Universidad Carlos III de Madrid

ESCUELA POLITÉCNICA SUPERIOR

Department of Bioengineering and Aerospace Engineering

Bachelor's Degree in Aerospace Engineering

Bachelor Thesis

Analysis of the Effect of Gusts on an Array of Plates with Ground Effect

---

Author: Álvaro Martín-Sampayo

Tutor: Óscar Flores

Madrid, June 2014

Acknowledgements

Agradecimientos en primer lugar a mi familia, por aguantarme, apoyarme y confiar ciegamente en mí. Son lo más importante en mi vida, y sin ellos no habría conseguido llegar hasta aquí. Un recuerdo especial para mi abuela Pepita, la mejor abuela del mundo.

Gracias también a todos mis profesores y compañeros de clase, con los cuales ha habido un ambiente de trabajo impresionante durante estos cuatro años y que sé que echaré mucho de menos. Agradecimientos especiales a mi tutor, Óscar, que aparte de haber sido un profesor ejemplar me ha ofrecido en todo momento una ayuda invaluable para la realización de este proyecto.

Y sin olvidarme de mis amigos de siempre, que espero que continúen durante muchos años a mi lado para que podamos disfrutar juntos de muchas alegrías. Gracias por los buenos ratos también a los que puedan estar un poco más lejos, en especial a mis amigos de ese fantástico año en Purdue, y a los que sé que siempre llevo conmigo.

Abstract

This thesis studies the aerodynamic forces on an array of flat plates near the ground. Discrete wind gusts overlap a constant free-stream velocity, and the behaviour of the forces acting on both plates are analyzed, emphasizing the influence of the trailing wakes on the system. The simulation employs a program based on unsteady two-dimensional potential flow, discretizing the airfoils into vortex elements. The program has been developed specifically as part of this thesis. All the work performed is thoroughly presented in this document.

The results of the analysis provide an insight into the influence of different parameters on the forces acting on the plates. The trailing plate is particularly affected by the strong vorticity of the first plate's wake, creating a visible interaction. Computational parameters have been chosen carefully via validation against analytical solutions. Lastly, an application of the results on solar panel design is proposed. Essentially, the problem has been positively analyzed with the proposed method. The findings attained open the door for future development of the method, for either this particular scenario or a different one.

Keywords: Computational, Unsteady Aerodynamics, Flat plate, Discrete vortex, Gust, Ground effect.

Contents

List of Figures

Chapter 1

Introduction

This document intends to synthesize all of the work performed for this Bachelor thesis. The thesis has been conceived as an open-ended research project, centered on the applications of unsteady computational aerodynamics to a specific problem: unsteady flow over flat plates in ground effect. The project is introduced in this section.

1.1 Scientific background

Fluid mechanics is the branch of Physics that deals with problems where materials are fluids (liquids, gases) rather than solids. Aerodynamics is the area within fluid mechanics whose objective is to study fluid forces on solids. It deals with the motion of any type of object in air, or the motion of masses of air over stationary objects. Aerodynamics is popularly known as the science of fast moving objects, like race cars, high-speed trains and specially aircraft. While these are some of the most extensive applications, aerodynamics is also used for Civil Engineering applications, atmospherical models, etc.

Fluid dynamics are thus a primary concern in many engineering problems. Until recently, many fluid dynamics problems were complicated and hard to solve, specially in complex cases. For aerodynamics particularly, finding exact solutions to most problems is practically impossible, and analytical approximations take enormous amounts of effort. In order to simplify the solution to the most usual problems, the potential flow theory was developed to save time and resources, creating a valid framework to analyze some typical situations, under some reasonable assumptions. Even though potential flow theory is overly simple for many situations, it is still employed in the aeronautical industry to address some engineering problems where its assumptions hold true, specially when reduced computational costs are required.

With the arrival of modern-era computing, calculations required by numerical potential flow methods could be done very rapidly. This made numerical solutions based on potential flow the most common method in use by aerodynamicists. These aerodynamic potential flow codes are also referred to as panel codes, since their elements take the shape of panels when in 3D.

The simplicity of potential flow theory allows its computational requirements to be much lower than those needed for other CFD approaches. However, these other methods, like for example Reynolds-Averaged Navier-Stokes equations solvers (RANS), are considered more accurate since they do not take so many assumptions. However, if the limitations of the potential flow method are considered and the problem is suitable, panel codes offer a simple, reliable tool to analyze aerodynamic problems. Typically, solutions based on panel codes are used to obtain preliminary results that are later on enhanced through advanced RANS methods, which may offer more detailed simulations.

1.2 Motivation

This thesis deals with the effects of wind gusts on an array of flat plates near the ground. It is a simple scenario that is very well suited for using unsteady potential methods. The existence of transient phenomena in the problem makes the necessary number of computations to vastly increase in size, as compared to a steady problem, as the complexity of the problem intensifies. The potential flow method, having reasonably small computational requirements, provides an appropriate method to efficiently investigate this type of problem. In order to perform the analysis, a MATLAB code was written from scratch. It was designed as a flexible program, capable of running a variety of scenarios that could provide an extensive set of results for posterior analysis.

As a research project, applications to this particular case are not predetermined. Several scenarios can be conceived where the response of such a system is of interest. For example, an array of solar panels exposed to strong winds, an array of metal roofs that provide shade on a parking lot, or even a structural component of an aircraft or moving vehicle with this particular shape.

The possible conclusions arising from this problem can not be inferred beforehand, as the effects that the different parameters of this scenario may play in the results is uncertain.

1.3 State of the art

The kind of unsteady potential method employed for this thesis is fairly developed. Most of the characteristics of the code created for this task follow the guidelines set by J. Katz and A. Plotkin in their book *Low-Speed Aerodynamics* [1], where Chapter 13 is entirely dedicated to unsteady incompressible potential flow.

Analytical solutions for unsteady potential flows have been found for very simple problems. The most representative one is the sudden acceleration of a flat plate, which was developed by Herbert A. Wagner in his 1925 paper "Über die Entstehung des dynamischen Auftriebes von Tragflügeln" (On the formation of dynamic lift on wings). His solution was proved valid, and states that lift gradually builds up as an airfoil is set into motion, progressing towards a steady value. His findings have been used in the validation of the results of this work.

J. Katz developed back in 1979 a discrete vortex method that included separation for airfoils at very high angles of attack. The results were positive, and reproduced periodic oscillations seen in separated flows. However, it was required to preset a point for the separation of the flow, which was done through experiments and flow visualization [2]. This type of analysis does still only consider a single flying plate. However, the model for the separation of the flow near the leading edge is an advancement in the capabilities of this kind of unsteady potential flow solver.

H. Aziz and R. Mukherjee studied in 2010 the sudden acceleration of a system of two airfoils. The method employed by them was very similar to the one that has been used in this project. Their results corroborate the existence of a notable influence of the first airfoil's wake on the forces acting on the second one. The most characteristic feature is the appearance of sudden, unsteady forces on the trailing airfoil when the intense wake of the first airfoil reached it. By displacing the airfoils away from each other vertically, these effects could be reduced. The distance among the airfoils also played a role in the intensity of the perturbations. In addition, the shapes of the trailing wakes were significantly altered from the typical, rolling vortex shape [3].

1.4 Document Structure

This document contains all information related to the project. After this introduction, Chapter 2 presents the objectives of the project and details the problem that is to be solved.

Then, Chapter 3 thoroughly describes the method that has been used. Firstly, the theory of potential flow is presented. Afterwards, the numerical model is developed. Finally, the MATLAB program is described. This chapter is complemented by appendices A, B and C, which provide additional information on the numerical method and the MATLAB program.

After this, Chapter 4 presents how the analysis has been carried out. The results of the analysis are presented in Chapter 5, and finally an example of a practical application of the method to solar panel fields design is presented in Chapter 6.

Lastly, Chapter 7 presents the conclusions extracted from this work, together with possible future enhancements and other remarks.

At the end of the document, the appendices provide additional information, such as the code validation process, the MATLAB program source code or the Budget of the project. After these, the Bibliography closes the document.

Chapter 2

Objectives

This chapter lays out the main purposes of the project. Section 2.1 *Geometry of the problem* describes in detail the specific environment that is to be analyzed.

The intention of the project is to investigate the behavior of several plates subject to wind gusts. The problem is to be solved using a numerical method based on the unsteady potential flow theory, as introduced in Chapter 1. Several steps need to be fulfilled in order for the project to be successful:

1. Understand the physical significance of the problem.
2. Determine a valid approach to solve the problem.
3. Build a suitable numerical model for the problem.
4. Solve the problem according to the chosen method.
5. Analyze and validate the findings in order to reach a conclusion.

Because of the nature of the numerical models, a series of assumptions are to be made. Then, the results are dependent upon the adherence to the established assumptions and circumstances. It is important to be aware of the limitations imposed by these conditions. These are developed and described in detail in Section 3.1.2. *Potential Flow Theory*.

The most vital part of the project is the numerical simulation that has to be developed. The following list provides the set of requirements that it needs to satisfy.

- A code written in MATLAB will carry a numerical simulation based on unsteady potential flow, using the discrete vortex method.
- The MATLAB code will allow different input parameters based on:
  - The geometry of the plates.
  - The amplitude and period of the wind gust.
  - Computational parameters that may affect the solution.
- Gathered data will be analyzed to achieve conclusions on the physical phenomena that may appear, as well as on the influence of the different parameters on the solution.
- It will be possible to obtain a visual representation of the problem as the scenario develops.
- Additionally, it will be possible to simulate different environments than those initially proposed.

2.1 Geometry of the problem

A detailed description of the geometry of the problem is given in this section. The primary scenario consists of two flat plates of chord c fixed in space and separated by a specific distance D. The plates will be located at a distance H from the ground. Their angle of attack will be α for both of them.

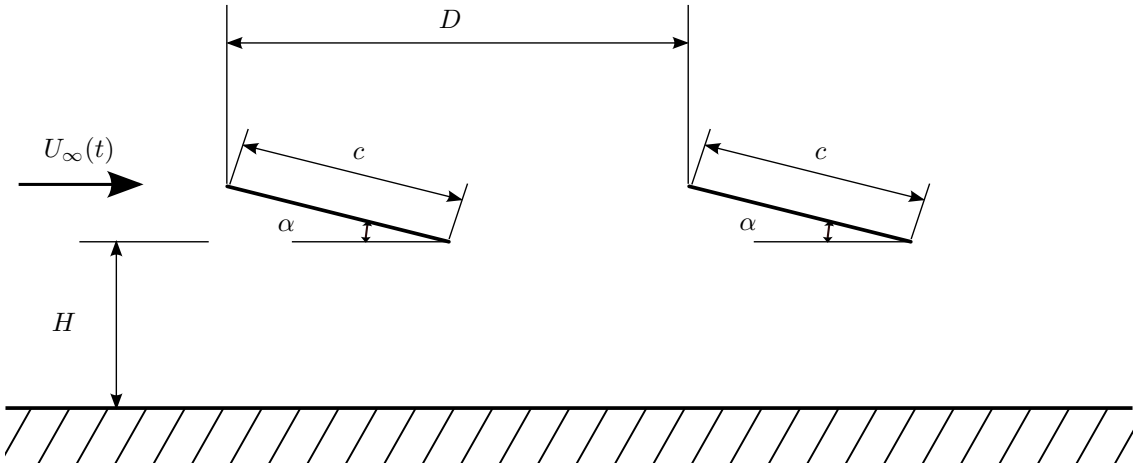


Figure 2.1: Geometry of the problem.

This basic configuration still allows for a wide range of parameters to be tested. For instance, by changing the ground distance or even removing the ground, such that $H = \infty$, different phenomena that may depend on this parameter will be easily identified.

This system of two plates is then subject to different wind gusts to observe the aerodynamic behavior and the loads. The method employed to model the gusts is presented in Section 3.2.2 *Modelling wind gusts*.

The data that has been presented in this chapter intends to be an introductory set of goals, which define the base for the development of the project. More exhaustive information on the cases of interest that have been analyzed, as well as an interesting reading on non-dimensionalization of some of the parameters presented in this section can be found in Chapter 4 *Procedure*.

Chapter 3

Method

This chapter first presents in detail the potential flow theory, the basis supporting the computational method that has been employed to solve the problem. The numerical method is introduced in Section 3.2. Then, in Section 3.3 the MATLAB program that has been written for the project is described.

3.1 Theoretical Basis

An introduction to the fundamental equations of fluid mechanics is required to understand the applied method. Following subsections provide a review of the most elementary concepts of fluid mechanics that are necessary for the task. Later, the potential flow theory is developed, as well as the discretization method employed for the numerical approach.

3.1.1 Air Dynamics

Continuum Hypothesis

Fluids are modeled according to the continuum hypothesis. Since gases and liquids are composed by molecules allowed to move freely, it would be impossible to keep track of all of them. Instead, the fluid is considered to be continuous, and its physical properties such as the density ρ, the temperature and the pressure p will vary along this continuum. A fluid particle is defined as a small parcel of this continuum. This particle will be located at position $\bar{x}$ at time t. Then, fluid characteristics such as density or velocity will be defined as functions of $\bar{x}$ and t[4].

Circulation and Vorticity

Vital concepts for understanding potential flow are circulation and vorticity. Circulation Γ physically represents fluid motion along a path. The circulation along a curve L is defined as

$$\Gamma = \int_L \bar{v} \cdot \mathrm{d}\bar{l} \tag{3.1}$$

being $\mathrm{d}\bar{l}$ the differential of the line element.

If the path is a close curve and we apply the Stokes theorem, then, if this closed curve bounds a surface Σ whose surface differential is $\mathrm{d}\sigma$ and its normal vector is $\bar{n}$:

$$\Gamma = \oint_L \bar{v} \cdot \mathrm{d}\bar{l} = \int_\Sigma (\nabla \wedge \bar{v}) \cdot \bar{n} \mathrm{d}\sigma \tag{3.2}$$

The vector $\bar{\omega} = \nabla \wedge \bar{v}$ is called vorticity [4].

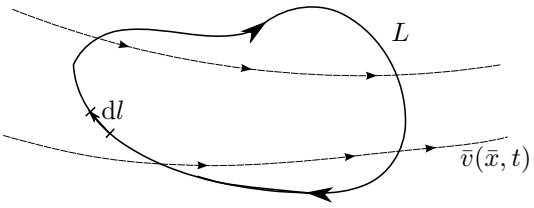


Figure 3.1: Representation of the curve L and the integration differential $\mathrm{d}\bar{l}$.

If Σ is a small enough surface, and we observe a very small closed curve contained in a plane normal to $\bar{n}$, then the circulation along $\mathrm{d}\bar{l}$ reduces to [4]:

$$\Gamma = \oint_L \bar{v} \cdot \mathrm{d}\bar{l} = (\nabla \wedge \bar{v}) \cdot \bar{n} \tag{3.3}$$

Thus, if $\bar{\omega} = \nabla \wedge \bar{v}$ and $\bar{n}$ have the same direction, the circulation along an infinitesimally small circular path contained in a plane normal to $\bar{n}$ will be maximum and is represented by the vorticity [4]. Vorticity is also typically interpreted as the rotation or spinning rate of fluid particles. An irrotational flow is one such that $\bar{\omega} = 0$ everywhere in the flow.

