpha}_k, f_{y,k-1}, p_{1,k-1}, p_{2,k-1}]^\top$

Table 5.1: Description and observation vectors for the three types of agents trained in the classical unsteady aerodynamics environment.

We conduct the training for each case with ten environments initialized with random seeds and train the agents with 750000 environment interactions (or steps). Figure 5.4 shows the training progression, or learning curve, in terms of the average episode return versus the number of environment interactions. The episode return represents the sum of the rewards collected over one episode. With the reward function (5.1), the maximum possible return for one episode is 200, which only occurs if the lift is zero at all times. The reported average episode return is computed every 1000 steps, with the average measured across ten episodes simulated with the latest policy version without applying action noise. The learning curves are smoothed with a moving average using a window of 20 averaged episode returns. As anticipated, the training curves show that the performance increases when the agent observes more states. In this simplified case, observing the pressure can adequately increase the agent's performance when full-wake observations are unavailable, aligning with the expectations.

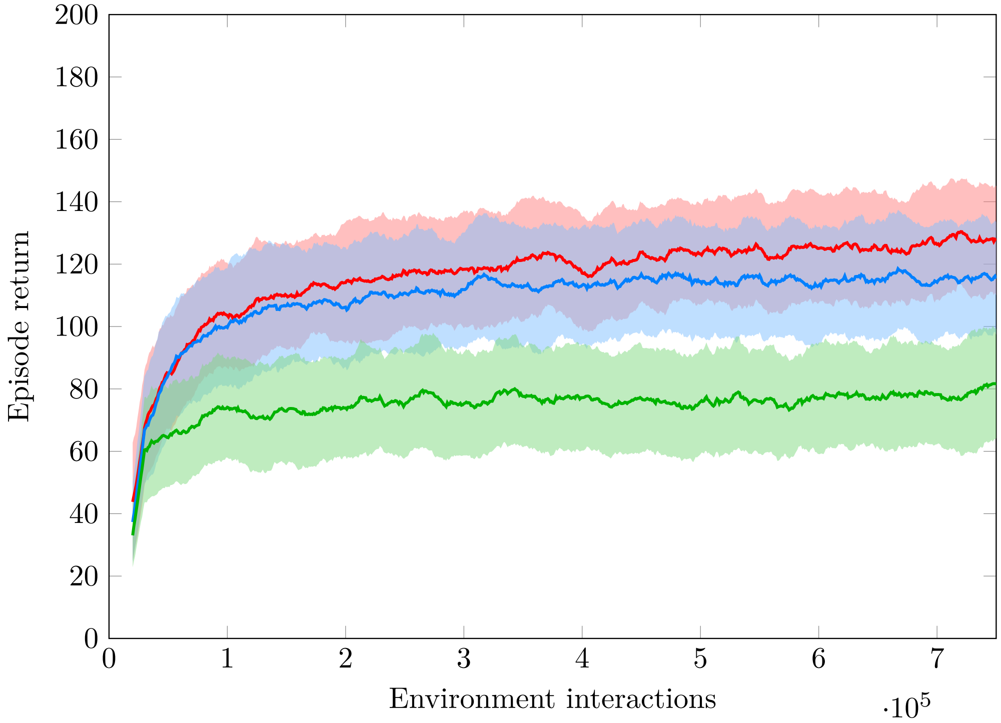


Figure 5.4: Mean (solid line) and standard deviation (shaded) of the average evaluation episode return for the three types of agents in the classical unsteady aerodynamics environment observing the most recent timestep: no wake info (—), pressure (—), and full wake info (—).

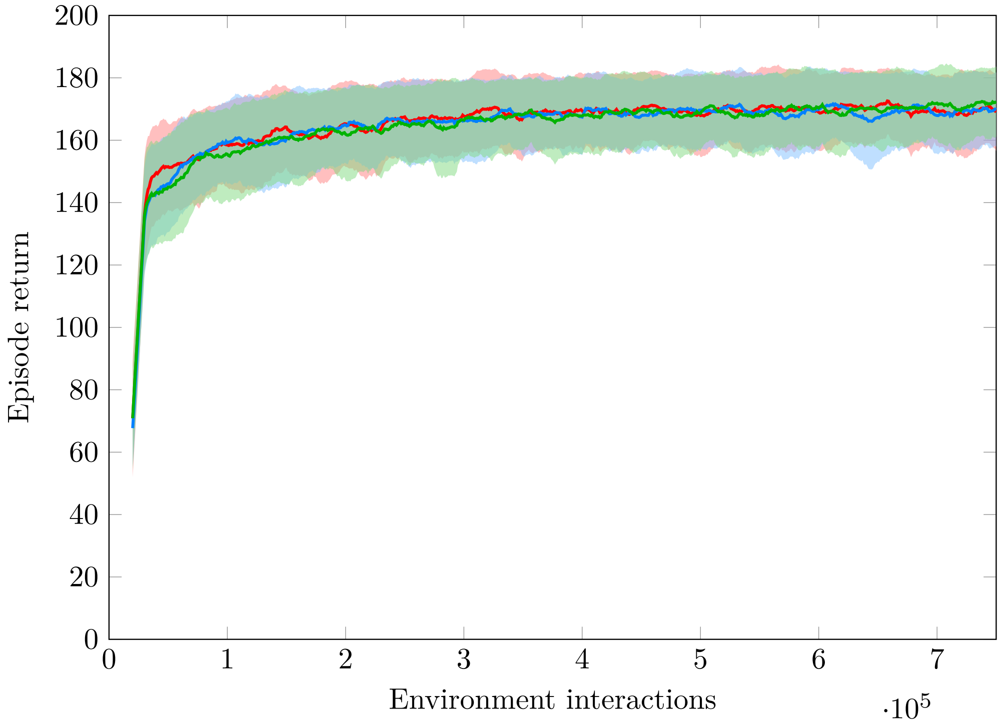


Figure 5.5: Mean (solid line) and one standard deviation (shaded) of the average evaluation episode return for the three types of agents in the classical unsteady aerodynamics environment observing the two most recent timesteps: no wake info (——), pressure (——), and full wake info (——)

Another interesting experiment is giving the agent access to the observations from the last two timesteps instead of one, similar to the reinforcement learning setup used to learn to play Atari games [Mnih et al., 2013]. Observing multiple timesteps gives access to more information about the states of the system and allows the agent to learn to make better decisions. Figure 5.5 clearly shows that the training drastically improves for each of the three cases, and their performances become indistinguishable for the chosen environment parameters.

Lastly, we select the full-wake policy and simulate an episode with the random $\ddot{h}$ variations used during training, and we compare it with an arbitrarily chosen proportional control law $\ddot{\alpha}/\ddot{\alpha}_{\text{max}} = 1.71 f_y/f_{y,\text{thres}}$. Figure 5.6 shows the results and confirms that the

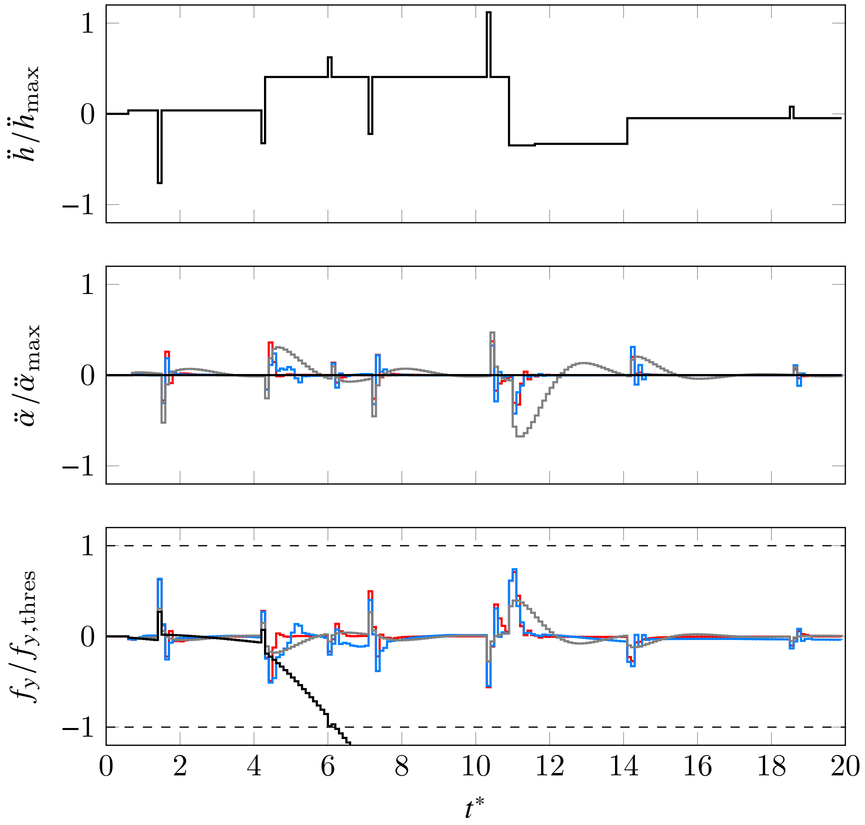


Figure 5.6: Values for the disturbance $\ddot{h}$, the control input $\ddot{\alpha}$, and the lift f_y during a simulated evaluation episode in the classical unsteady aerodynamics environment using no controller (—), a proportional controller (—), and the trained agent observing the two most recent timesteps with full wake info (—) and pressure info (—).

reinforcement learning policy adequately rejects the disturbances. The comparison with the proportional controller does not indicate that the reinforcement learning method is a better approach than linear control because the proportional controller only observes the lift error, and better controllers can be developed using the many powerful tools available from linear control theory.