Conservation Equations

The conservation equations provide the mathematical sustain on which flows can be studied. These equations can be written to express the conservation of mass, momentum and energy over a control volume or to describe the balance occurring locally. The formulation of interest is the latter, using local variation terms (derivatives), which is presented below.

The first conservation equation of fluid mechanics is the mass conservation law.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \bar{v}) = 0 \tag{3.4}$$

The mass conservation equation accounts for the preservation of mass in a fluid. When considering an infinitesimal control volume, the first term in equation (3.4) represents the rate of change in mass of the fluid [5], and the second term the net variation of mass due to motion of the fluid.

The second one is the momentum conservation law:

$$\rho \frac{\mathrm{D}\bar{v}}{\mathrm{D}t} = -\nabla p + \nabla \cdot \bar{\bar{\tau}} + \rho \bar{f}_m \tag{3.5}$$

where $\frac{\mathrm{D}(\ )}{\mathrm{D}t}$ is a material derivative; $\bar{\bar{\tau}}$ is the stress tensor, responsible for viscous stresses; and the term $\bar{f}_m$ represents mass forces [4].

The conservation of momentum, usually referred to as the Navier-Stokes equations, represents Newton's second law of motion. It is thus formulated in forces. If an infinitesimal control volume is considered, then the left-hand side in equation (3.5) represents the rate of increase of momentum

within the control volume plus the net rate at which momentum leaves the control volume.[1] This term hence equals the forces acting on the fluid inside the control volume [5]. There are three types of forces represented on the right-hand side: pressure forces, represented by $-\nabla p$; viscous forces, represented by $\nabla \cdot \bar{\bar{\tau}}$; and mass forces, represented by $\rho \bar{f}_m$, which are typically due to gravity.

The third equation is the energy conservation law:

$$\rho \frac{\mathrm{D}e}{\mathrm{D}t} = -p\nabla \cdot \bar{v} + \bar{\bar{\tau}} : \nabla \bar{v} - \nabla \cdot \bar{q} + Q_q + Q_r \tag{3.6}$$

where $\bar{q}$ is the heat flux vector and Q_q and Q_r are sources of energy related to chemical reactions and radiation.

The energy equation states that energy is conserved, providing balance between the different forms of energy. The left-hand side represents the rate of change of internal energy, and the right-hand side terms represent respectively: the compression work, the work due to viscous stresses, corresponding to the mechanical energy dissipation into heat; the heat transfer, and the internal sources of energy [4].

3.1.2 Potential Flow Theory

This section develops the physical and mathematical concepts on which the computational method is based.

Required simplifications

From the conservation equations presented in the previous section, together with the equation of state for a perfect gas like air, it is possible to describe simple flows, for which analytical solutions have been found to the Navier-Stokes equations. However, the flexibility to examine complex scenarios, as required in aerodynamics, makes it necessary to simplify this model in order to be able to find solutions to a broader range of problems. A series of assumptions will allow us to use the potential flow approach.

[1] Recalling the definition of material derivative, $\frac{\mathrm{D}(\ )}{\mathrm{D}t} = \frac{\partial(\ )}{\partial t} + \bar{v} \cdot \nabla(\ )$, it is apparent that the first term of the material derivative of the velocity $\left(\frac{\mathrm{D}\bar{v}}{\mathrm{D}t} = \frac{\partial \bar{v}}{\partial t} + \bar{v} \cdot \nabla \bar{v}\right)$ represents the rate of increase of momentum inside the control volume, whilst the second term represents the flux of momentum, i.e. the net rate at which momentum is leaving the local infinitesimal control volume [4, 5].

The first simplification comes from neglecting the viscous stresses. This is a very reasonable assumption, specially for gases at high Reynolds numbers, as they are typically encountered in aerodynamics.[2] This set of simplified equations is called the Euler equations. The following equation shows the Euler equation for momentum conservation, derived from (3.5).

$$\rho \frac{\mathrm{D}\bar{v}}{\mathrm{D}t} = \rho \left(\frac{\partial \bar{v}}{\partial t} + \bar{v} \cdot \nabla \bar{v} \right) = -\nabla p + \rho \bar{f}_m \tag{3.7}$$

Another perfectly plausible assumption for the case of interest is to consider the flow as incompressible, such that the continuity equation (3.4) becomes

$$\nabla \cdot \bar{v} = 0. \tag{3.8}$$

Only for fluid velocities close to the speed of sound do compressibility effects become apparent, so the incompressibility assumption is wise for low-speed flows.

Finally, irrotational flow is necessary to allow the use of potential flow theory. Irrotational flow implies that

$$\bar{\omega} = \nabla \wedge \bar{v} = 0 \tag{3.9}$$

everywhere in the flow. This is a direct conclusion of the previous assumptions of incompressibility and negligible viscous forces due to high Reynolds numbers in two-dimensional space [7]. In order to understand the concept in a physical way, let us consider that the free-stream velocity has zero vorticity $\bar{\omega} = 0$, as expected in a uniform flow. Then, since Re is high, the diffusion of vorticity due to viscosity is negligible [7], so there is no cause for the fluid particles to start rotating as they move across the flow field [5]. The vorticity of the fluid field remains zero throughout. This corresponds to a state of pure translation, as presented in figure 3.2.

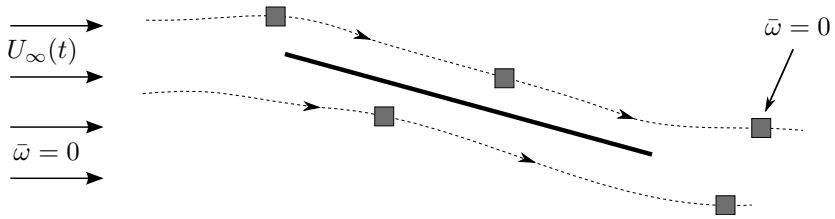


Figure 3.2: Representation of an irrotational flow around a flat plate. The squared boxes represent fluid particles, which do not rotate.

Ultimately, the assumptions that are required in order to have a pontential flow are recollected below.

- Inviscid Flow: $\bar{\bar{\tau}} = 0$.

[2]The Reynolds number $Re = \frac{UL}{\nu}$, where U is the characteristic velocity of the flow, L is the characteristic length and ν is the kinematic viscosity, is a paramount non-dimensional parameter in fluid mechanics. It quantifies the relative importance that inertial forces have with respect to viscous forces. For low-viscosity fluids such as air, and specially at high characteristic lengths and velocities, Re is so high that viscous forces can be entirely neglected. Consider for instance an airfoil of chord 1m subject to a free stream flow of 10 $\mathrm{ms}^{-1} = 36$ kmh. If the fluid is air, whose kinematic viscosity at sea level in the standard atmosphere is 1.4607×10^{-5} $\mathrm{m}^2\mathrm{s}$ [6], then the corresponding Re number would be $Re = 6.85 \times 10^5$. This would render inertial forces around 600,000 times larger in magnitude than viscous forces.

- Incompressible Flow: $\nabla \cdot \bar{v} = 0$.
- Irrotational Flow: $\bar{\omega} = \nabla \wedge \bar{v} = 0$.

These conditions impose a series of limitations on the types of flows on which the potential flow theory can be applied. Flows where viscous forces are relatively important cannot be analyzed in this manner. For instance, the boundary layer does not exist in potential flow. Also, any flow where turbulence plays a major role, such as in stalling airfoils or blunt bodies cannot be analyzed using potential flow. This is why for more complex flow fields where these effects play a major part, other techniques are used in numerical methods; for example finite difference methods [1], albeit at a much higher computational cost.

However, there exist current solvers for airfoil design which employ potential flow in parts of the flow analysis. A typical approach is to couple a boundary layer solver for the flow near the surfaces, where viscous forces are relevant, and use a simpler potential flow solver for the remainder of the flow. In some cases, some corrections can be applied to the potential solutions to account for compressibility effects at a very low cost.

Potential flows

Recalling the simplifications proposed in the previous section, a two-dimensional, inviscid, incompressible and irrotational flow is considered. An irrotational flow as defined in equation (3.9), since $\nabla \wedge (\nabla \bar{x}) = 0$ for any vector $\bar{x}$, mathematically allows the following definition:

$$\bar{v} = \nabla \phi \tag{3.10}$$

The scalar function ϕ is called the velocity potential, hence the term potential flow. This allows the following definitions:

$$u = \frac{\partial \phi}{\partial x} \qquad \text{and} \qquad v = \frac{\partial \phi}{\partial y} \tag{3.11}$$

where u is the component of the velocity along the x direction and v along the y direction. Substituting (3.11) into the continuity equation for incompressible flow (3.8) yields

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0 \tag{3.12}$$

which is the definition of Laplace's equation for ϕ:

$$\nabla^2 \phi = 0. \tag{3.13}$$

In the same fashion, equation (3.8) allows the following definition of a stream function ψ for two-dimensional flow:

$$u = \frac{\partial \psi}{\partial y} \qquad \text{and} \qquad v = -\frac{\partial \psi}{\partial x} \tag{3.14}$$

and finally, substituting (3.14) into (3.9) yields

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = 0 \tag{3.15}$$

which is again Laplace's equation for ψ. This means that both the velocity potential ϕ and the stream function ψ are harmonic potential functions. Then, any solution of Laplace's equation can be a potential flow, since it will satisfy equations (3.12) and (3.15).

Moreover, since Laplace's equation is linear, the useful superposition principle can be employed: any sum of solutions of Laplace's equation will be another valid solution. For this reason, potential flow problems are solved superimposing a set of elementary solutions [5].

The flow can be analyzed introducing the complex potential

$$f(z) = \phi(x, y) + i\psi(x, y) \tag{3.16}$$

which is an analytic function[3] in the complex plane

$$z = x + iy = re^{i\theta}. \tag{3.17}$$

This means that both its real and imaginary parts satisfy Laplace's equation, and are always valid forms of the velocity potential and the stream function [7]. The derivative of the complex potential is the complex velocity

$$\frac{\mathrm{d}f}{\mathrm{d}z} = v_x - iv_y = (v_r - iv_\theta)\, e^{-i\theta} \tag{3.18}$$

as can be obtained by derivation of (3.16) in either the $z = x$ or the $z = iy$ direction, recalling the definitions of ϕ and ψ given in (3.11) and (3.14).

Elementary solutions

The most common complex potentials f that represent elementary flow solutions are the free stream, the source, the sink, the vortex and the doublet. In this study only the free stream and the vortex are utilized.

The simplest one is the free stream flow, whose complex potential takes the form

$$f(z) = U_\infty z \tag{3.19}$$

and represents constant flow of velocity U_∞ along the x direction, which can be easily proved plugging equation (3.19) into (3.18) and solving for the velocities. The resulting flow field is presented in figure 3.3.

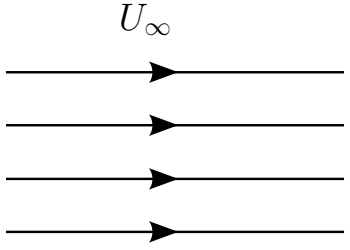


Figure 3.3: Uniform flow. Solution for $f(z) = U_\infty z$.

[3] "A function of complex variable $f(z) = \mathcal{R}(x, y) + i\mathcal{I}(x, y)$ is said to be analytic in a region if it is differentiable at every point of that region, and the partial derivatives satisfy the Cauchy-Riemann equations $\left(\frac{\partial \mathcal{R}}{\partial x} = \frac{\partial \mathcal{I}}{\partial y} \quad \text{and} \quad \frac{\partial \mathcal{R}}{\partial y} = -\frac{\partial \mathcal{I}}{\partial x}\right)$, which guarantee that the value of $\mathrm{d}f/\mathrm{d}z$ is independent of the direction in which it is calculated. If a function is analytic, then it can be demonstrated that its real and imaginary parts satisfy Laplace's equation."[7]

The vortex element creates a circular motion around a point, as represented in figure 3.4. It takes the following form:

$$f(z) = -\frac{\Gamma i}{2\pi} \ln (z - z_o) \tag{3.20}$$

By setting the location of the vortex at the origin such that $z_o = 0$ and using (3.18), recalling the definition of z given in (3.17), the velocities induced by a vortex element are found to be

$$\frac{\mathrm{d}f}{\mathrm{d}z} = -\frac{\Gamma i}{2\pi z} = -\frac{\Gamma i}{2\pi r} e^{-i\theta} \quad \longrightarrow \quad v_r = 0 \quad \text{and} \quad v_\theta = \frac{\Gamma}{2\pi r}. \tag{3.21}$$

Considering the concept of circulation introduced in page 7, the strength Γ of the vortex element is seen to be equal to the circulation induced by itself on the flow: $\int_0^{2\pi} v_\theta r \mathrm{d}\theta = \Gamma$.

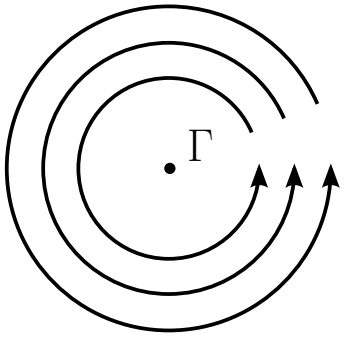


Figure 3.4: Vortex element. Solution for $f(z) = -\frac{\Gamma i}{2\pi} \ln (z - z_o)$.

Vortex elements are very useful in modeling flying objects, since circulation is required for a body to generate lift.[4] Airfoils, inclined plates are objects that generate lift, and thus they need circulation created by vortex elements to be modelled. The source element and the doublet element are used to account for the thickness of a body, but since this study deals with flat plates, they are not necessary.

Let us present a sample problem where vortex elements are applied to model an airfoil. This simplified example uses the small-disturbance approach, which neglects terms that are relatively small, and separates the problem into its symmetric and antisymmetric parts. The symmetric part would account for the thickness of the body, and the antisymmetric for its angle of attack and camber [1]. Imagine a zero-thickness flat plate at an angle of attack α, so that the symmetric contribution is zero. It turns out that the airfoil can be modeled by a continuous distribution of vorticity, which creates a differential of induced velocity on the upper and lower surfaces, as presented in figure 3.5.

When the free stream velocity is added to the perturbation velocity created by the vortices, the resulting flow is the solution to the flow past a flat plate. A scheme of the flow past a flat plate, as resulting from the sum of a free stream element plus the vortex distribution is presented in figure 3.6.

[4]The Kutta-Joukowski Theorem states that, for a cylinder of any shape, or more interestingly any airfoil, the lift per unit span L' is related to the circulation Γ of the body by the expression $L' = \rho U \Gamma$, where U is the speed of the flow approaching the body [5]. This theorem is developed in page 18.

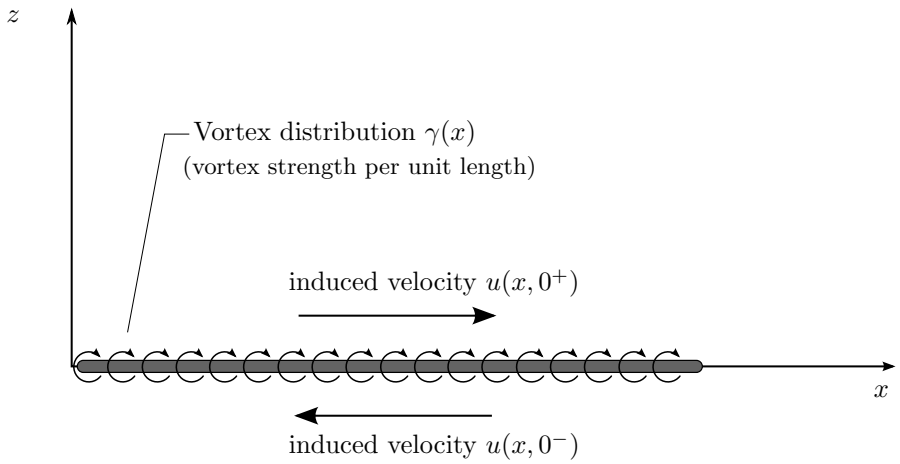


Figure 3.5: Representation of a flat plate modelled as a vortex distribution.

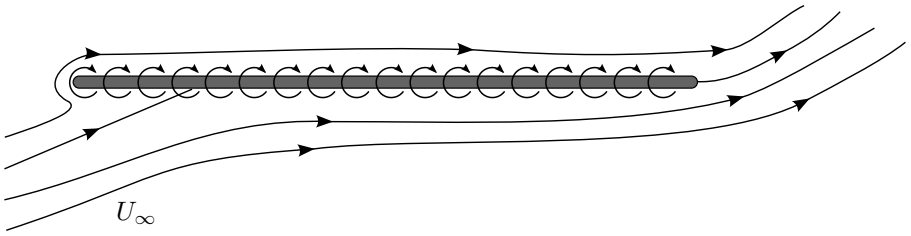


Figure 3.6: Flow past a flat plate at an angle of attack.

The required values of circulation in the plate are found thanks to the boundary conditions of the problem, but this will be developed further in the forthcoming section 3.2, when dealing with the numerical model.

The ground effect

Superimposing elementary potential flow solutions allows us to build more complex flows. Nevertheless, it is still necessary to emulate the effect of the ground, as required by the problem that is to be analyzed. This can be easily performed by means of the method of mirror images.

In essence, in potential flow any streamline can represent a solid boundary. In order to simulate the ground, it is thus required to have a horizontal streamline located where the ground should be. If the elements that constitute the model are mirrored with respect to a horizontal axis representing the ground, then the vertical velocities induced on the horizontal axis by the mirror elements and those induced by the original elements will cancel each other. Hence, this horizontal axis is the ground as it will be a horizontal streamline. A sample illustration of this concept can be seen in figure 3.7.

Ground effect has diverse effects on the aerodynamics of different bodies. For two-dimensional cases, the ground restricts the flow over a narrower space between the body and the ground wall. This also means that free elements in the flow like vortices will not be allowed to move downwards, due to the pushing effect of the mass of air close to the ground, preventing these vortices to get closer to the ground and keeping them in line with the flow. Race cars use the ground effect to

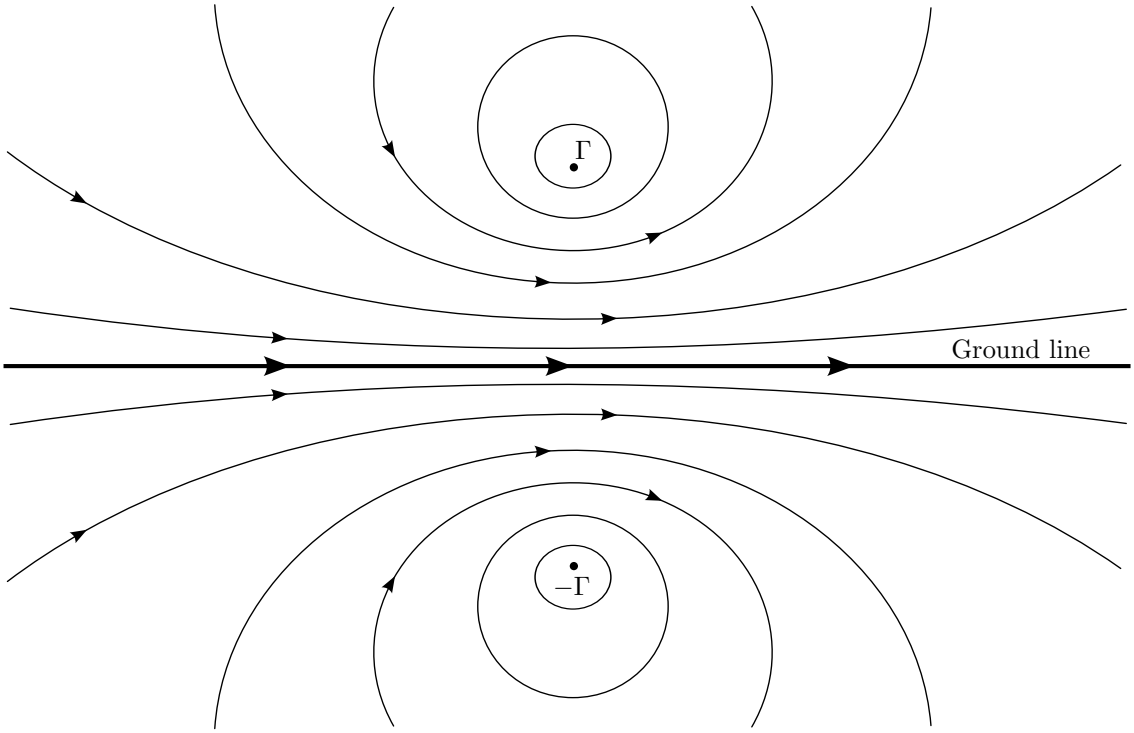


Figure 3.7: Method of mirror images applied to a single vortex to generate the ground. Lines represent streamlines.

accelerate air through a narrow section, between the car and the ground, in order to obtain a suction force towards the ground.

However, for aircraft wings in three dimensional space there are more relevant effects. Since the air is not allowed to freely move downwards, wingtip vortices are not allowed to form completely. Wingtip vortices are not observed in 2D, but they generate a downwash in the flow surrounding the aircraft, resulting in reduced effective angle of attack, meaning lower lift and increased drag. Drag due to downwash is called induced drag. Thanks to the ground effect, the downwash is greatly reduced, offering a boost in lift and a reduction in drag and thus in the thrust required [5]. This allows the aircraft to virtually *float* when it is approaching the ground.

Pressures and Loads

The procedure to find the forces created by the flow utilizes the Bernoulli equation. The Bernoulli equation is a derivation of the momentum equation for inviscid flows, the Euler equation (3.7) introduced in page 10. It is derived below into a form valid for unsteady cases.

We begin by recalling the Euler equation

$$\rho \frac{\mathrm{D}\bar{v}}{\mathrm{D}t} = \rho \left(\frac{\partial \bar{v}}{\partial t} + \bar{v} \cdot \nabla \bar{v} \right) = -\nabla p + \rho \bar{f}_m \qquad \text{(3.7 revisited)}$$

then neglect mass forces[5] and use the following vector identity

$$\bar{v} \cdot \nabla \bar{v} = \nabla \frac{\bar{v}^2}{2} - \bar{v} \wedge (\nabla \wedge \bar{v}) = \nabla \frac{v^2}{2} - \bar{v} \wedge \bar{\omega} \tag{3.23}$$

to rewrite (3.7) as

$$\frac{\partial \bar{v}}{\partial t} + \nabla \frac{v^2}{2} - \bar{v} \wedge \bar{\omega} = -\frac{\nabla p}{\rho}. \tag{3.24}$$

The term containing $\bar{\omega}$ is readily zero because an irrotational flow is being considered. Then, the first term on the left-hand side can be rewritten as

$$\frac{\partial \bar{v}}{\partial t} = \frac{\partial}{\partial t} \nabla \phi = \nabla \left(\frac{\partial \phi}{\partial t} \right) \tag{3.25}$$

Substituting equation (3.25) into (3.24) and rearranging provides

$$\nabla \left(\frac{p}{\rho} + \frac{v^2}{2} + \frac{\partial \phi}{\partial t} \right) = 0, \tag{3.26}$$

which means that the term in parentheses is a function of time exclusively, such that

$$\frac{p}{\rho} + \frac{v^2}{2} + \frac{\partial \phi}{\partial t} = C(t) \tag{3.27}$$

which is the Bernoulli equation for unsteady, inviscid, incompressible, irrotational flow [1].

Since the term on the Bernoulli equation is, at a given moment, constant everywhere, it is possible to compute the pressures by comparing this value at two different points in the fluid. Consider a point far away into the undisturbed upstream flow where the velocity is simply U_∞, and another point P where pressures need to be found and the velocities are known. Then, applying (3.27):

$$\left[\frac{p}{\rho} + \frac{v^2}{2} + \frac{\partial \phi}{\partial t} \right] \Bigg|_P = \left[\frac{p_\infty}{\rho_\infty} + \frac{U_\infty^2}{2} + \frac{\partial \phi_\infty}{\partial t} \right] \tag{3.28}$$

so that if $\phi_\infty = \text{constant}$, one can write the following expression,

$$\frac{p_\infty - p}{\rho} = \frac{\partial \phi}{\partial t} + \frac{v^2 - U_\infty^2}{2} \tag{3.29}$$

which permits to find the pressures at any given point.

Once the pressures are computed on the surface of a body, the aerodynamic loads can be inferred. In this project lift L, drag D and momentum M_y about the leading edge have been evaluated. Consider a flat plate as in figure 3.8 whose pressure on the surface is known. Then the definition of normal force is,

$$F_n = \int_0^c \Delta p \mathrm{d}c, \tag{3.30}$$

[5] Mass forces $\bar{f}_m$ are usually neglected when dealing with gases. This is associated to their comparatively negligible order of magnitude. This is acceptable because the following equation holds, where the subscript $_c$ indicates characteristic magnitude [7]:

$$\frac{O(\rho \bar{v} \cdot \nabla \bar{v})}{O(\rho \bar{f}_m)} = \frac{v_c^2}{L_c \bar{f}_{m_c}} \gg 1 \tag{3.22}$$

.

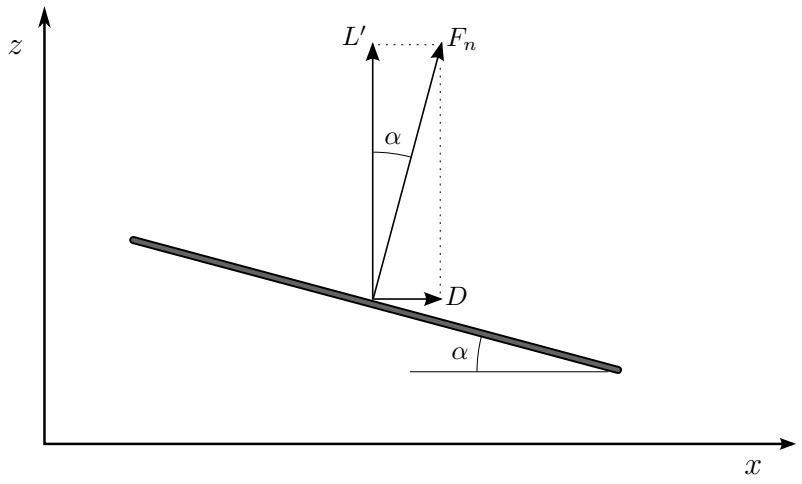


Figure 3.8: Lift and Drag forces on a flat plate.

where Δp is the differential of pressure between the upper and lower surfaces, c is the chord distance and $\mathrm{d}l$ the differential of length along the chord, as in figure 3.9. If F_n is known, then the aerodynamic forces are defined as

$$L = F_n \cos\alpha, \quad D = F_n \sin\alpha \quad \text{and} \quad {M_{y_{LE}}}^{6} = -\int_0^c \Delta p l \mathrm{d}l. \tag{3.31}$$

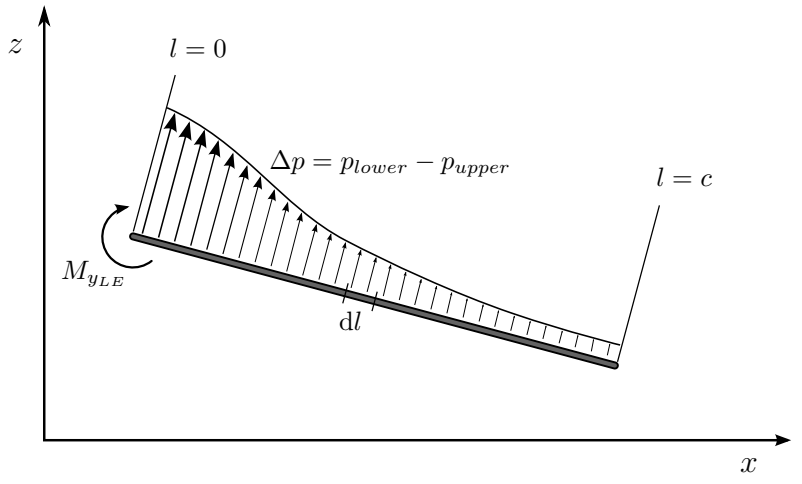


Figure 3.9: Moment about the leading edge on a flat plate.

[6]The $(-)$ sign in the equation for $M_{y_{LE}}$ is simply due to the sign convention. This work uses as standard the x direction as the horizontal direction and the z direction as the vertical direction when in two-dimensional space, as in figure 3.9. In consequence, for the right-hand-rule sign convention to hold, the y axis points in the direction towards the document from the reader's point of view.

These definitions aim to introduce the concepts of aerodynamic loads to the reader. Nevertheless, the actual method that has been used in the MATLAB program to calculate the forces is detailed in section 3.3.

3.1.3 Circulation and Lift

The Kutta condition

So far, potential flow has been presented as a simple yet powerful tool to analyze inviscid, irrotational, incompressible flows. But for lifting problems, potential flow alone does not provide any mathematical constraint that imposes a circulation on a flying body: effectively any value of circulation creates a valid flow. Despite this, when an airfoil is generating lift, its circulation must be necessarily constant and unique. The question, then, is what the correct value of its circulation is. The Kutta condition provides an answer to this question.

The Kutta condition is a mathematical consideration that serves to provide a unique solution to the circulation of a given airfoil, based on the physical fact that flow leaves the trailing edge smoothly in the absence of trailing edge separation, which is the case for low angles of attack [1].

By forcing the flow to leave the trailing edge smoothly, the circulation can be fixed to a unique value. This condition emanates from what is empirically observed in most flows, and is physically related to viscous effects. Without the Kutta condition, these effects would be absent, and the flow represented would not maintain correlation to what physically truly happens.

A flow parallel to the trailing edge is accomplished by setting the circulation at the trailing edge to zero, thus disabling the possibility of having velocities perpendicular to the trailing edge.

The Kutta-Joukowski Theorem

There exists a very powerful theorem that facilitates the calculation of lift when the circulation of a body is known. This theorem is called the Kutta-Joukowski theorem, named after both physicists [5]. It relates the lift of a body of any shape to its circulation Γ.

By calculating pressures on the surface of a cylinder through the Bernoulli equation, the same result can be obtained. The interesting aspect of this theorem is that it applies to a body of any shape.

If a body is producing lift in a fluid, this upward force must be supported by the surrounding fluid. We observe a body of circulation Γ surrounded by a very large circular control circuit of radius r, as per figure 3.10. If r is large enough, the circuit will be very far from the body, such that the velocities v induced by the body are independent of its shape, and thus only its circulation Γ is involved, rendering this process valid for any body shape.