Parameter	Description	Range
A	Amplitude	$[-1, 1]$
x_0	Spatial center x-coordinate relative to mid-chord point	$[-0.74c, -0.76c]$
y_0	Spatial center y-coordinate relative to mid-chord point	$[-0.1c, 0.1c]$
t_0^*	Temporal center	0.5
σ_x, σ_y	Spatial width	$[0.08c, 0.12c]$
σ_{t^*}	Temporal width	$[0.049, 0.051]$

Table 5.2: Parameters of the point forcing used in the viscous flow environment.

5.4 Training an agent in a viscous flow environment

Now we focus again on the regulation problem, aiming to minimize the lift variations caused by disturbances, but in a viscous flow environment. Note that the problem can be easily extended to follow a varying reference lift and to minimize the lift variations from, e.g., bluff-body vortex shedding. Here we look at a different type of disturbance than in the previous section: a transient vertical point force $\mathbf{F} = (0, F_y)$ introduced upstream of the airfoil with random parameters for each episode. Specifically, the disturbance is Gaussian in space and time and has the form

$$F_y(x, y, t) = \rho U^2 c \frac{A}{\pi^{3/2}\sigma_x \sigma_y \sigma_t} \exp\left(-\frac{(x - x_0)^2}{\sigma_x^2}\right) \exp\left(-\frac{(y - y_0)^2}{\sigma_y^2}\right) \exp\left(-\frac{(t^* - t_0^*)^2}{\sigma_{t^*}^2}\right) \tag{5.2}$$

where the parameters are uniformly sampled from the ranges shown in Table 5.2. Eldredge et al. [2021] shows that the aerodynamic response of the flat plate is nonlinearly dependent on the strength of this disturbance. The control input $\ddot{\alpha}$ is limited to $[-\ddot{\alpha}_{\max}, \ddot{\alpha}_{\max}]$ with $\ddot{\alpha}_{\max} = 10U^2/c^2$.

Each episode has one disturbance and lasts for three convective times. One agent-environment interaction occurs every three timesteps of the flow solver, and the control input is held constant between those interactions. We apply the same training algorithm as we used in the previous section, but for the evaluation, we only simulate one episode with parameters $A = 1$, $x_0 = -0.75c$, $y_0 = 0$, $t_0^* = 0.5$, $\sigma_x = \sigma_y = 0.1c$, and $\sigma_{t^*} = 0.05$. Furthermore, we initialize each episode with a steady-state flow field with fully developed boundary layers, obtained by running one simulation for five convective times with $\alpha = 0$ before the training.

We also modify the reward function from the previous section with the following additions: a negative reward for high variations $\ddot{\alpha}$ between two timesteps to prevent rapid oscillatory behavior, a negative reward when the lift variations exceed the lift threshold $f_{y,\text{thres}} = 0.6\rho c U^2$ to accelerate the learning, and a negative reward when $|\dot{\alpha}|$ exceeds its limit $\dot{\alpha}_{\max} = 2U/c$ to prevent numerical instabilities. In the latter two cases, the episode is terminated early. Furthermore, we scale the lift by the smaller $f_{y,\text{scale}} = 0.1\rho c U^2$ instead of its threshold, which also improves the training. Thus, the reward function is as follows,

$$R_{k+1} = 1 - \frac{|f_{y,k}|}{f_{y,\text{scale}}} - \frac{|\ddot{\alpha}_k - \ddot{\alpha}_{k-1}|}{\ddot{\alpha}_{\max}} + \begin{cases} -100, & \text{if } |\dot{\alpha}_k| > \dot{\alpha}_{\max} \\ 0, & \text{otherwise} \end{cases} + \begin{cases} -100, & |f_{y,k}| > f_{y,\text{thres}} \\ 0, & \text{otherwise} \end{cases}. \quad (5.3)$$

Because of the computational cost of the simulations, we limit the total training steps in this exploratory study to 100000. Furthermore, while it is possible to observe the entire vorticity field (and thus the state of the flow) and feed this to a convolutional layer in the policy network, the number of required training steps for adequate results with the TD3 algorithm would be too high. We limit ourselves, therefore, to the following three types of agents: one that observes no info about the wake, one that observes the pressure jump over the plate at three positions, and one that observes the pressure jump over the plate at seven positions. All the agents additionally observe α, $\dot{\alpha}$, and f_y, and they observe this information from the four most recent timesteps. These cases are summarized in table 5.3.

Agent type	Description	Observation vector for one timestep
No wake info	Only the lift and angular states are observed.	$[\alpha_k, \dot{\alpha}_k, f_{y,k-1}]^\top$
3 pressure sensors	The pressure jump at $\tilde{x}/c = -0.3, 0, 0.3$ is observed in addition to the lift and angular states.	$[\alpha_k, \dot{\alpha}_k, f_{y,k-1}, p_{1,k-1}, p_{2,k-1}, p_{3,k-1}]^\top$
7 pressure sensors	The pressure jump at $\tilde{x}/c = -0.3, -0.2, -0.1, 0, 0.1, 0.2, 0.3$ is observed in addition to the lift and angular states.	$[\alpha_k, \dot{\alpha}_k, f_{y,k-1}, p_{1,k-1}, ..., p_{7,k-1}]^\top$

Table 5.3: Description and observation vectors for the three types of agents trained in the viscous flow environment.

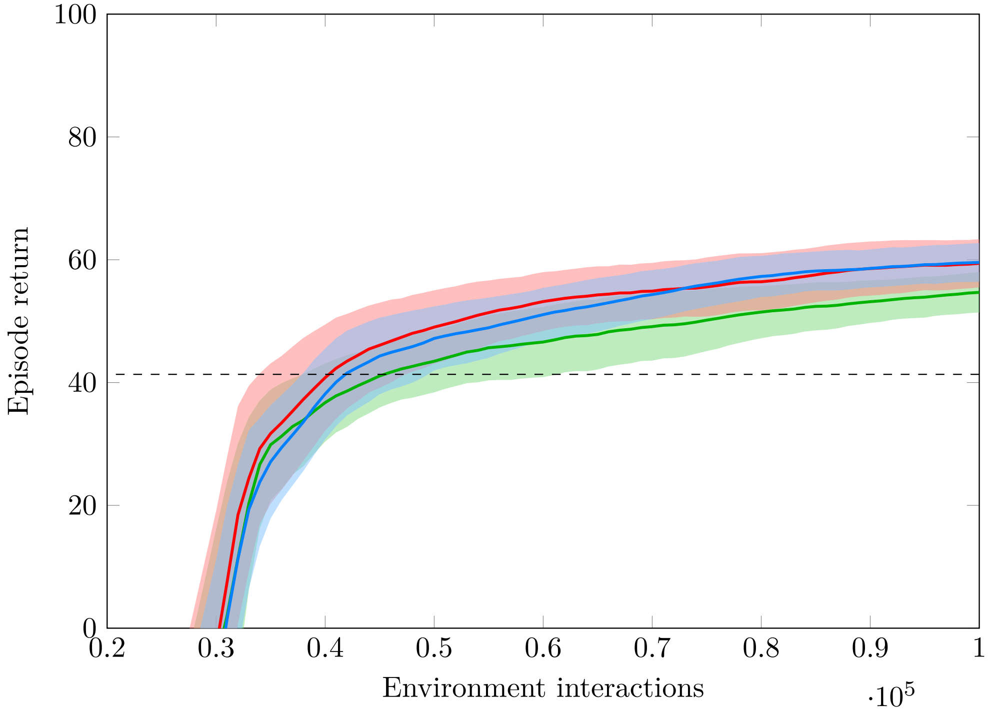


Figure 5.7: Mean (solid line) and one standard deviation (shaded) of the average evaluation episode return for the three types of agents in the viscous flow environment observing the four most recent timesteps: no wake info (——), pressure from three sensors (——), and pressure from seven sensors (——). Overlaid is the return for the same episode when no control is applied (- - -).

Figure 5.7 shows the training curves for the three cases using a moving average window of 20 evaluation episodes. We can make several observations. Firstly, the policy rapidly learns to keep the lift and angular velocity within their limits and avoid the large negative rewards. Secondly, the training has not converged yet. Nevertheless, the performance of the learned policies easily exceeds the no-control case with a static airfoil. Thirdly, the agents that observe pressure outperform the agent that doesn't. Finally, the training does not yet show a clear advantage of using seven pressure sensors instead of three for this case.

We can compare the performance of the trained policies by observing the flow fields, shown in Figure 5.8, and the trajectory of the lift, shown in Figure 5.9. Because the two cases that observe pressure perform similarly, we only show an evaluation episode simulated with an agent using seven pressure sensors and compare it to when no control is applied. The point force in the evaluation episode is upward, and the generated vortex dipole travels over the top side of the airfoil. Note that the point force is not divergence-free and will induce an instantaneous pressure on the plate while active [Eldredge et al., 2021]. This effect is visible as a drop in lift in both cases and is hard for the controller to counteract. The upward point force generates a vortex dipole that propels itself upward and is convected downstream by the relative freestream flow. Therefore, the vortex dipole and the associated low-pressure region travel over the top side of the airfoil and exert an upward suction on the plate while also drawing in more vorticity from the sharp leading edge. The suction manifests itself as a positive lift force when no control is applied. The trained reinforcement learning agent responds by decreasing its angle of attack to about minus nine degrees while the dipole is reaching the leading edge, reducing the interaction of the dipole with the leading edge. The combination of the added-mass effect from the angular velocity and the weaker vortex interactions greatly reduces the lift peak that appears in the no-control case, which indicates a successful, although not converged, training of the agent.

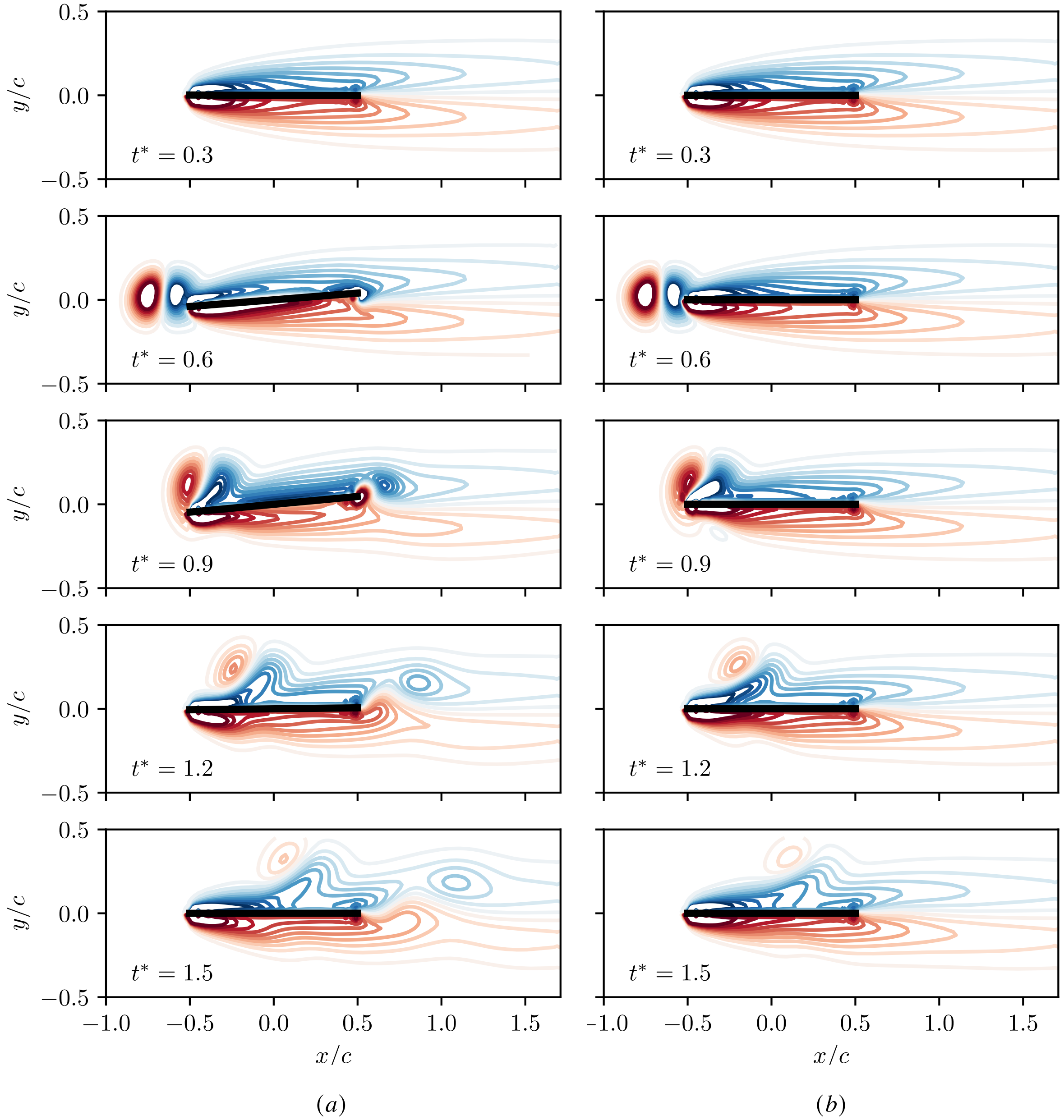


Figure 5.8: Snapshots of the flow field showing the vorticity contours during a simulated evaluation episode in the viscous flow environment using the trained agent observing the four most recent timesteps with pressure info from seven sensors (*a*) and no controller (*b*).