The vertical static pressure force or buoyancy l_b on the circular boundary is the sum of the vertical components of the pressure forces acting on the circuit. By using Bernoulli's equation (3.27) for a steady case:

$$p_0 + \frac{1}{2}\rho U^2 = p + \frac{1}{2}\rho\left(U^2 + v^2 + 2Uv\sin\theta\right) \tag{3.32}$$

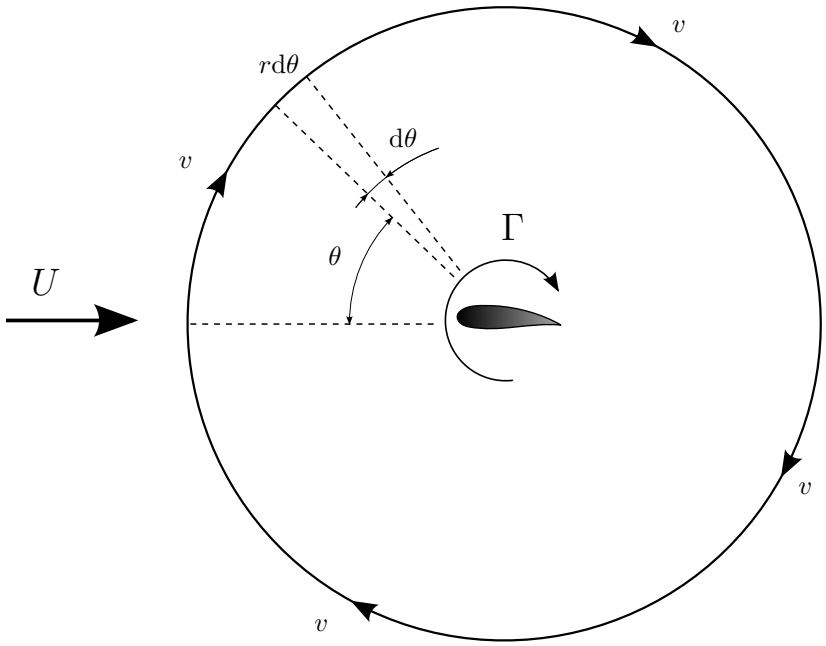


Figure 3.10: Flying body of circulation Γ inside a circular control volume.

which, if v is neglected because $v^2 \ll U^2$ [5], yields:

$$p = p_0 - \rho U v \sin\theta \tag{3.33}$$

The vertical component of the pressure force exerted by a pressure p on a differential element of length $r\mathrm{d}\theta$ is

$$-pr\sin\theta\mathrm{d}\theta \tag{3.34}$$

Now, by substituting for p from (3.33) and integrating along the boundary, the contribution to the vertical force from pressure forces is

$$l_b = -\int_0^{2\pi} (p_0 - \rho U v \sin\theta) r \sin\theta \mathrm{d}\theta = +\rho U v r \pi \tag{3.35}$$

Now we must consider the contribution to lift generated by the rate of change in vertical momentum, the inertia contribution to the lift l_i. The mass flow through a differential element is $\rho U r \cos\theta \mathrm{d}\theta$, and this mass has a vertical velocity increase of $v\cos\theta$. Therefore the change in downward momentum through a differential element is [5]:

$$-\rho U v r \cos^2\theta \mathrm{d}\theta \tag{3.36}$$

So by integrating around the boundary the lift contribution of momentum change becomes:

$$l_i = +\int_0^{2\pi} \rho U v r \cos^2\theta \mathrm{d}\theta = \rho U v r \pi \tag{3.37}$$

Adding both contributions l_b and l_i the final lift is found to be

$$L = l_b + l_i = 2\rho U v r \pi \tag{3.38}$$

At this point, it only remains to substitute into v the velocity induced by a vortex of circulation Γ, which was derived in (3.21), to obtain finally the lift per unit span L:

$$L = \rho U \Gamma. \tag{3.39}$$

An interesting remark is that the aerodynamic force according to this theorem will always be perpendicular to the velocity seen by the body. Thus, if there is no vertical velocity, the drag will be zero. However, if there are vertical components of velocity in the flow, a horizontal force will appear, which depending on the direction of the vertical flow and the sign of the circulation Γ will generate either positive or negative "drag".

The Kutta-Joukowski theorem is very useful in the calculation of aerodynamic forces in potential flow. It is on this theorem on which the force model of the computational method is based. The formulas on which lift and drag have been calculated are developed in section 3.3.1 *Calculation of loads*.

3.2 Numerical Model

3.2.1 Computational Panel Method: Lumped-vortex Discretization

Previous sections introduce the potential flow theory, which can provide analytical solutions for some simplistic or simplified cases. Even so, using numerical techniques it is possible to solve more realistic geometries. A typical formulation that is valid for both 2D and 3D flow consists of utilizing source and doublet elements to discretize the body [1]. However, since the focus of this work is on two-dimensional, thin bodies, this approach is not optimal. From now onwards, only the lumped-vortex method will be considered.

It is clear by now that any superposition of elementary solutions to the Laplace equation (3.13) is a valid solution to the potential flow. Hence it is required to find a valid combination of elements that construct the required solution. For thin plates, it is necessary to account for the lift generated and the boundary condition on the surface, which establishes that the flow normal to the surface of the plate is zero, $\bar{v} \cdot \bar{n} = 0$, where $\bar{n}$ is the unitary vector normal to the plate [1].

The discrete-vortex or lumped-vortex method discretizes the surface into lift panel elements, composed by one vortex, the lifting element; and one collocation point, which is the point where the boundary condition is imposed [1]. An example is presented in figure 3.11.

Since the lift of the symmetric airfoil acts at the center of pressure, located at the quarter chord for a flat plate [1], then the vortex element Γ will be located at $c/4$. The position of the collocation point is determined in the following manner.

It is known that the analytical solution for the circulation of a flat plate is

$$\Gamma = \pi c U_\infty \alpha \text{ [1]}. \tag{3.40}$$

Thus, the collocation point located at kc will have zero normal velocity if

$$\frac{-\Gamma}{2\pi \left[kc - (1/4)\, c\right]} + U_\infty \alpha = 0 \tag{3.41}$$

which includes the free stream velocity and the velocity induced by the vortex as derived in (3.21). Solving for k by substituting (3.40) into (3.41) yields that the collocation point must be

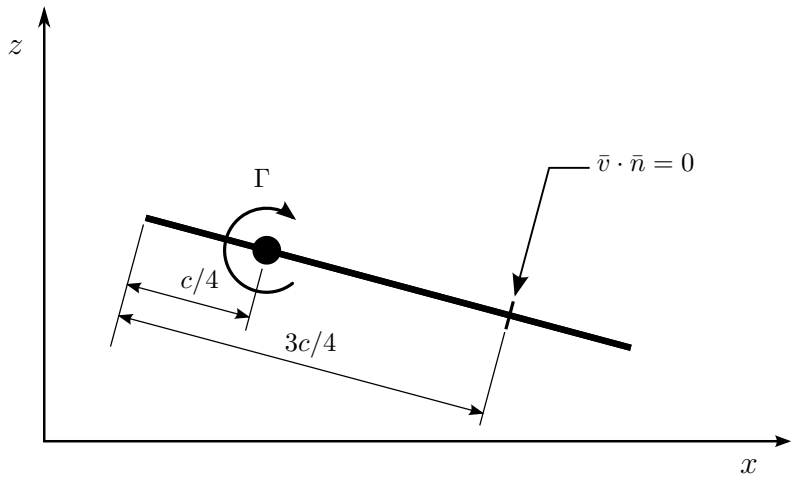


Figure 3.11: Discrete-vortex equivalent of a flat panel.

located at $k = 3/4$. By evaluating the zero-normal-flow boundary condition at $k = 3/4$, the result automatically accounts for the Kutta condition as well [1], which establishes that the circulation at the trailing edge must be zero, as was discussed in Section 3.1.3 *The Kutta condition.*

The vortex element described above forms a single panel, but it is possible to divide a surface into any number of panels with its respective discrete vortices, as in figure 3.12. This is the equivalent of a finer mesh in finite element methods, and this provides higher resemblance with a continuous model and is thus considered more accurate. Since each panel has its own collocation

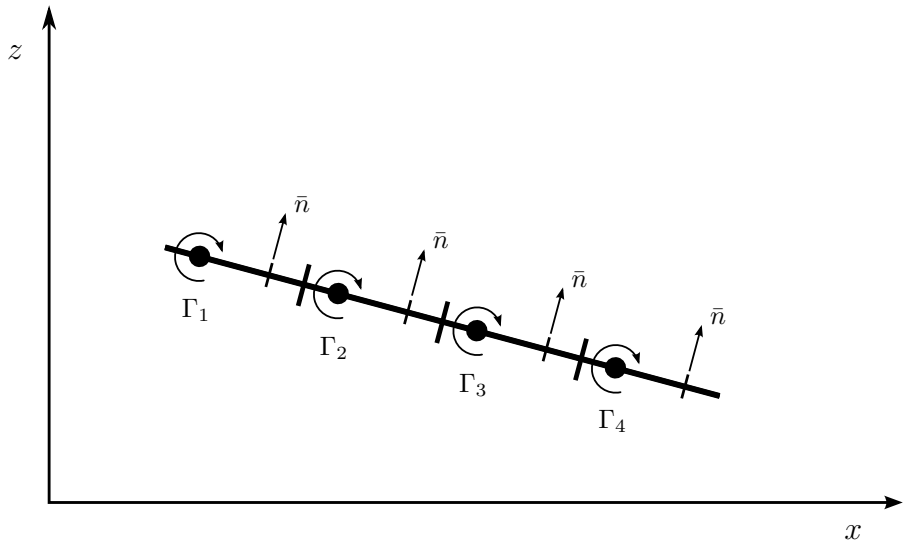


Figure 3.12: Flat plate divided into four panels.

point, if there are n panels then there are n vortices and n collocation points. By evaluating

the influence of all of the vortices Γ_n on each collocation point, n equations are obtained for n unknown values of Γ_n.

The circulation Γ of all the n vortices if found via a simple algebraic equation. Let us define the influence coefficients $a_{i,j}$ as the normal component of the velocity induced by a unit strength vortex element Γ_j on the collocation point i:

$$a_{i,j} = \bar{v}_{i,j}(\Gamma_j = 1) \cdot \bar{n}_i \tag{3.42}$$

where the velocity $\bar{v}_{i,j} = (u, w)$ induced by a vortex Γ_j is, as derived from (3.21),

$$\begin{pmatrix} u \\ w \end{pmatrix}_i = \frac{\Gamma_j}{2\pi r_{ij}^2} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} x_i - x_j \\ z_i - z_j \end{pmatrix} \tag{3.43}$$

where

$$r_{ij}^2 = (x_i - x_j)^2 + (z_i - z_j)^2 \qquad [1]. \tag{3.44}$$

Once all n influence coefficients have been calculated, the velocities induced by these vortices on the collocation point i is

$$\bar{v}_{\text{induced on } i} \cdot \bar{n}_i = \sum_{j=1}^{n} a_{i,j}\Gamma_j \tag{3.45}$$

so that the zero-normal flow condition on the collocation i can be expressed as

$$\sum_{j=1}^{n} a_{i,j}\Gamma_j + \bar{U}_\infty \cdot \bar{n}_i = 0 \tag{3.46}$$

where $\bar{U}_\infty$ is the free-stream velocity vector. This is typically rearranged as

$$\sum_{j=1}^{n} a_{i,j}\Gamma_j = -\bar{U}_\infty \cdot \bar{n}_i \tag{3.47}$$

to differentiate the unknown left-hand side (LHS) and the known right-hand side (RHS).

To find the circulation of all vortices of a plate, n equations (3.47), one for each collocation point i, are solved as a matrix problem. However, before reaching that point, a new factor needs to be introduced, the wake vorticity.

Wake model in unsteady flows

The wake is the vortex sheet that a flying object in 2D leaves as a trail when its circulation varies, i. e., during transient flows. In three dimensions, the wake model is more complicated and is being created at all times, but for a two-dimensional case the wake only develops due to unsteadiness.

Wake vortices are shedded from the trailing edge of the airfoil when the circulation of the plate varies, so it will need to be accounted for in an unsteady problem. A wake develops behind an airfoil whenever its circulation varies. This can happen for instance when the airfoil is accelerated into motion or when the incoming velocity changes, such as when subject to gusts.

The wake vortices are shedded and their circulation remain constant all the time, as presented in figure 3.13. They are free to move with the local velocity of the flow, and hence when an airfoil

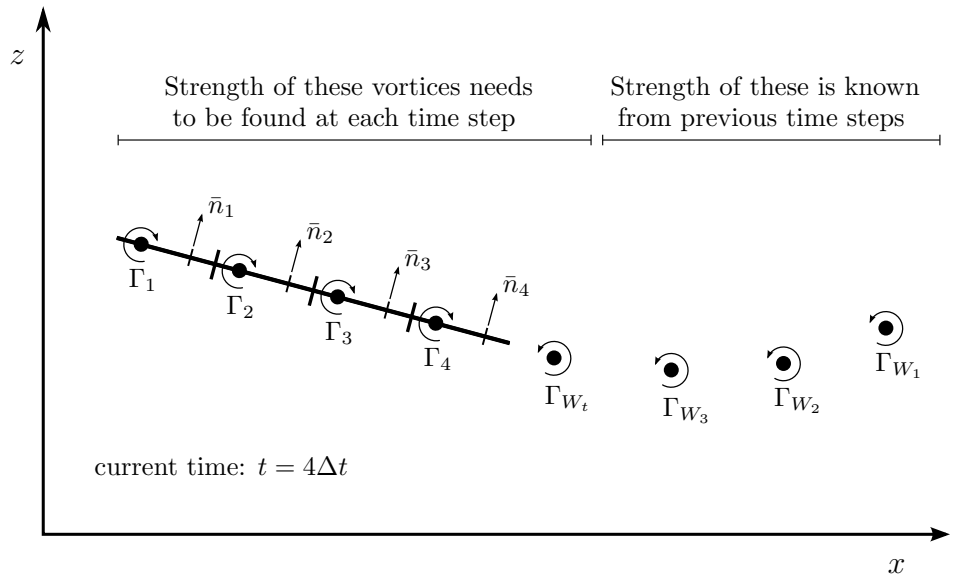


Figure 3.13: Development of the wake of a flat plate in an unsteady problem.

is set into motion their effects tend to shade off when the strongest wake vortices end up too far away.

Nevertheless, the strength of the wake vortex that is being shed at a given time is unknown, and an additional equation must be considered so that there is a unique solution. This additional equation is the Kelvin condition [1] and is illustrated in figure 3.14.