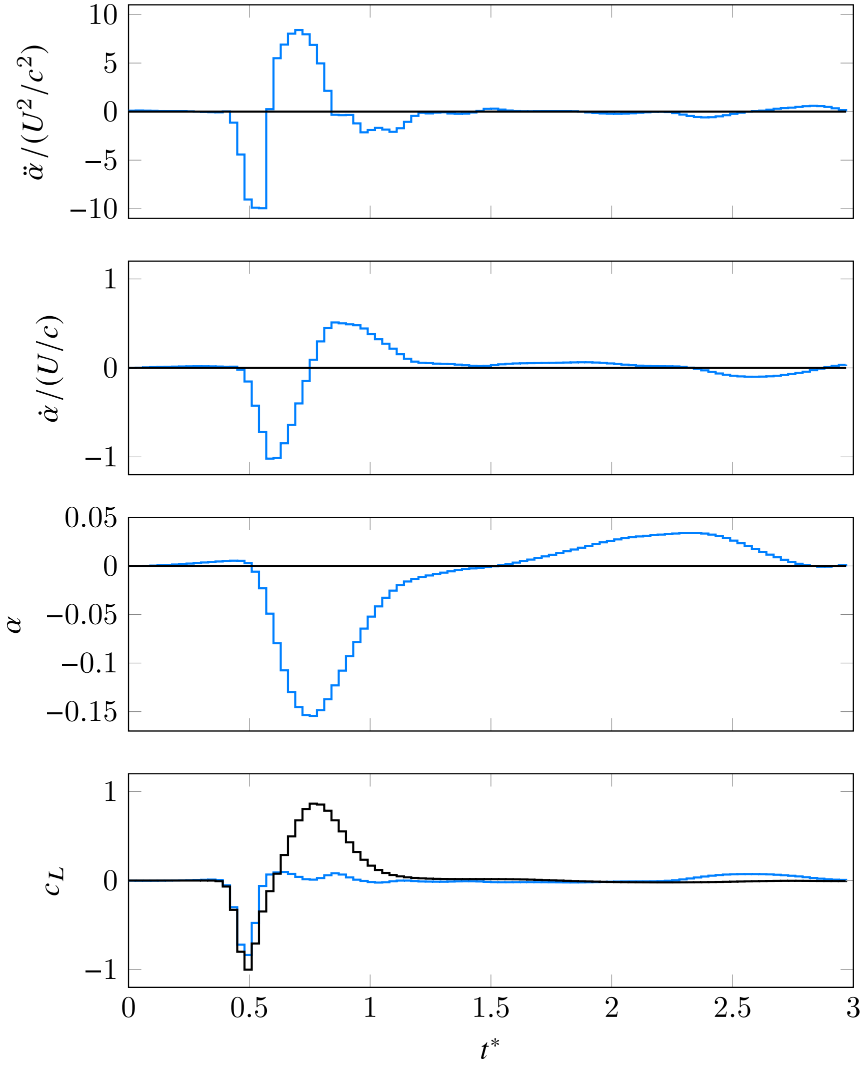


Figure 5.9: Values for the control input $\ddot{\alpha}$, the angular velocity $\dot{\alpha}$, the angle of attack α, and the lift f_y during a simulated evaluation episode in the viscous flow environment using no controller (——) and the trained agent observing the four most recent timesteps with pressure info from seven sensors (——).

CHAPTER 6

Conclusion

The modeling and control of unsteady aerodynamics is often complicated by the high dimensionality and nonlinearity of the flow. Specifically, the high dimensionality greatly increases the cost to model or analyze the flow, while the nonlinearity increases the complexity of the flows and reduces the effectiveness of the abundant linear tools to perform flow control. Recent advancements in data-driven techniques and machine learning provide a growing number of tools that deal with these challenges. Still, these tools require large amounts of accurate data to be available.

In this work, we investigated numerical methods that can accurately and efficiently model unsteady aerodynamic flows and explore the use of reinforcement learning to control such flows. We introduced different approaches for modeling incompressible flows with various levels of fidelity in Chapter 2: linear classical unsteady aerodynamics models that assume small perturbations and inviscid flow, inviscid vortex models that discretize the vorticity and can model nonlinear flow behavior using potential flows, and viscous flow models that can be efficiently computed in the body-fixed reference frame using the immersed boundary or immersed layers projection method.

In Chapter 3, we presented a treatment of potential flow on Cartesian grids for use in vortex models. This work is based on the computation of the discrete streamfunction through the streamfunction-vorticity Poisson equation with singular vortex elements as vorticity sources. In our potential flow treatment, we used two algebraic techniques that allowed us to mimic the analytical treatment of potential flows around sharp-edged bodies with bound vortex sheets. The first technique is to account for surfaces in the flow by using the immersed boundary projection method. We introduced a Lagrange multiplier for the no-penetration constraint in the streamfunction-vorticity Poisson equation and identified it as a discrete version of the continuous strength distribution of a bound vortex sheet on the body. The discrete equations that solve the associated saddle-point system are then completely analogous to the continuous boundary integral equations. The second algebraic technique is to decompose the discrete bound vortex sheet strength for sharp-edged bodies into a singular and non-singular part. One can then add constraints on the elements of the non-singular part that are located at the edges to make the recomposed bound vortex sheet well-behaved. This way, we enforced the Kutta condition in a way that is similar to analytical treatments of the Kutta condition and that allows for generalized edge conditions as well.

In Chapter 4, we investigated the effects of wind tunnel walls and an irrotational gust generator on the numerical simulations of airfoil gust responses, as these types of simulations commonly accompany wind tunnel experiments but often don't fully account for wind tunnel effects. To analyze these effects, we developed a computational framework to model a wind tunnel with an irrotational gust-generating mechanism. The framework couples a viscous flow solver to model the flow near the test subject and a potential flow solver to compute an irrotational and inviscid flow that cancels out the normal velocity of the viscous flow at the wind tunnel walls. The potential flow also serves to introduce the irrotational gust into the overall flow. We found that at zero angle of attack and low gust strength, viscous models with and without wind tunnel corrections predict the same response and flow field, indicating that the wind tunnel effects are negligible. However, when the angle of attack of the airfoil is increased or the gust strength increases, (1) nonlinear effects such as separation cause the predictions by the viscous models to differ from the linear Wagner model, and (2) the confinement effect of the wind tunnel walls becomes an important factor that cannot be ignored. We also demonstrated that the wind tunnel model can be used in a data-assimilation approach for, e.g., parameter estimation by inferring the gust strength from synthetic velocity measurements upstream of the airfoil. We computed the corresponding flow response of a flat plate and noted that the inclusion of the wind tunnel corrections causes the flow to separate, which is not the case for the flat plate in an equivalent time-varying uniform flow.

In Chapter 5, we explored the use of deep reinforcement learning to train an agent to regulate the lift on a flat plate through the control of the angular acceleration of the plate while encountering or experiencing disturbances. The agents' task is to minimize the lift variations, and they learn to do so by maximizing the sum of the rewards they receive during a finite-time episode. These rewards decrease if the absolute value of the lift increases. The learning was conducted using the TD3 algorithm in two types of environments with different disturbances. The first environment was a classical unsteady aerodynamics environment, and the disturbances were random vertical accelerations of the plate (or freestream). We compared three types of agents: one that observes all the states of the system, one that observes the angular states of the plate, the lift, and pressure, and one that only observes the angular states of the plate and the lift. The training showed that all agents learned to reject the disturbances but with different levels of success. As expected, the agent that doesn't observe any information about the wake states performs the poorest, and the agent that observes all the states performs the best. The agent that observes pressure performs almost as well as the latter. When the agents observe the same type of observations but for each of the two most recent timesteps, their performance drastically increases, and all agents perform equally well. The second environment was a viscous flow environment, and the disturbance was a single point-force pulse with random parameters applied upstream of the airfoil. We compared three agents: one that observes the angular states and the lift, one that observes additional information from three pressure sensors, and one from seven sensors. All agents observe their observations from the four most recent timesteps. The aerodynamic response of the flow to the point force exhibits strong nonlinear behavior, partly due to interactions with the leading edge, and the lift variations depend nonlinearly on the forcing parameters. Nevertheless, the reinforcement learning algorithm again successfully trained the agents to reject the disturbance by performing a pitching maneuver. The two agents that observed pressure performed better than the one agent that didn't. Interestingly, the agent that observed information from three pressure sensors performed as well as the agent that observed information from seven pressure sensors.

The tools and frameworks introduced in this work provide promising pathways for future research:

- The grid-based potential flow framework only considered the streamfunction and vortices in detail. One can easily extend this to compute the discrete scalar potential and account for sources and sinks in the flow. A more elaborate study could lead to a framework for computing the discrete scalar potential as the discrete conjugate harmonic of the discrete streamfunction and vice versa.

- One can apply the reinforcement learning framework to obtain many interesting flow control results with different types of airfoils, reward functions, and disturbances than the ones considered here. Interesting performance objectives that could affect the reward are, for example, the required power, drag, lift-to-drag ratio, etc.

- The policies obtained with the reinforcement learning framework can be compared to other types of control, such as model-predictive control, dynamic mode decomposition with control, balanced proper orthogonal decomposition with control, etc.

- The viscous flow environment is still very expensive to evaluate on modern computers and can be too slow for comprehensive reinforcement learning campaigns that include multiple types of disturbances. A promising alternative to the model-free, off-policy algorithms used in this work is model-based reinforcement learning, where the agent can access a cheaper model for fast, short-horizon trajectory predictions between environment interactions, resulting in an even higher sample efficiency.

APPENDIX A

Solution of general saddle-point systems

A general block system (with positive semi-definite matrix $\mathcal{A}$) can be decomposed as follows:

$$\begin{bmatrix} \mathcal{A} & \mathcal{B}_1^T \\ \mathcal{B}_2 & -\mathcal{C} \end{bmatrix} = \begin{bmatrix} \mathcal{A} & 0 \\ \mathcal{B}_2 & \mathcal{S} \end{bmatrix} \begin{bmatrix} \mathsf{I} & \mathcal{A}^{-1}\mathcal{B}_1^T \\ 0 & \mathsf{I} \end{bmatrix}, \tag{A.1}$$

where

$$\mathcal{S} \equiv -\mathcal{C} - \mathcal{B}_2\mathcal{A}^{-1}\mathcal{B}_1^T \tag{A.2}$$

is the *Schur complement* of the matrix system and I is the identity. By this decomposition, we can develop an algorithm for the solution of the block system

$$\begin{bmatrix} \mathcal{A} & \mathcal{B}_1^T \\ \mathcal{B}_2 & -\mathcal{C} \end{bmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{r_1} \\ \mathbf{r_2} \end{pmatrix}. \tag{A.3}$$

We will refer to $\mathbf{x}$ as the solution vector and $\mathbf{y}$ as the constraint force. We define the intermediate solution vector $(\mathbf{x}^*, \mathbf{y}^*)^T$ as the solution of the lower-triangular system

$$\begin{bmatrix} \mathcal{A} & 0 \\ \mathcal{B}_2 & \mathcal{S} \end{bmatrix} \begin{pmatrix} \mathbf{x}^* \\ \mathbf{y}^* \end{pmatrix} = \begin{pmatrix} \mathbf{r_1} \\ \mathbf{r_2} \end{pmatrix} \tag{A.4}$$

and then the solution we seek can be found by back substitution of

$$\begin{bmatrix} \mathsf{I} & \mathcal{A}^{-1}\mathcal{B}_1^T \\ 0 & \mathsf{I} \end{bmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{x}^* \\ \mathbf{y}^* \end{pmatrix} \tag{A.5}$$

The algorithm we derive from this is

$$\begin{aligned}
\mathcal{A}\mathbf{x}^* &= \mathbf{r_1}, \\
\mathcal{S}\mathbf{y}^* &= \mathbf{r_2} - \mathcal{B}_2\mathbf{x}^*, \\
\mathbf{y} &= \mathbf{y}^*, \\
\mathbf{x} &= \mathbf{x}^* - \mathcal{A}^{-1}\mathcal{B}_1^T\mathbf{y}.
\end{aligned} \tag{A.6}$$

It is also useful to have an inverse representation of the block matrix system:

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{bmatrix} \mathcal{A}^{-1} + \mathcal{A}^{-1}\mathcal{B}_1^T\mathcal{S}^{-1}\mathcal{B}_2\mathcal{A}^{-1} & -\mathcal{A}^{-1}\mathcal{B}_1^T\mathcal{S}^{-1} \\ -\mathcal{S}^{-1}\mathcal{B}_2\mathcal{A}^{-1} & \mathcal{S}^{-1} \end{bmatrix} \begin{pmatrix} \mathbf{r_1} \\ \mathbf{r_2} \end{pmatrix}. \tag{A.7}$$

APPENDIX B

Some geometric relations for discrete surfaces

Consider a closed surface $\mathcal{S}_b$ with unit normal $\boldsymbol{n}$. We will recall some basic geometric relations here, and then provide some discrete versions of these relations based on the set of points with coordinates r_x, r_y, normal components n_x and n_y (which, the reader will recall, contain the surface length or area of each segment or panel associated with the points).

The volume $\mathcal{V}_b$ of the region enclosed by $\mathcal{S}_b$ can be computed from the integral

$$\mathcal{V}_b = \frac{1}{n_d} \int_{\mathcal{S}_b} \boldsymbol{x} \cdot \boldsymbol{n} \, \mathrm{d}S, \tag{B.1}$$

where n_d is the number of spatial dimensions (2 or 3). Using the notation above, the approximate form of this expression is

$$\mathcal{V}_b \approx \frac{1}{n_d} \sum_j \mathsf{r}_j^T \mathsf{n}_j, \tag{B.2}$$

where the sum is taken over the n_d components.

An alternative formula for the volume is

$$\mathcal{V}_b \boldsymbol{e}_j = -\frac{1}{n_d - 1} \int_{\mathcal{S}_b} \boldsymbol{x} \times (\boldsymbol{n} \times \boldsymbol{e}_j) \, \mathrm{d}S. \tag{B.3}$$

The components of this integral can be written discretely as

$$\mathcal{V}_b \boldsymbol{e}_j \approx \frac{\boldsymbol{e}_j}{n_d - 1} \sum_{k \neq j} \mathsf{r}_k^T \mathsf{n}_k \tag{B.4}$$

And finally, a third alternative is

$$\mathcal{V}_b \boldsymbol{I} = \int_{\mathcal{S}_b} \boldsymbol{x} \boldsymbol{n} \, \mathrm{d}S, \tag{B.5}$$

where $\boldsymbol{I}$ is the identity. The discrete form of this is a diagonal matrix with $\mathsf{r}_x^T \mathsf{n}_x$, $\mathsf{r}_y^T \mathsf{n}_y$, and $\mathsf{r}_z^T \mathsf{n}_z$ along the diagonal.

Thus, we can conclude that the volume of the body is approximately

$$\mathcal{V}_b \approx \langle \mathsf{r}_x, \mathsf{n}_x \rangle_{\mathcal{S}^{N_s}} \approx \langle \mathsf{r}_y, \mathsf{n}_y \rangle_{\mathcal{S}^{N_s}} \approx \langle \mathsf{r}_z, \mathsf{n}_z \rangle_{\mathcal{S}^{N_s}}, \tag{B.6}$$

or any average of some combination of these.

The centroid of the body can be derived from the equation

$$\boldsymbol{X}_c \mathcal{V}_b = \frac{1}{2} \int_{\mathcal{S}_b} \boldsymbol{x} \cdot \boldsymbol{x} \boldsymbol{n} \, \mathrm{d}S, \tag{B.7}$$

or, in discrete form,

$$X_c \approx \frac{1}{2\mathcal{V}_b} \langle \mathsf{D}_{\mathsf{r}_x} \mathsf{r}_x + \mathsf{D}_{\mathsf{r}_y} \mathsf{r}_y, \mathsf{n}_x \rangle_{\mathcal{S}^{N_s}}, \qquad Y_c \approx \frac{1}{2\mathcal{V}_b} \langle \mathsf{D}_{\mathsf{r}_x} \mathsf{r}_x + \mathsf{D}_{\mathsf{r}_y} \mathsf{r}_y, \mathsf{n}_y \rangle_{\mathcal{S}^{N_s}}. \tag{B.8}$$

APPENDIX C

Data assimilation with experiments

The wind tunnel model in 4 contains uncertainties that arise from simplifications and approximations. This includes the simplifications mentioned in that chapter, but also, unknown inlet, outlet, and suction boundary conditions, or the geometry and turbulence modeling. On the other hand, experiments observe the real flow but with measurements that are often sparse and noisy. Data assimilation can combine experimental measurements with a dynamical model to improve the accuracy of either. Experiments can improve state or parameter estimates in a dynamical model West and Harrison [1997] the dynamical model can help to filter, smooth, or forecast measurements Durbin and Koopman [2012].

The computational wind tunnel model can serve as the dynamical model for a data assimilation procedure. For example, it can predict the flow field for the next time step based on an estimate of the flow at the current time step. Measurements from pressure, hot-wire, or force balance sensors from the next time step can be observed in the actual wind tunnel to correct the a priori estimate from the dynamical model such that its error with respect to those measurements is minimized. Because the numerical model is high dimensional and non-linear, appropriate estimation techniques such as the ensemble Kalman filter should be considered da Silva and Colonius [2018].

In the examples in this work, we demonstrate data assimilation using the simpler example of inferring a single parameter of the wind tunnel model from experiments. Suppose we have vertical velocity measurements z_k available at discrete times t_k, with $k = 1, \dots, N$, from a single crosswire probe positioned at $\boldsymbol{x}_{\text{probe}}$, upstream of the airfoil, such as shown in Figure 4.1. The measurements are disturbed by measurement noise ν_k, which we will assume is normally distributed with constant variance σ_ν^2 and zero mean. We wish to model the experiments with the computational wind tunnel model to obtain more information about the flow, but we need to know what the suction velocity was to match the experiments. The suction velocity was not directly measured but can be estimated through the observation model

$$\begin{aligned} z_k &= h(\theta_k, t_k) + \nu_k \\ &= v_{\boldsymbol{\omega}}(\boldsymbol{x}_{\text{probe}}, \theta, t_k) + v_\phi(\boldsymbol{x}_{\text{probe}}, \theta, t_k) + V_\infty + \nu_k, \end{aligned} \tag{C.1}$$

where we consider $\theta = V_{\text{s}}$ a parameter (or state) of the system. Notice that, in this case, the measurements depend non-linearly on the state. However, for simplicity, we will make the assumption that, if the probe is far enough upstream of the test subject, the influence of the test subject on the flow at $\boldsymbol{x}_{\text{probe}}$ is negligible. Furthermore, by construction in our model, the vertical component of the uniform flow V_∞ is zero at all times. If we assume $v_{\boldsymbol{\omega}}(\boldsymbol{x}_{\text{probe}})$ and $v_{\phi,\text{c}}(\boldsymbol{x}_{\text{probe}})$ are zero, then we obtain an observation function that is linear[1] in θ:

$$\begin{aligned} z_k &= v_{\phi,\text{s}}(\boldsymbol{x}_{\text{probe}}, \theta, t_k) + \nu_k \\ &= H\theta_k + \nu_k \end{aligned} \tag{C.2}$$

where H is called the observation matrix, which in this case is just a constant scalar, representing the vertical velocity component of the suction flow $\boldsymbol{u}_{\phi,\text{s}}$ evaluated at the probe when the suction velocity is set to unity. Note that H is time invariant in this case and can be computed a priori.