The Kelvin condition states that the total circulation of a system Γ_T must remain constant in time:

$$\frac{\mathrm{d}\Gamma_T}{\mathrm{d}t} = 0 \tag{3.48}$$

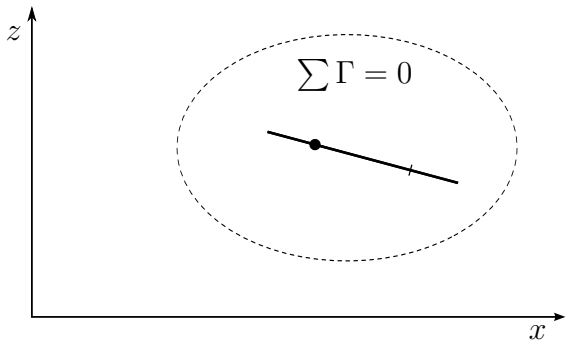
Thus, for any variation of the circulation of a plate, wake vortices are released to counteract this change in circulation of the system. The released wake vortex Γ_{W_t} is the latest vortex in the wake at each time step, and its circulation needs to be found. Its position with respect to the trailing edge is fixed throughout the computations. The distance from the trailing edge the wake vortices are shedded is discussed in Section 3.3.2 *Relevant Computational Parameters*. There can be any number of previous wake vortices, whose circulation is known and fixed. Equation (3.48) can be rewritten as

$$\Gamma(t) - \Gamma(t - \Delta t) + \Gamma_{W_t} = 0 \tag{3.49}$$

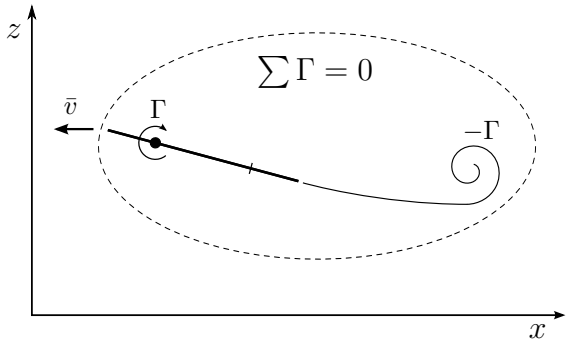
where $\Gamma(t)$ is the total instantaneous airfoil circulation, which is the sum of all the airfoil's vortices [1]:

$$\Gamma(t) = \sum_{j=1}^{n} \Gamma_j(t) \tag{3.50}$$

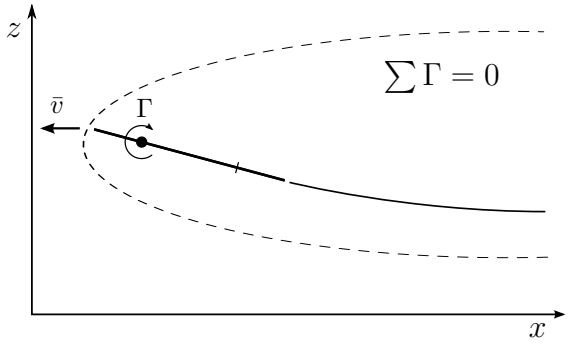
The development of a wake for unsteady problems makes it necessary to modify slightly equation (3.47). On the LHS, the new wake vortex Γ_{W_t} is added as a new unknown circulation. Fortunately, since its position is predetermined, a corresponding influence coefficient a_{iW} can be calculated for its influence on the collocation points i of the plate. The RHS term must be modified as well to account for, in addition to the free stream flow, the induced velocities produced



(a) Plate before being set into motion.



(b) Plate is suddenly set into motion. A wake develops.



(c) Plate translating. Even though the starting wake vortex is far downstream, it remains part of the system.

Figure 3.14: Kelvin condition represented by the sudden acceleration of a plate in three stages.

by all previous wake vortices, whose circulation is known. The new RHS term for the equation at collocation point i is written as [1]:

$$RHS_i = -\left[U_\infty(t) + u_W, W_\infty(t) + w_W\right]_i \cdot \bar{n}_i \tag{3.51}$$

where the free stream velocity $\bar{U}_\infty = [U_\infty(t), W_\infty(t)]$, being typically $W_\infty(t) = 0$; and u_W and w_W are the velocities induced by the known wake vortices, and can be found using equation (3.43).

Summing up, equation (3.47) can be rewritten to account for the wake development, taking the following shape for a collocation point i:

$$\sum_{j=1}^{n} a_{i,j}\Gamma_j + a_{i,W}\Gamma_{W_t} = -\left[U_\infty(t) + u_W, W_\infty(t) + w_W\right]_i \cdot \bar{n}_i \tag{3.52}$$

When all n collocation points are considered, a set of n of these equations is obtained for $n+1$ unknowns. Equation (3.49) provides the missing equation, resulting in a deterministic solution. The system can be rewritten in matrix form:

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} & a_{1W} \\ a_{21} & a_{22} & \cdots & a_{2n} & a_{2W} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & a_{nW} \\ 1 & 1 & \cdots & 1 & 1 \end{pmatrix} \begin{pmatrix} \Gamma_1 \\ \Gamma_2 \\ \vdots \\ \Gamma_n \\ \Gamma_{W_t} \end{pmatrix} = \begin{pmatrix} RHS_1 \\ RHS_2 \\ \vdots \\ RHS_n \\ \Gamma(t-\Delta t) \end{pmatrix} \tag{3.53}$$

where evidently the last row accounts for the Kelvin condition (3.49).

Solving this matrix provides the circulation of all vortices in the flow for one plate. The equivalent system of equations and matrix problem required to solve for several plates is developed in Appendix A, *Derivation of the matrix problem for N plates*.

At this point, the problem is solved: the flow field can be obtained and loads and forces can be found as presented in previous sections. The process in which the solution is being iterated in time, which involves displacing the free wake vortices in the flow, is explained in Section 3.3 *The MATLAB program*.

3.2.2 Modelling wind gusts

The perturbation that was chosen to excite the system of plates is the discrete wind gust. Wind gusts occur very frequently in the atmosphere and are typically modeled as a $1 - cos$ shape. They are standardized by, for instance, aviation authorities, as they pose a common scenario for certification requirements.

In the present case, the gust will be seen by the plates as a perturbation of the initial free-stream flow $U_{steady} \equiv U$ by an amount ΔU. The definition of the total velocity $U_\infty(t)$ including a gust starting at time $t = 0$ is as follows:

$$U_\infty(t) = \begin{cases} U_{steady} & \text{if } t < 0 \\ U_{steady} + \frac{\Delta U}{2}\left(1 - cos\left(\frac{2\pi t}{T_{gust}}\right)\right) & \text{if } 0 \le t \le T_{gust} \\ U_{steady} & \text{if } t > T_{gust} \end{cases} \tag{3.54}$$

where ΔU is defined as the maximum velocity perturbation and T_{gust} is the period of the gust. The use of this simplified gust model means that the velocity increment is seen at the same time in all the flow.

The values of T_{gust} to analyze are chosen according to the non-dimensionalization of the remaining time-related parameters. Non-dimensionalization and choice of parameters is explained in Chapter 4 *Procedure*. However, the value of the parameter ΔU is not directly related to any other parameter and it needs to be chosen carefully.

The main concern related to a choice of ΔU is to obtain real-world applicability. For this reason the atmospheric conditions near the ground were analyzed.

The lowest part of the planetary boundary layer is called the surface layer. In this layer the so-called friction velocity u_* is typically of the order of 0.3ms^{-1} [8]. It is also known that a typical velocity gradient will be in the order of $\Delta U \sim 3u_* \sim 1\text{ms}^{-1}$. Then, by knowing average values of U it will be possible to determine useful values of $\Delta U/U$.

The wind profile can be approximated by the following expression for distances around 0–10 m to the ground:

$$\bar{u} = \frac{u_*}{k} \ln\left(\frac{z}{z_0}\right) \tag{3.55}$$

where k is a the von Karman's constant, with an experimentally determined value of $k \approx 0.4$, and z_0 is the roughness length, which for grassy fields ranges in between 1–4 cm [8, pp. 129-130]. Evaluating this expression at a reasonable height for a practical application, such as solar panels located at a height of $h = 3\text{m}$, the results are:

Field	z_0	ΔU	$\bar{u}$	$\Delta U/\bar{u}$
Cut grass	1cm	1ms^{-1}	4.28ms^{-1}	0.23
High grass	4cm	1ms^{-1}	3.24ms^{-1}	0.31

So, for smooth fields, it is wise to consider a value of approximately $\Delta U/U \approx 0.2$ to account for common gusts in such a scenario.

The numerical model presented in this section together with the gust model provide the tools necessary to formulate a valid problem, that can be solved by a computer program, as presented in the next section.

3.3 The MATLAB program

This section introduces the MATLAB program that has been used for the analysis based on the concepts presented in Sections 3.1 *Theoretical Basis* and 3.2 *Numerical Model*.

3.3.1 Overview of the Program

The MATLAB code is a time-stepping, discrete-vortex potential-flow solver that has been developed based on the panel methods described in Section 3.2.1 *Computational Panel Method: Lumped-vortex Discretization*, following the guidelines set by Joseph Katz and Allen Plotkin [1, pp. 407-416].

Panel methods have been comprehensively described in Section 3.2.1. This unsteady method solves the problem freezing time at intervals, called time steps. It is based on the fact that the instantaneous solution is independent of time derivatives [1], and thus the flow can be solved at each step independently in a manner very similar to steady problems.

Physically, the main concern with unsteadiness is the formation of the wake, whose circulation can be found thanks to the Kelvin condition presented in Section 3.2.1. In addition, realistically the wake would not have any velocities normal to it [1], in other words, the wake must run parallel

to the local flow velocity. This is already acomplished as well, given that the wake vortices are displaced according to their local streamline at each timestep. The process of translating the wake vortices is called wake roll-up.

The program is structured according to the flow diagram shown in figure 3.15. It is composed of one main time loop, in which time develops discretized as time steps Δt, and some previous calculations that take place before the time loop, such as the initial steady state solver.

Geometry

The geometry has to be generated first. This includes processing the input parameters and discretizing the plates into vortices and collocation points in space. If the ground is to be considered, the program accounts for this and duplicates the geometry according to the mirror images method, as described in section *3.1.2 The ground effect*. An example of the geometry generated by the program for a case with two plates and ground effect is shown in figure 3.16.

The geometry of the plates is generated once on the fixed reference frame. Opposite to some models in which the flow remains stationary with respect to the inertial reference frame and the body moves [1], the best option to model a set of plates in ground effect seems to keep the airfoils at a fixed position and force the flow on this fixed frame of reference. This also reduces the need to keep track of inertia forces appearing in non-inertial reference frames.

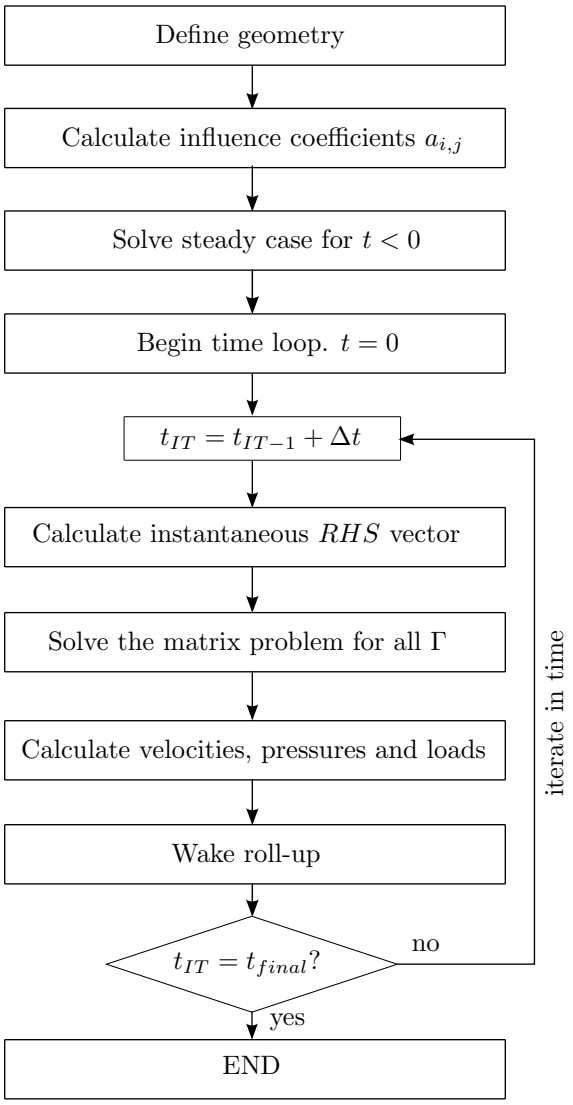


Figure 3.15: Program flowchart.

Solving the matrix problem

Next, the flow must be calculated using the method proposed in Section 3.2.1. First, the influence coefficients of the matrix that needs to be solved are found. Once the system matrix is known, the circulation of the vortices in steady state can be found. When analyzing gusts, the system will depart from a steady-state initial flow. The initial flow represents the initial conditions at $t < 0$ right before the gust appears. Once these conditions are known, the unsteady case can be solved starting from those initial conditions. Once the solution is known at each time step, eventually loads can be calculated.

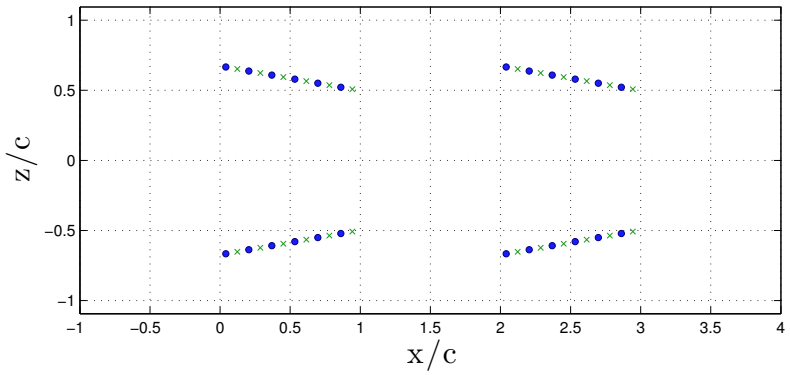


Figure 3.16: Discretization of a system of two flat plates with ground effect for 6 panels on each plate. Blue dots represent vortex elements and green crosses collocation points.

However, since the solver will typically deal with more than one plate, the system (3.53) must be modified to account for all plates, so that all vortices are affecting all collocation points of the system. In addition, since there will be as many wakes as airfoils, the Kelvin condition (3.49) for the conservation of total circulation of a system has to be accounted for once for each airfoil-wake system. With these modifications, the system (3.53) can be transformed to provide solutions for any number of plates. This process is described in Appendix A, *Derivation of the matrix problem for N plates*.

Calculation of loads

When the circulation problem has been solved, the fluid field is known everywhere. Velocities can be found and thus forces as well. The model for the calculation of loads in an unsteady problem is based on the guidelines set by J. Katz and A. Plotkin [1, pp. 414-415]. Instead of using directly the equations proposed by them, these were completed with other terms, accounting for the effect of the induced velocities of several other plates in both lift and drag.