To accommodate a non-stationary suction velocity, we provide an evolution (or dynamical) model for θ_k. It is possible that the dynamics of the suction flow are known, for example through knowledge of its opening and closing step response (see, e.g., He et al. [2022]). In

[1] Note that $v_{\boldsymbol{\omega}}(\boldsymbol{x}_{\text{probe}}) = 0$ and $v_{\phi,\text{c}}(\boldsymbol{x}_{\text{probe}}) = 0$ are not necessary assumptions to obtain a linear observation equation. It is sufficient to assume that they are insensitive to θ or that $v_{\boldsymbol{\omega}}(\boldsymbol{x}_{\text{probe}})$ depends only linearly on θ and that $v_{\phi,\text{c}}(\boldsymbol{x}_{\text{probe}})$ does not depend non-linearly on θ through its coupling with $\boldsymbol{u}_{\boldsymbol{\omega}}$.

that case, one can predict the suction velocity based on the user commands for the louvers. For conciseness, we won't try to model the actual dynamics here. Instead, we model the evolution of θ as a purely stochastic process, an approach that would be appropriate if we expect θ to vary slowly. Specifically, we use a Gaussian random walk

$$\theta_{k+1} = \theta_k + \omega_k, \tag{C.3}$$

where ω_k is the step size (or process noise) that is normally distributed with constant variance σ_ω^2 and zero mean. Overall, equations (C.2) and (C.3) comprise a simple linear state estimation problem and can easily be solved using linear state space methods. However, if we had used a more thorough wind tunnel model, e.g., to account for bias errors due to grid resolution errors, then the resulting non-linear estimation could be treated with ensemble methods, e.g., Mons et al. Mons et al. [2016] and da Silva & Colonius da Silva and Colonius [2018], in which the state vector would comprise the discretized viscous velocity field. Nevertheless, the current formulation is more cost-efficient, because the expensive Navier-Stokes solution is not in the estimation loop, and the framework can be trivially added to a flow solver.

If we assume that θ_0, ν_k, and ω_k are all mutually independent, the linear estimation (or filtering) problem (C.2) and (C.3) can be straightforwardly solved for θ at every timestep t_k using Kalman filtering (or, equivalently, recursive Bayesian estimation). For this one-dimensional random walk model, the Kalman filter equations that give the filtered estimate $\hat{\theta}_{k|k}$ and the one-step-ahead prediction $\hat{\theta}_{k+1}$ for θ_k, with associated variances $P_{k+1|k+1}$ and P_k, to update a known $\hat{\theta}_k$ and P_k from t_k to t_{k+1} are

$$\begin{aligned} \hat{\theta}_{k|k} &= \hat{\theta}_k + K_k(z_k - H\hat{\theta}_k), & P_{k|k} &= P_k(1 - K_k H), \\ \hat{\theta}_{k+1} &= \hat{\theta}_{k|k}, & P_{k+1} &= P_{k|k} + \sigma_\omega^2, \end{aligned} \tag{C.4}$$

where $K_k = HP_k/(H^2\sigma_\omega^2 + \sigma_\nu^2)$.

Bibliography

S. Ahuja and C. W. Rowley. Feedback control of unstable steady states of flow past a flat plate using reduced-order estimators. *Journal of fluid mechanics*, 645:447–478, 2010.

C. Badrya, J. D. Baeder, and A. R. Jones. Application of prescribed velocity methods to a large-amplitude flat-plate gust encounter. *AIAA Journal*, 57(8):3261–3273, 2019. doi: 10.2514/1.J057978.

C. Badrya, H. Biler, A. R. Jones, and J. D. Baeder. Effect of gust width on flat-plate response in large transverse gust. *AIAA Journal*, 59(1):49–64, 2021. doi: 10.2514/1.J059678.

G. R. Baker. The "cloud in cell" technique applied to the roll up of vortex sheets. *Journal of Computational Physics*, 31(1):76–95, apr 1979. ISSN 00219991. doi: 10.1016/0021-9991(79)90063-9. URL `https://linkinghub.elsevier.com/retrieve/pii/0021999179900639`.

D. Beckers and J. D. Eldredge. JuliaIBPM/GridPotentialFlow.jl v0.3.2. Available at: `https://github.com/JuliaIBPM/GridPotentialFlow.jl/tree/v0.3.2`, June 2021.

D. Beckers and J. D. Eldredge. Planar potential flow on Cartesian grids. *Journal of Fluid Mechanics*, 941:A19, 2022. doi: 10.1017/jfm.2022.238.

M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 1:1–137, 2005.

K. Bieker, S. Peitz, S. L. Brunton, J. N. Kutz, and M. Dellnitz. Deep model predictive flow control with limited sensor data and online learning. *Theoretical and computational fluid dynamics*, 34:577–591, 2020.

S. L. Brunton and C. W. Rowley. Empirical state-space representations for theodorsen's lift model. *Journal of Fluids and Structures*, 38:174–186, 2013. ISSN 0889-9746. URL `https://doi.org/10.1016/j.jfluidstructs.2012.10.005`.

Steven L. Brunton and J. Nathan Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2 edition, 2022. doi: 10.1017/9781009089517.

P. Chatelain, A. Curioni, M. Bergdorf, D. Rossinelli, W. Andreoni, and P. Koumoutsakos. Billion vortex particle direct numerical simulations of aircraft wakes. *Computer Methods in Applied Mechanics and Engineering*, 197(13-16):1296–1304, feb 2008. ISSN 00457825. doi: 10.1016/j.cma.2007.11.016. URL `https://linkinghub.elsevier.com/retrieve/pii/S0045782507004574`.

R. Chatelin and P. Poncet. Hybrid grid-particle methods and Penalization: A Sherman-Morrison-Woodbury approach to compute 3D viscous flows using FFT. *Journal of Computational Physics*, 269:314–328, 2014. ISSN 10902716. doi: 10.1016/j.jcp.2014.03.023. URL `http://dx.doi.org/10.1016/j.jcp.2014.03.023`.

S. S. Chen. Vibration of nuclear fuel bundles. *Nuclear Engineering and Design*, 35(3): 399–422, dec 1975. ISSN 00295493. doi: 10.1016/0029-5493(75)90071-0. URL `https://linkinghub.elsevier.com/retrieve/pii/0029549375900710`.

A. J. Chorin and P. S. Bernard. Discretization of a vortex sheet, with an example of roll-up. *Journal of Computational Physics*, 13(3):423–429, nov 1973. ISSN 00219991. doi: 10.1016/0021-9991(73)90045-4. URL `https://linkinghub.elsevier.com/retrieve/pii/0021999173900454`.

J. P. Christiansen. Numerical simulation of hydrodynamics by the method of point vortices. *Journal of Computational Physics*, 13(3):363–379, nov 1973. ISSN 00219991. doi: 10.1016/0021-9991(73)90042-9. URL `https://linkinghub.elsevier.com/retrieve/pii/0021999173900429`.

T. Colonius and K. Taira. A fast immersed boundary method using a nullspace approach and multi-domain far-field boundary conditions. *Computer Methods in Applied Mechanics and Engineering*, 197(25-28):2131–2146, 2008. ISSN 00457825. doi: 10.1016/j.cma.2007.08.014.

M. Coquerelle and G.-H. Cottet. A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies. *Journal of Computational Physics*, 227 (21):9121–9137, 2008. ISSN 10902716. doi: 10.1016/j.jcp.2008.03.041.