The derivation of the forces is based on the Bernoulli equation for the calculation of pressures (3.29) presented in 3.1.2. The results for the lift L, the drag D and the moment $M_{y_{LE}}$ for each plate are:

$$L \equiv F_z = \sum_{j=1}^{n} \left(\rho(U_\infty + u_W + u_P)_j \Gamma_j + \Delta l_j \frac{\partial}{\partial t} \sum_{k=1}^{j} \Gamma_k \cos\alpha \right) \tag{3.56}$$

$$D \equiv F_x = -\sum_{j=1}^{n} \left(\rho(W_\infty + w_W + w_P)_j \Gamma_j + \Delta l_j \frac{\partial}{\partial t} \sum_{k=1}^{j} \Gamma_k \sin\alpha \right) \tag{3.57}$$

$$M_{y_{LE}} = \sum_{j=1}^{n} l_j \left(\rho(U_\infty + u_W + u_P)_j \Gamma_j \cos\alpha + \rho(W_\infty + w_W + w_P)_j \Gamma_j \sin\alpha + \Delta l_j \frac{\partial}{\partial t} \sum_{k=1}^{j} \Gamma_k \right) \tag{3.58}$$

where n is the number of panels in each plate, Δl_j is the length of one panel and l_j is the chordwise distance from the leading edge to the element j, as was defined in figure 3.9. The terms u_P and w_P represent the velocities induced by the remaining plates of the system. Γ_k are the values of the circulation of the vortices of the plate being considered. The steady term is consequence of the Kutta-Jouukowski theorem presented in Section 3.1.3.

Note that the last element in (3.56), (3.57) and (3.58) corresponds to the unsteady term in Bernoulli. Since the solution is not continuous but follows a time-stepping method, the time-derivative is discretized according to this expression

$$\left(\frac{\partial \Gamma_k}{\partial t}\right)_m = \frac{\Gamma_k^{m+1} - \Gamma_k^m}{\Delta t} \tag{3.59}$$

so that the the value of the derivative for the time step m is calculated at time step $m+1$. The calculation of forces is completely independent of the flow solution method that calculates Γ, and thus does not influence the flow solution.

These forces need to be non-dimensionalized for the analysis of results, a process explained in Chapter 4 *Procedure*.

Wake roll-up

After forces have been computed and before passing onto the next time step, the wake vortices must be moved along with the local stream velocity [1]. This is accomplished by calculating the velocity field at each location i of wake vortices and displacing the vortex elements the corresponding distance:

$$(\Delta x, \Delta z)_i = (u, w)_i \Delta t \tag{3.60}$$

Figure 3.17 shows the local velocities $\bar{v}_i$ of each wake vortex i. These are calculated for all required

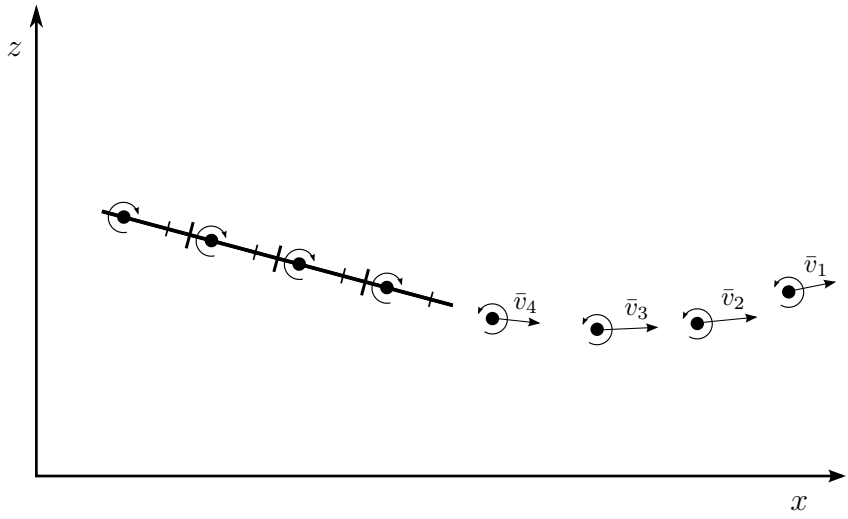


Figure 3.17: First step in wake roll-up consists of calculating local flow velocities.

positions first and then the wake elements are displaced according to (3.60) as illustrated in figure 3.18.

The wake roll-up is the latest step in the program before reiterating. With the description of the MATLAB program given in this section, it is hoped that the reader obtains a notion of how the solver works. The program has been carefully validated before carrying out the set of cases presented in Chapter *4 Procedure*. This validation is included in Appendix *B Validation of the code*. Additionally, the program code can be seen in Appendix *C The MATLAB source code*.

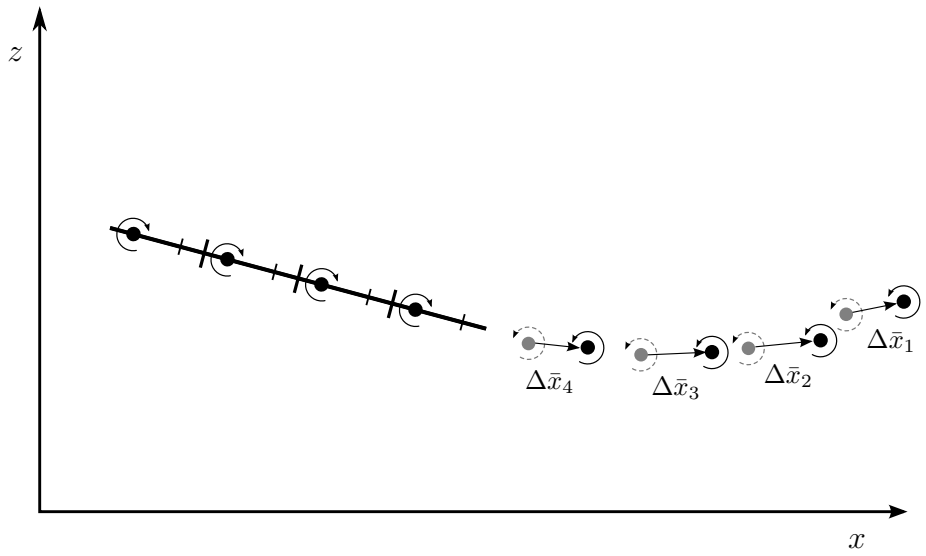


Figure 3.18: Second step in wake roll-up consists of displacing the wake elements.

3.3.2 Relevant Computational Parameters

There are a number of parameters that directly affect the problem resolution and the results. The most important are three: the number of vortices in each plate, the distance the new wake vortices are located from the trailing edge and the Courant–Friedrichs–Lewy (CFL) condition.

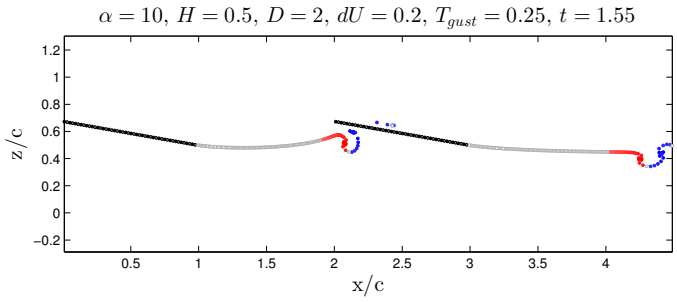
Number of vortices on each plate N_V

In order to discretize a flat plate into panel elements, it is necessary to fix the number of panels in which it will be divided. This number N_V represents the number of panels in each plate. The choice of this number is a trade-off between two main factors.

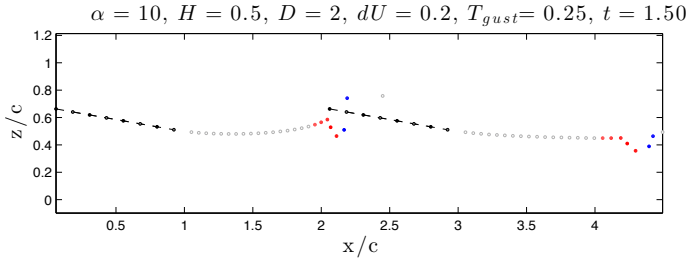
The first factor is accuracy. In principle, the higher N_V the higher the accuracy of the method. A higher number of panels will better approach the continuous distribution of vorticity of a flat panel that analytical solutions yield.

The second factor is the computational efficiency. More vortices mean more calculations, and this accumulation can result very time-consuming. In addition, N_V influences the time step Δt according to the expression (3.64), so also more iterations are required to reach a given final time t_{final} if N_V is increased.

This parameter was eventually chosen through validation tests. Not only these two factors influenced the final decision: erratic results were observed to appear and varied greatly with N_V. These were mainly due to undesired interactions of discrete vortices, wholly due to the discretization method. The final choice was $N_V = 24$ per plate, and the results and discussion leading to this choice are presented in Appendix B *Validation of the code*.



(a) A case being simulated with $N_V = 24$.



(b) Same simulation with $N_V = 4$.

Figure 3.19: Precision of the method depends directly on N_V.

New vortex shedding distance D_W

The necessity to drop vortices from the trailing edge at each time step in an unsteady problem has been presented in 3.2.1 *Computational Panel Method: Lumped-vortex Discretization*. The chosen distance D_W will affect the circulation of the plate, and it must be chosen carefully.

The distance D_W from the trailing edge that a vortex is placed at is measured along the path covered by the motion of the trailing edge, as shown in figure 3.20. Typical values of D_W range between 0.2–0.3$U_\infty(t)\Delta t$ according to J. Katz and A. Plotkin [1]. It is considered that wake vortices would be shedded approximately in the position where the trailing edge was located in the previous time step [1]. Since in the present case the airfoil is considered stationary and the flow is in motion, the newly shed wake vortex must be located along the path that the trailing edge would move along, if it were in motion at the free stream velocity $U_\infty(t)$ in a stationary fluid.

The 0.2–0.3 factor places the vortex closer to the current position of the trailing edge instead of at the exact position where the trailing edge was at the previous time step. This is so because placing the wake vortex in the middle of the interval defined by the translation of the trailing edge in a time step underestimates the induced velocities of the wake when compared to a continuous wake vortex result [1, p. 390].

Eventually D_W was fixed to $0.2U_\infty(t)\Delta t$ since it was found to be more accurate than $0.3U_\infty(t)\Delta t$ after validation tests. The validation results can be seen in Appendix B, *Validation of the code*.

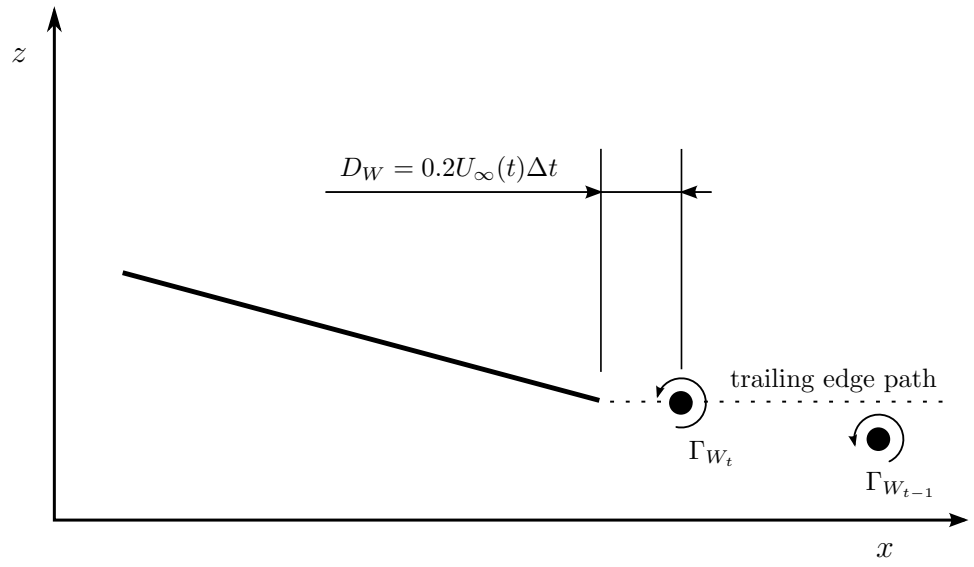


Figure 3.20: Shedding distance of new wake vortices.

Courant–Friedrichs–Lewy condition

The Courant–Friedrichs–Lewy (CFL) number establishes a necessary condition to have stability in the solution of a partial differential equation when using discrete methods. Therefore, for the numerical method employed in this project, it is required to consider this factor.

The concept behind the CFL number is to be able to observe all phenomena happening in the flow at the lowest scale, which is the panel length. In other words, if a fluid particle is able to travel more than one panel domain in one time step, then all interactions happening between that skipped element and the fluid particle are not being accounted for [9, p. 70]. This is avoided by guaranteeing that the discretization of time allows for all these interactions to occur.

We define the characteristic length to be the panel chord length:

$$c_{panel} = \frac{c_{airfoil}}{N_V} \tag{3.61}$$

so that the CFL number can be defined as [9]:

$$\text{CFL} = \frac{U_\infty \Delta t}{c_{panel}} = \frac{U_\infty \Delta t}{c_{airfoil}/N_V} \tag{3.62}$$

The CFL number represents the dimensionless ratio of the distance traveled by a fluid element $U_\infty \Delta t$ to the characteristic length c_{panel}. Since the distance travelled by fluid particles is desired to be smaller than the panel domain, this condition can be written as:

$$\text{CFL} = \frac{U_\infty \Delta t}{c_{airfoil}/N_V} \leq 1 \tag{3.63}$$

In principle, for any CFL satisfying (3.63) the solution will faithfully represent the flow. The value of CFL eventually chosen for the numeric solver was CFL = 0.25. A participating factor

in this choice was the gust profile, because the CFL condition determines the time step in the following way,

$$\Delta t = \frac{\text{CFL } c_{airfoil}}{U_\infty N_V}, \tag{3.64}$$

so that, for some common values like $U_\infty = 1\text{ms}^{-1}$, $c_{airfoil} = 1\text{m}$ and $N_V = 24$ the time step is $\Delta t = 1/96\text{s} = 0.010\text{s}$. This time step is sufficiently small for both complying with the CFL condition as well as providing a fine discretization for the shortest type of gust; for example, one with period $T_{gust} = 0.25\text{s}$, which is the lowest to be considered, can be discretized into 24 time steps at this value of Δt. The reasons why $T_{gust} = 0.25$ is the shortest gust considered are explained in Chapter *4*. Figure 3.21 shows the gust model for these parameters, visibly smooth for the fastest case of $T_{gust} = 0.25$.

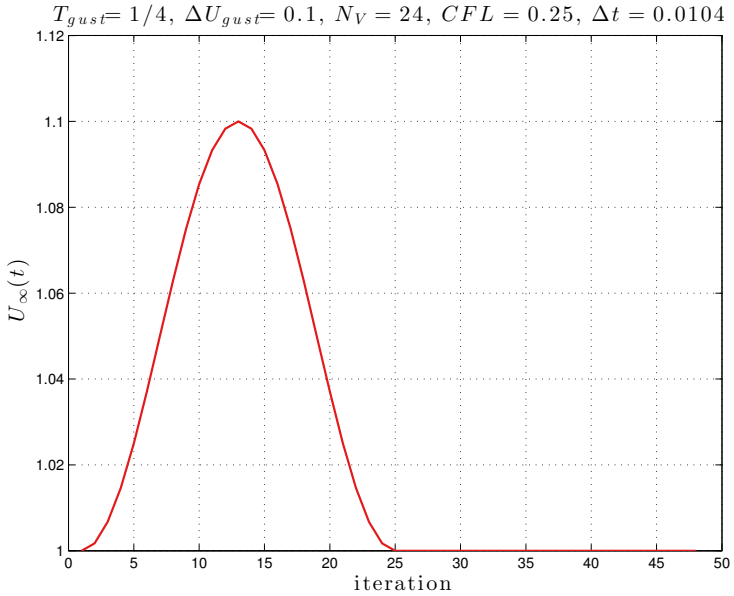


Figure 3.21: Discretized wind profile shape for the shortest possible gust.