G.-H. Cottet and P. Koumoutsakos. *Vortex Methods: Theory and Practice.* Cambridge University Press, Cambridge, UK, mar 2000. ISBN 0521621860. URL `https://www.cse-lab.ethz.ch/wp-content/papercite-data/pdf/cottet2000b.pdf`.

G.-H. Cottet and P. Poncet. Advances in direct numerical simulations of 3D wall-bounded flows by Vortex-in-Cell methods. *Journal of Computational Physics*, 193(1):136–158, jan 2004. ISSN 00219991. doi: 10.1016/j.jcp.2003.08.025. URL `https://linkinghub.elsevier.com/retrieve/pii/S0021999103004248`.

B. Couët, O. Buneman, and A. Leonard. Simulation of three-dimensional incompressible flows with a vortex-in-cell method. *Journal of Computational Physics*, 39(2):305–328, feb 1981. ISSN 00219991. doi: 10.1016/0021-9991(81)90154-6. URL `https://linkinghub.elsevier.com/retrieve/pii/0021999181901546`.

J. Cserti. Application of the lattice green's function for calculating the resistance of an infinite network of resistors. *Am. J. Phys*, 68:896–906, 2000.

A. F. C. da Silva and T. Colonius. Ensemble-based state estimator for aerodynamic flows. *AIAA Journal*, 56(7):2568–2578, 2018. doi: 10.2514/1.J056743.

D. Darakananda and J. D. Eldredge. A versatile taxonomy of low-dimensional vortex models for unsteady aerodynamics. *J. Fluid Mech.*, 858:917–948, 2019. doi: 10.1017/jfm.2018.792. URL `https://drive.google.com/open?id=15z7f2cRW4Dk7gMHwT80Kjh2FxnU7tieq`.

J. Durbin and S. J. Koopman. *Time Series Analysis by State Space Methods.* Oxford University Press, Oxford, 05 2012. ISBN 9780199641178. doi: 10.1093/acprof:oso/9780199641178.001.0001.

A. B. Ebiana and R. W. Bartholomew. Design considerations for numerical filters used in Vortex-in-cell algorithms. *Computers & Fluids*, 25(1):61–75, jan 1996. ISSN 00457930. doi: 10.1016/0045-7930(95)00031-3. URL `https://linkinghub.elsevier.com/retrieve/pii/0045793095000313`.

J. D. Eldredge. *Mathematical Modeling of Unsteady Inviscid Flows*, volume 50 of *Interdisciplinary Applied Mathematics.* Springer, 2019. ISBN 978-3-030-18318-9. doi: 10.1007/978-3-030-18319-6.

J. D. Eldredge. A method of immersed layers on cartesian grids, with application to incompressible flows. *Journal of Computational Physics*, 448:110716, 2022.

J. D. Eldredge. JuliaIBPM/ViscousFlow.jl v0.6.0, June 2023. URL `https://doi.org/10.5281/zenodo.8098488`.

J. D. Eldredge and A. R. Jones. Leading-edge vortices: Mechanics and modeling. *Annual Review of Fluid Mechanics*, 51(1):75–104, 2019. doi: 10.1146/annurev-fluid-010518-040334. URL `https://doi.org/10.1146/annurev-fluid-010518-040334`.

J. D. Eldredge, M. Le Provost, R. Baptista, and Y. Marzouk. Applications of ensemble kalman filtered vortex modeling to gust–wing interactions. In *AIAA Scitech 2021 Forum*, page 1936, 2021.

S. Fujimoto, H. van Hoof, and D. Meger. Addressing function approximation error in actor-critic methods. *CoRR*, abs/1802.09477, 2018. URL `http://arxiv.org/abs/1802.09477`.

M. Gad-el Hak. *Flow Control: Passive, Active, and Reactive Flow Management.* Cambridge University Press, 2000. doi: 10.1017/CBO9780511529535.

I. E. Garrick. On some reciprocal relations in the theory of nonstationary flows. Technical Report NACA-TN-629, NACA, 1938.

M. Gazzola, P. Chatelain, W. M. van Rees, and P. Koumoutsakos. Simulations of single and multiple swimmers with non-divergence free deforming geometries. *Journal of Computational Physics*, 230(19):7093–7114, 2011. ISSN 10902716. doi: 10.1016/j.jcp.2011.04.025. URL http://dx.doi.org/10.1016/j.jcp.2011.04.025.

T. Gillis, G. Winckelmans, and P. Chatelain. An efficient iterative penalization method using recycled Krylov subspaces and its application to impulsively started flows. *Journal of Computational Physics*, 347:490–505, oct 2017. ISSN 00219991. doi: 10.1016/j.jcp.2017.07.015. URL http://dx.doi.org/10.1016/j.jcp.2017.07.015https://linkinghub.elsevier.com/retrieve/pii/S002199911730517X.

T. Gillis, G. Winckelmans, and P. Chatelain. Fast immersed interface Poisson solver for 3D unbounded problems around arbitrary geometries. *Journal of Computational Physics*, 354:403–416, feb 2018. ISSN 00219991. doi: 10.1016/j.jcp.2017.10.042. URL https://linkinghub.elsevier.com/retrieve/pii/S0021999117308094.

T. Gillis, Y. Marichal, G. Winckelmans, and P. Chatelain. A 2D immersed interface Vortex Particle-Mesh method. *Journal of Computational Physics*, 394:700–718, 2019. ISSN 10902716. doi: 10.1016/j.jcp.2019.05.033. URL https://doi.org/10.1016/j.jcp.2019.05.033.

A. Goza, S. Liska, B. Morley, and T. Colonius. Accurate computation of surface stresses and forces with immersed boundary methods. *Journal of Computational Physics*, 321: 860–873, 2016. ISSN 10902716. doi: 10.1016/j.jcp.2016.06.014. URL http://dx.doi.org/10.1016/j.jcp.2016.06.014.

A. L. Grubb, A. Moushegian, D. J. Heathcote, and M. J. Smith. Physics and computational modeling of nonlinear transverse gust encounters. *AIAA Scitech 2020 Forum*, 2020. doi: 10.2514/6.2020-0080.

T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018.

X. He and D. R. Williams. Unsteady aerodynamic loads on an airfoil at high angle of attack in a randomly surging flow. In *AIAA Scitech 2020 Forum*, page 0557, 2020. doi: https://doi.org/10.2514/6.2020-0557.

X. He, D. R. Williams, and Scott T. M. Dawson. Transverse gust generation in a wind tunnel: a suction-driven approach. *Experiments in Fluids*, 63(8):125, August 2022. doi: 10.1007/s00348-022-03484-9.

M. M. Hejlesen, P. Koumoutsakos, A. Leonard, and J. H. Walther. Iterative Brinkman penalization for remeshed vortex methods. *Journal of Computational Physics*, 280:547–562, 2015. ISSN 10902716. doi: 10.1016/j.jcp.2014.09.029. URL `http://dx.doi.org/10.1016/j.jcp.2014.09.029`.

Marvin A. Jones. The separated flow of an inviscid fluid around a moving flat plate. *Journal of Fluid Mechanics*, 496(496):405–441, 2003. ISSN 00221120. doi: 10.1017/S0022112003006645.

R. T. Jones. Operational treatment of the non-uniform lift theory in airplane dynamics. Technical Report NACA-TN-667, NACA, 1938.

S. Katsura and S. Inawashiro. Lattice Green's functions for the rectangular and the square lattices at arbitrary points. *J. Math. Phys.*, 12:1622–1630, 1971.

R.J. LeVeque and Z. Li. The Immersed Interface Method for Elliptic Equations with Discontinuous Coefficients and Singular Sources. *SIAM Journal on Numerical Analy-*

sis, 31(4):1019–1044, aug 1994. ISSN 0036-1429. doi: 10.1137/0731054. URL `http://epubs.siam.org/doi/10.1137/0731054`.

S. Liska and T. Colonius. A parallel fast multipole method for elliptic difference equations. *J. Comput. Phys.*, 278:76–91, 2014a.

S. Liska and T. Colonius. A parallel fast multipole method for elliptic difference equations. *Journal of Computational Physics*, 278:76–91, 2014b. doi: https://doi.org/10.1016/j.jcp.2014.07.048.

S. Liska and T. Colonius. A fast immersed boundary method for external incompressible viscous flows using lattice Green's functions. *Journal of Computational Physics*, 331:257–279, 2016. doi: https://doi.org/10.1016/j.jcp.2016.11.034.