3.3.3 Code performance

Code performance is essential if it is intended to process several cases. The program has been developed in MATLAB, which is a programming language oriented towards mathematical and matrix operations. The code was optimized as much as possible using practices recommended by The MathWorks [10], for example vectorizing loop operations. The full code is available in Appendix C *The MATLAB source code*.

The final version of the program would be required to run hundreds of cases. Depending on whether graphical output was required or not, the computation time would vary. Most of the time the code was executed without graphics output. The code would use the maximum capacity of up to one processor core. In multi-core computers, the total computational time could be reduced by launching several MATLAB instances and running different cases at the same time. For this purpose, the script in charge of sending cases to the solver simply needs to keep record of which

cases are being run by other instances at the same time, which can be accomplished simply by marking which cases are being processed by preallocating their space for the data output.

Considering an average case with 4 plates (2 real plates in ground effect), $N_V = 24$ panels per plate, $CFL = 0.25$ and final time $t_{final} = 10$, the total number of iterations is 960. On a modern i5 processor, with enough RAM memory to avoid the use of hard drive memory paging, this particular case would take approximately 10 minutes.

Chapter 4

Procedure

4.1 Configuration of parameters

Once the computational method is ready, in order to carry out any analysis it is necessary to secure the range of parameters that will be tested.

Nondimensionalization

As described in previous chapters, the solution is affected by geometric parameters, gust parameters and computational parameters. Computational parameters were set to the most appropriate values after validation of the program (See Appendix B *Validation of the code*). Geometric parameters were described in Section 2.1 *Geometry of the problem*, and wind gusts were defined in Section 3.2.2 *Modelling wind gusts*.

These parameters are recalled next. Let's remember that the airfoil considered is a flat plate, modelled as a zero-thickness plate.

$\boldsymbol{c}$	The airfoil chord.
$\boldsymbol{D}$	Separation between airfoils' leading edges.
$\boldsymbol{H}$	Distance from the ground to the airfoil's trailing edge.
$\boldsymbol{\alpha}$	Angle of attack of the airfoil.
$\boldsymbol{U}$	Undisturbed free stream velocity.
$\boldsymbol{\Delta U}$	Maximum velocity perturbation due to gust.
$\boldsymbol{t}$	Time.
$\boldsymbol{T_{gust}}$	Period of the gust.
$\boldsymbol{U_\infty(t)}$	Total free stream velocity. Depends on $\boldsymbol{T_{gust}}$, $\boldsymbol{U}$ and $\boldsymbol{\Delta U}$ as defined in (3.54).

These parameters are all but α measured in different units of length and time. In order to keep the scope of this research as wide as possible, it is wise not to focus solely on a particular set or dimensions, say those applicable to open roofs measuring 10 meters in span. By means of nondimensionalization, it is possible to get rid of the dimensional constrains imposed by a particular case and parametrize the problem so that it covers a wider set of scenarios.

Nondimensionalization implies normalizing all quantities with respect to a reference value. This yields dimensionless parameters, whose dimensionless results can be together transformed back into dimensional quantities by using the equivalent reference value. Let's clarify the topic by nondimensionalizing these parameters and setting an example.

We define the characteristic length as the chord length c. The $\sim$ symbol represents nondimensional quantities. The parameters of length reduce to:

$$\tilde{c} = \frac{c}{c} = 1 \tag{4.1}$$

$$\tilde{H} = \frac{H}{c} \tag{4.2}$$

$$\tilde{D} = \frac{D}{c} \tag{4.3}$$

The characteristic velocity is defined as the undisturbed free stream velocity, U. Again this variable becomes unity:

$$\tilde{U} = \frac{U}{U} = 1 \tag{4.4}$$

Also, the velocity perturbation due to wind gust becomes

$$\widetilde{\Delta U} = \frac{\Delta U}{U} \tag{4.5}$$

and since the total free stream U_∞ solely depends on U and ΔU, when set as a function of $\tilde{U}$ and $\widetilde{\Delta U}$ it automatically becomes a nondimensional quantity we shall call $\tilde{U}_\infty$.

Now the remaining parameters with units of time are considered. The time of the system can be nondimensionalized as

$$\tilde{t} = \frac{t}{c/U} \tag{4.6}$$

and the period of the gust as

$$\tilde{T}_{gust} = \frac{T_{gust}}{c/U} \tag{4.7}$$

Now all our parameters are nondimensional. This means that the results will not depend on any dimension: they will be able to adapt to any real value of c and U, which are the reference magnitudes.

For instance, let's consider a problem A with known solution, whose parameters are $\tilde{H} = 1$, $\tilde{D} = 4$, $\widetilde{\Delta U} = 0.2$ and $\tilde{T}_{gust} = 1$. Then, this result would be valid for a real-world case where $c = 0.5$m, $H = 0.5$m, $D = 2$m, $U = 2$m/s and $T_{gust} = 0.25$s. But, it would also be valid for a real problem with parameters $c = 3$m, $H = 3$m, $D = 12$m, $U = 1$m/s and $T_{gust} = 0.33$s.

In conclusion, it is much easier to extrapolate results to different cases when the problem is solved with dimensionless parameters. However, as it was seen with the previous example A, one nondimensional case provides the capability of *stretching* or *compressing* the geometry of a problem. It does not allow for different ratios between those dimensions. It is thus necessary to analyze several cases, which should cover a wide range of different relations between the intervening parameters.

The advantages of nondimensionalization are clear: solutions do not apply to unique cases with fixed dimensions, instead they apply to all real cases whose dimensionless parameters agree with those of the solution. The next step is then to establish what ratios will be more common and are thus worth analyzing.

Choice of dimensionless parameters

The first parameter will be the angle of attack α of the plates. This is by itself dimensionless. It was determined through testing that $\alpha = 10°$ was appropriate to observe all interactions related to steady lift and unsteady gusts. In order to check the effect of other configurations of α, the inverted airfoil $\alpha = -10°$ was considered too. The range of α analyzed is then

$$\alpha = [10°, -10°] = \left[\frac{\pi}{18}, -\frac{\pi}{18}\right] \tag{4.8}$$

For the remaining geometric parameters it was determined that typical situations where ground would have an effect on the system would be, undeniably, close to the ground. So the range of values on which $\tilde{H}$ will be considered are mostly distances around the chord length:

$$\tilde{H} = [0.5, 1, 2, \infty] \tag{4.9}$$

where ∞ represents no ground.

For the distance between plates, it was possible to consider longer relative distances, but enormous ones would probably lose most of the interaction. The selected final range is

$$\tilde{D} = [2, 4, 8, \infty] \tag{4.10}$$

where ∞ is effectively a case with a single plate.

The velocity perturbation due to wind $\widetilde{\Delta U}$ was already determined in section *3.2.2 Modelling wind gusts* to have typically values of 0.2 in the lowest part of the planetary boundary layer, the surface layer. Moreover, one extra value was tested as well, in order to be able to observe the effects of varying this parameter. The final range is

$$\widetilde{\Delta U} = [0.1, 0.2] \tag{4.11}$$

Next are the time-dependent parameters. The time of the system is invariable, since the time t that a particle will take to travel a distance c at velocity U will always be the same, c/U time. But the variable $\tilde{T}_{gust}$ has to be chosen.

Variable $\tilde{T}_{gust}$ will register how fast the gust occurs with respect to a characteristic time c/U. Reasonable values, in line with previous choices, would be in the range of [0.5–2]. However, there are other important characteristic times that relate the fluid velocity to other lengths, such as H or D, that can be considered too. Thus, if we want all cases to consider a variation of at least [0.5–2] with respect to all these characteristic times, it is necessary to verify these three ranges:

$$\frac{T_{gust}}{c/U} \equiv \tilde{T}_{gust} \in [0.5\text{–}2] \tag{4.12}$$

$$\frac{T_{gust}}{H/U} \in [0.5\text{–}2] \tag{4.13}$$

$$\frac{T_{gust}}{D/U} \in [0.5\text{–}2] \tag{4.14}$$

Ranges (4.13) and (4.14) can take use of previous definitions to be rewritten with respect to the main characteristic time c/U in the following way:

$$\frac{T_{gust}}{H/U} = \frac{T_{gust}}{c/U}\frac{c}{H} \longrightarrow [\tilde{T}_{gust}] \in [\frac{T_{gust}}{H/U}] \cdot [\tilde{H}] = [0.5\text{–}2] \cdot [0.5\text{–}2] = [0.25\text{–}4] \tag{4.15}$$

$$\frac{T_{gust}}{D/U} = \frac{T_{gust}}{c/U}\frac{c}{D} \longrightarrow [\tilde{T}_{gust}] \in [\frac{T_{gust}}{D/U}] \cdot [\tilde{D}] = [0.5\text{–}2] \cdot [2\text{–}8] = [1\text{–}16] \tag{4.16}$$

Eventually it is possible to summarize all desired ranges to be analyzed into a single range of $\tilde{T}_{gust}$, which is:

$$\tilde{T}_{gust} \in [0.25\text{–}16] \tag{4.17}$$

From this range a set of discrete values must be chosen for analysis. These have been:

$$\tilde{T}_{gust} = [0.25, 1, 4, 16] \tag{4.18}$$

For the sake of simplicity, all throughout the following chapters the $\sim$ symbol will be dropped. Therefore all variables without dimensions will always refer to the dimensionless parameters that have been defined in this chapter, so that for instance $\tilde{T}_{gust}$ will be represented as simply T_{gust} without units.

4.2 Coefficients of forces

It is proper to mention in this chapter that the aerodynamic loads are also nondimensionalized. This is typical in aerodynamics, since these coefficients display the capability of an airfoil to produce a force, with respect to the fluid properties, rather than the current force for a specific situation. These nondimensional quantities are extremely common and are presented here as a reference.

The lift and drag forces are nondimensionalized by using the dynamic pressure $q = \frac{1}{2}\rho v^2$ and the airfoil planform area S. For 2D cases the chord c is used instead of the planform. The corresponding dimensionless quantities are the lift coefficient C_L and the drag coefficient C_D:

$$C_L = \frac{L}{\frac{1}{2}\rho v^2 c} \tag{4.19}$$

$$C_D = \frac{D}{\frac{1}{2}\rho v^2 c} \tag{4.20}$$

The coefficient of moment with respect to the leading edge $C_{M_{LE}}$ takes a form equivalent to L and D with an extra length dimension, in this case the chord c:

$$C_{M_{LE}} = \frac{M_{y_{LE}}}{\frac{1}{2}\rho v^2 c^2} \tag{4.21}$$

4.3 Computational Setup

At this point, all dimensionless parameters affecting the solution have been chosen. These are presented in ranges (4.8), (4.9), (4.10), (4.11) and (4.18). Together, they offer a set of $2 \times 4 \times 4 \times 2 \times 4 = 256$ possible combinations. These 256 cases have been computed to obtain a data base from which conclusions can be extracted.

The solver is a MATLAB program that accepts any input parameters. Another program was written to run all 256 cases through the solver and store the results in files. These files can then be read by another code in charge of presenting results, be it in a numerical or in a visual manner.

Chapter 5

Results

Data was gathered for the 256 cases proposed in Chapter 4 *Procedure*. These data need to be analyzed so to obtain meaningful information out of the simulation.

Since the amount of data was so large, the approach taken was to compare small pieces of information at a time. However, before focusing on the influence of the different parameters, we will introduce the common responses that are observed in all cases.

5.1 Wake-plate interaction

The dominant effect in our two-plate system was the influence on the trailing plate of the first plate's wake vorticity, created during the gust encounter. This fact only happens in the free stream direction, so that the first plate is virtually unaffected by the unsteady response of the trailing plate.

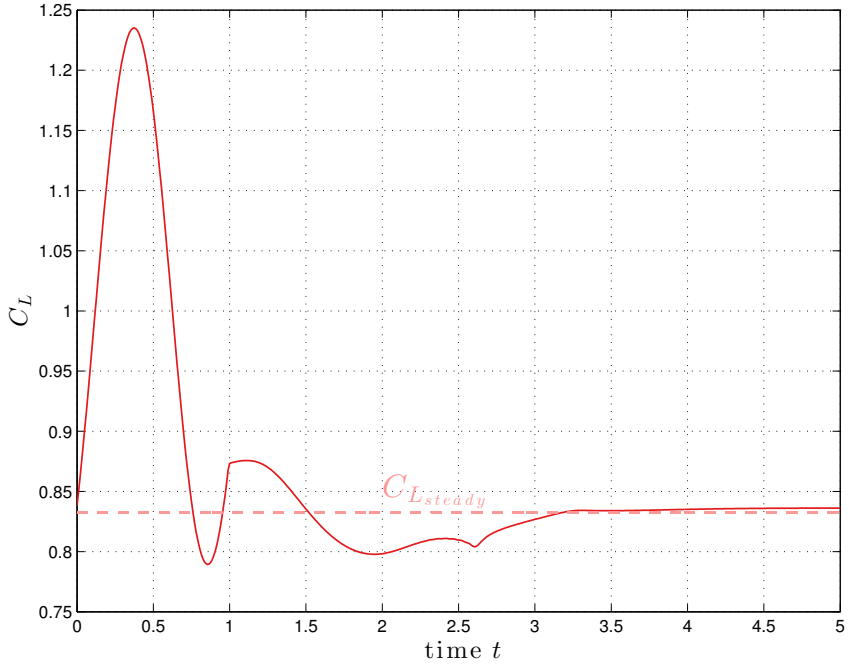
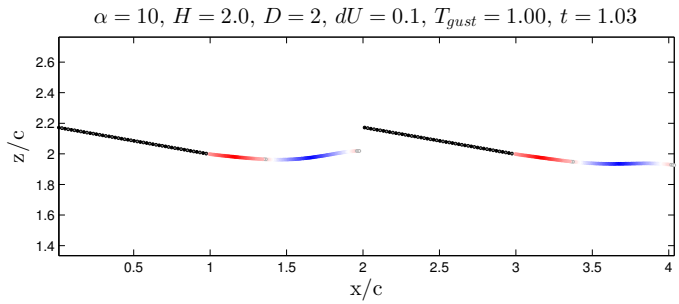


Figure 5.1: C_L of the trailing plate for $\alpha = 10°$, $H = 2$, $D = 2$, $\Delta U = 0.1$, $T_{gust} = 1$

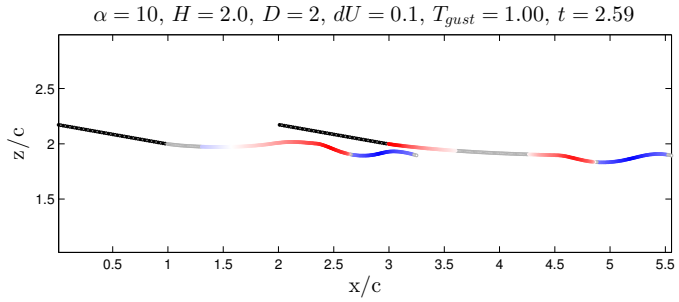
Figure 5.1 shows the C_L in the trailing edge of a particular case. It is visible that after the main spike due to the wind gust, there exists an increase in lift, followed by a noticeable decrease with respect to the steady value before eventually the force comes back to the value previous to the gust.