M. R. Malik and D. M. Bushnell. Role of computational fluid dynamics and wind tunnels in aeronautics R&D. Technical Report 2012-217602, NASA, 2012.

Y. Marichal, P. Chatelain, and G. Winckelmans. An immersed interface solver for the 2-D unbounded Poisson equation and its application to potential flow. *Computers and Fluids*, 96:76–86, 2014. ISSN 00457930. doi: 10.1016/j.compfluid.2014.03.012. URL `http://dx.doi.org/10.1016/j.compfluid.2014.03.012`.

J. C. S. Meng and J. A. L. Thomson. Numerical studies of some nonlinear hydrodynamic problems by discrete vortex element methods. *Journal of Fluid Mechanics*, 84(3):433–453, feb 1978. ISSN 0022-1120. doi: 10.1017/S0022112078000269. URL `https://www.cambridge.org/core/product/identifier/S0022112078000269/type/journal{_}article`.

V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and A. Riedmiller. Playing atari with deep reinforcement learning, 2013.

J. J. Monaghan. Extrapolating B splines for interpolation. *Journal of Computational Physics*, 60(2):253–262, sep 1985. ISSN 00219991. doi: 10.1016/0021-9991(85)90006-3. URL https://linkinghub.elsevier.com/retrieve/pii/0021999185900063.

V. Mons, J.-C. Chassaing, T. Gomez, and P. Sagaut. Reconstruction of unsteady viscous flows using data assimilation schemes. *Journal of Computational Physics*, 316:255–280, 2016. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2016.04.022.

M. Moriche, A. Gonzalo, O. Flores, and M. García-Villalba. Three-dimensional effects on plunging airfoils at low reynolds numbers. *AIAA Journal*, 59(1):65–74, 2021. doi: 10.2514/1.J058569.

A. Moushegian and M. J. Smith. Analysis of a wing moving through a nonlinear gust. *AIAA Scitech 2019 Forum*, 2019. doi: 10.2514/6.2019-0637. URL https://arc.aiaa.org/doi/abs/10.2514/6.2019-0637.

S. Neumark. Pressure distribution on an airfoil in nonuniform motion. *Journal of the Aeronautical Sciences*, 19(3):214–215, 1952. URL https://doi.org/10.2514/8.2219.

J. B. Perot. An analysis of the fractional step method. *Journal of Computational Physics*, 108(1):51–58, 1993.

C. S. Peskin. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10(2):252–271, 1972. ISSN 0021-9991. doi: https://doi.org/10.1016/0021-9991(72)90065-4. URL https://www.sciencedirect.com/science/article/pii/0021999172900654.

P. Poncet. Analysis of an immersed boundary method for three-dimensional flows in vorticity formulation. *Journal of Computational Physics*, 228(19):7268–7288, 2009. ISSN 10902716. doi: 10.1016/j.jcp.2009.06.023. URL http://dx.doi.org/10.1016/j.jcp.2009.06.023.

K. Ramesh, A. Gopalarathnam, K. Granlund, M. V. OL, and J. R. Edwards. Discrete-vortex method with novel shedding criterion for unsteady aerofoil flows with intermittent leading-edge vortex shedding. *J. Fluid Mech.*, 751:500–548, 2014.

J. T. Rasmussen, G.-H. Cottet, and J. H. Walther. A multiresolution remeshed Vortex-In-Cell algorithm using patches. *Journal of Computational Physics*, 230(17):6742–6755, 2011. ISSN 10902716. doi: 10.1016/j.jcp.2011.05.006.

K. J. Åström and R. M. Murray. *Feedback systems : an introduction for scientists and engineers*. Princeton University Press, 2021. ISBN 9780691193984. URL `http://fbsbook.org`.

M. R. Rennie, B. Catron, D. R. Williams, and M. Z. Feroz. Mathematical modeling of wind tunnels for low reynolds number unsteady aerodynamic testing. In *55th AIAA Aerospace Sciences Meeting*, page 1532, jan 2017. doi: 10.2514/6.2017-1532.

D. Rossinelli, M. Bergdorf, G.-H. Cottet, and P. Koumoutsakos. GPU accelerated simulations of bluff body flows using vortex particle methods. *Journal of Computational Physics*, 229(9):3316–3333, 2010. ISSN 10902716. doi: 10.1016/j.jcp.2010.01.004. URL `http://dx.doi.org/10.1016/j.jcp.2010.01.004`.

P. G. Saffman. *Vortex Dynamics*, volume 87 of *Cambridge Monographs on Mechanics*. Cambridge University Press, jan 1993. ISBN 9780521477390. doi: 10.1017/CBO9780511624063. URL `https://www.cambridge.org/core/product/identifier/9780511624063/type/book`.

S. Skogestad and I. Postlethwaite. *Multivariable feedback control: Analysis and Design*. John Wiley, Hoboken, US-NJ, 2005.

H. J. Spietz, M. M. Hejlesen, and J. H. Walther. Iterative Brinkman penalization for simulation of impulsively started flow past a sphere and a circular disc. *Journal of Computa-*

tional Physics, 336:261–274, 2017. ISSN 10902716. doi: 10.1016/j.jcp.2017.01.064. URL `http://dx.doi.org/10.1016/j.jcp.2017.01.064`.

K. Taira and T. Colonius. The immersed boundary method: A projection approach. *Journal of Computational Physics*, 225(2):2118–2137, 2007. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2007.03.005. URL `https://www.sciencedirect.com/science/article/pii/S0021999107001234`.

T. Theodorsen. General theory of aerodynamic instability and the mechanism of flutter. Technical Report NACA-TN-496, NACA, 1935.

B.G. van der Wall and J. G. Leishman. On the influence of timevarying flow velocity on unsteady aerodynamics. *Journal of the American Helicopter Society*, 39(4):25–36, 1994. doi: doi:10.4050/JAHS.39.25.

C. Vignon, J. Rabault, and R. Vinuesa. Recent advances in applying deep reinforcement learning for flow control: Perspectives and future directions. *Physics of Fluids*, 35(3): 031301, 03 2023. ISSN 1070-6631. doi: 10.1063/5.0143913. URL `https://doi.org/10.1063/5.0143913`.

T. H. von Kármán and W. R. Sears. Airfoil theory for non-uniform motion. *Journal of the Aeronautical Sciences*, 5(10):379–390, 1938. URL `https://doi.org/10.2514/8.674`.

H. Wagner. Über die entstehung des dynamischen auftriebes von tragflügeln. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 5(1):17–35, 1925. doi: https://doi.org/10.1002/zamm.19250050103.

Z. J. Wang. Efficient implementation of the exact numerical far field boundary condition for poisson equation on an infinite domain. *Journal of Computational Physics*, 153(2): 666–670, 1999. ISSN 0021-9991. doi: https://doi.org/10.1006/jcph.1999.6289.

M. West and J. Harrison. *Bayesian Forecasting and Dynamic Models (2nd Ed.)*. Springer New York, New York, 1997. ISBN 0387947256. doi: https://doi.org/10.1007/b98971.

A. Wiegmann and K. P. Bube. The Explicit-Jump Immersed Interface Method: Finite Difference Methods for PDEs with Piecewise Smooth Solutions. *SIAM Journal on Numerical Analysis*, 37(3):827–862, jan 2000. ISSN 0036-1429. doi: 10.1137/S0036142997328664. URL `http://epubs.siam.org/doi/10.1137/S0036142997328664`.

D. R. Williams and R. King. Alleviating unsteady aerodynamic loads with closed-loop flow control. *AIAA Journal*, 56(6):2194–2207, 2018.

X. Xu, A. Gementzopoulos, G. Sedky, A. R. Jones, and F. D. Lagor. Design of optimal wing maneuvers in a transverse gust encounter through iterated simulation or experiment. *Theoretical and Computational Fluid Dynamics*, pages 1–20, 2023.

X. Yang, X. Zhang, Z. Li, and G. W. He. A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations. *Journal of Computational Physics*, 228(20):7821–7836, 2009. ISSN 10902716. doi: 10.1016/j.jcp.2009.07.023. URL `http://dx.doi.org/10.1016/j.jcp.2009.07.023`.

K. Yu. *Multi-resolution Lattice Green's Function Method for High Reynolds Number External Flows*. PhD thesis, Caltech, Pasadena, CA, June 2021.