This effect is attributable to the wake vortices. By looking at figure 5.2, it is possible to understand the reasons behind this. First of all, it must be noted that counterclockwise (CCW) vorticity (negative vorticity) is represented in color blue in the wake. Clockwise (CW) vorticity (positive vorticity, such as the one that airfoil panels produce to generate lift) is represented in color red in the wake.

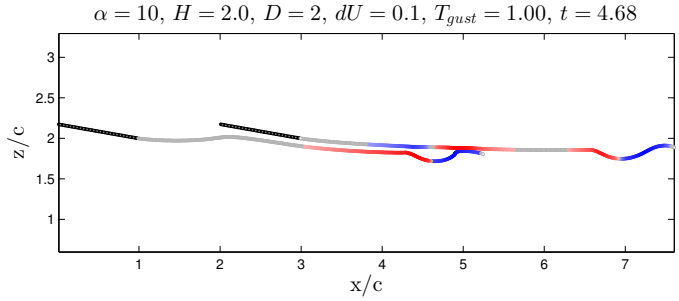
When an airfoil accelerates, its circulation Γ increases. In order for the Kelvin condition, presented in page 23, to hold, same-intensity but opposite-sign wake vortices are shedded.



(a) Vortices upstream of second plate



(b) CCW vortex past the trailing edge of the second plate



(c) Wake vorticity past the second plate

Figure 5.2: Effect of leading airfoil's wake on trailing airfoil.

When the blue CCW vortex, created when the first plate accelerated due to the gust, approaches the second plate, it induces an upwash that increases the lift. Then, in the second image, as this CCW vortex gets past the second plate it produces the contrary effect, a downwash on the trailing plate that reduces the lift. In addition, the red CW vortex, shedded when the first plate decelerated back to normal velocity, is also producing a downwash when approaching the second plate.

Lastly, these two strong vortices are past the second airfoil, and their effects dissipate. One could expect the latest CW vortex to induce also an upwash upon leaving the trailing edge of the plate, and in some cases a small bump in lift is appreciated, but it is neutralized by the not-so-far CCW vortex. In this particular case this last lift bump is very lightly appreciated in figure 5.1.

Another visible effect is the interference between plates in the horizontal force. This applies for the steady state as well and is visible in the drag coefficient C_D. Due to velocities induced by each plate on the other one, the first plate experiments a reduction in drag while the trailing one sees its drag increased by the same magnitude.

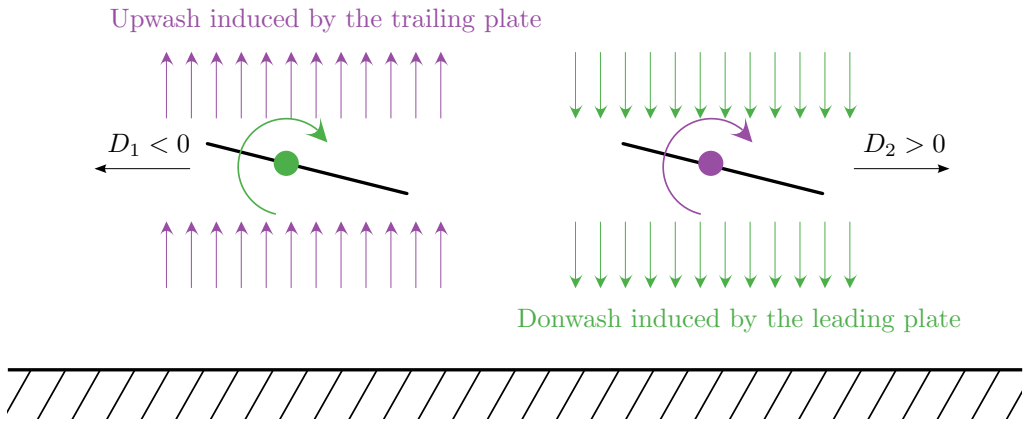


Figure 5.3: Downwash and upwash mutually induced by the plates generate horizontal forces.

As a whole, the system has zero net horizontal force, agreeing with the paradox of D'Alembert (which states that in potential flow the drag of a body is zero) but each plate effectively sees a horizontal force of opposite direction. Figure 5.3 represents this effect. Even though the addition of both plate's drags is zero, as expected in potential flow, their interaction produces that the leading plate sees a reduction in drag while the trailing plate observes an increase in drag.

This effect also happens when the plates are flying inverted, i.e., $\alpha = -10$. Even though the upwash and downwash are exchanged, the circulation of both plates has opposite sign as well. After all, the induced horizontal forces remain of the same direction. In conclusion, the leading plate always sees a drag reduction while the trailing one sees an increase in drag.

Figure 5.4 illustrates in further detail the direction of the forces relative to the velocity seen by the airfoil.

It must be noted that several parameters affect this interference. For example, the ground effect provides an obstacle to both vertical induced velocities, and thus ground proximity inhibits this effect. More information on the influence of each parameter is given in the following section, where quantitative results of this induced drag are also shown.

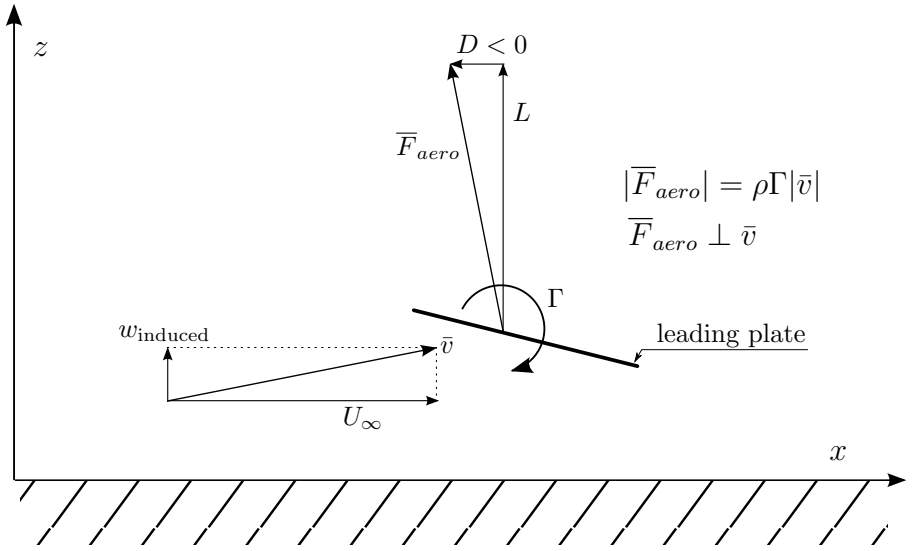


Figure 5.4: The upwash induced by the trailing plate w_{induced} generates a "negative drag" on the leading plate.

With respect to the influence on the steady state lift that another airfoil has, the leading plate has greater lift while the trailing plate has a lower lift than a single plate system. The closer the two plates are, the more intense this effect is, as is discussed below, in the section dedicated to the influence of the separation between plates D.

5.2 Influence of individual parameters

Influence of the angle of attack α

A positive angle of attack $\alpha = 10°$ shows around 2% larger increase in lift due to gust than $\alpha = -10°$ when there is ground. It is unclear whether this is related to the ground effect or not. Since the geometry is not totally symmetric because the ground distance to the different parts of the airfoil varies, this small variation may as well be attributed to the different geometrical shape of the system with inverted airfoils.

Inverting the airfoil so that $\alpha = -10°$ does not change other effects of the interaction between plates that have been described before. Their C_M are opposite in sign though, as expected.

Influence of the ground distance H

The ground effect has a great importance on the steady forces. Figure 5.5 presents the drag for both airfoils in different ground distances. The closer to the ground the less importance the velocities induced by the plates among themselves have. The ground offers damping to vertical induced velocities, which are restricted. The case $H = \infty$, representing no ground, is seen to offer the greatest variation in C_D due to the effects of the induced velocities discussed before.

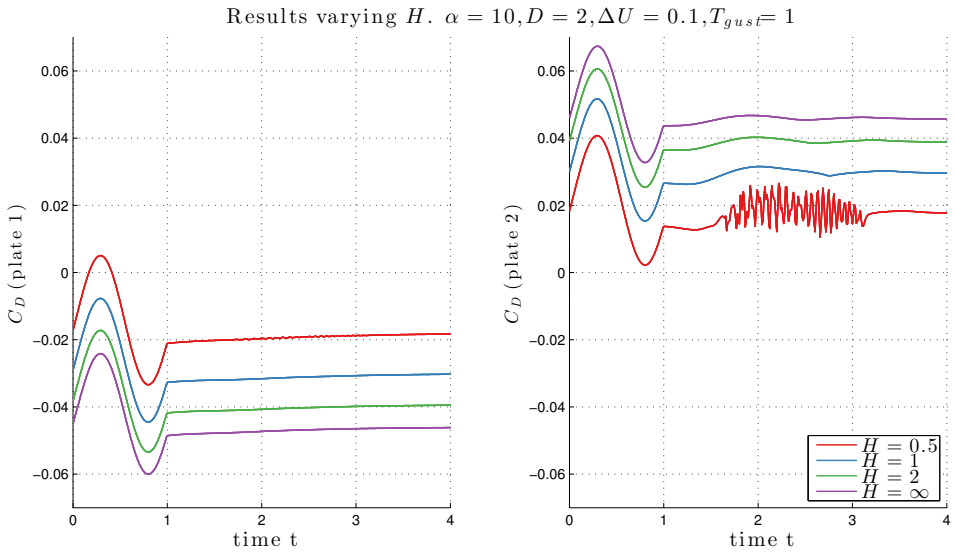


Figure 5.5: C_D of both plates for different values of H. Note the discretization-induced oscillations on the right hand side plot: these are analyzed in Appendix B *Validation of the code*.

With respect to the steady lift, the following table presents some values for different heights H, for $D = 2$ and $\alpha = 10°$.

	$H = 0.5$	$H = 1$	$H = 2$	$H = \infty$
C_L(plate 1)	1.1596	1.2108	1.2706	1.3619
C_L(plate 2)	0.9934	0.9001	0.8326	0.8145
C_D(plate 1)	-0.0177	-0.0295	-0.0387	-0.0455
C_D(plate 2)	0.0177	0.0295	0.0387	0.0455

It is visible that, in the same fashion that the induced C_D values are larger the further from the ground, the effect on the lift of the velocities induced by the plates among themselves is also greater the further the distance from the ground.

Figure 5.6 presents the unsteady response in lift of the trailing plate for several heights. The closer to the ground, the more visible a damping in the transient C_L is, even though still of very low magnitude, in the order of a 2% variation in the maximum C_L among different heights, for this particular case and for all ranges of H from 0.5 to ∞.

Transient results practically do not vary because of the ground. It needs to be recalled that two-dimensional ground effect is very different from a true three dimensional ground effect, as explained in Section 3.1.2 *The ground effect* on page 14.

The case $H = 0.5$ is also the more prone to unwanted oscillations. These are due to the numerical method, as discussed in Appendix B *Validation of the code*, and are probably induced because the wake is being kept in line with the plates, and the wake vortices get too close to the plate vortices and interact in unwanted ways.

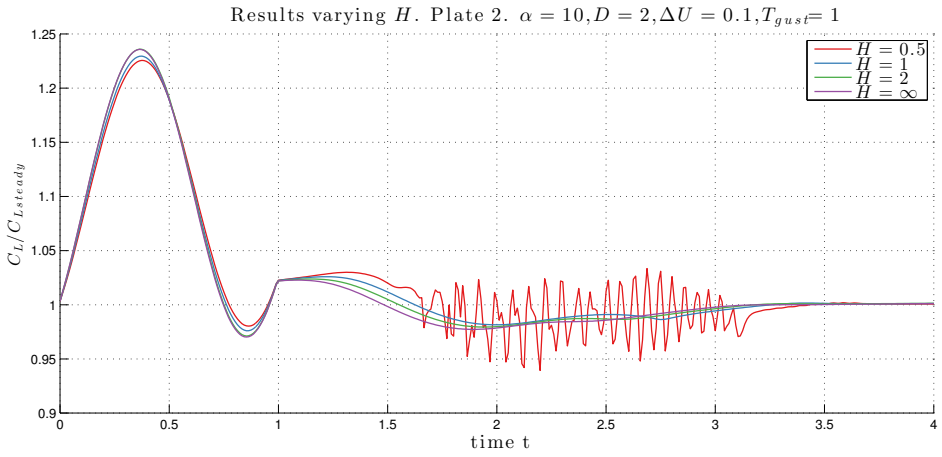


Figure 5.6: $C_L/C_{L_{steady}}$ of the trailing plate for different values of H.

Influence of the plates separation D

The distance D greatly influences the interaction between the plates in a steady flow as well. The closer the plates are, the greater their influence becomes. C_D, as discussed earlier, is reduced for the leading plate while it is increased for the trailing plate, as represented in figure 5.7. In this figure, the case $D = \infty$, which represents a single plate, is seen to reach the steady value of $C_D = 0$ expected in potential flow for a single airfoil.

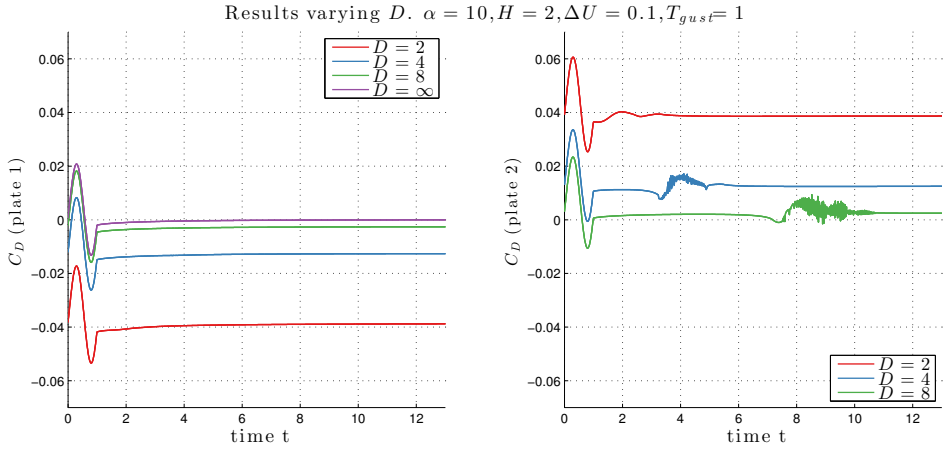


Figure 5.7: C_D of both plates for different values of D.

As introduced earlier on, the steady state lift is affected in a similar manner as C_D. The leading plate has greater lift while the trailing plate has a lower lift than a single plate system. The closer the two plates are, the more intense this effect is, because of the increased circulation on the leading airfoil with respect to a decrease circulation in the trailing one. This is due to the induced velocities discussed before, which effectively change the angle of attack of the free

stream flow as seen by the airfoils. The following table shows the steady values of C_L and C_D for a no-ground configuration for different plate separation distances D, with $\alpha = 10°$.

	$D = \infty$	$D = 2$	$D = 4$
C_L(plate 1)	1.0911	1.3619	1.2255
C_L(plate 2)	1.0911	0.8145	0.9555
C_D(plate 1)	0	-0.0455	-0.0235
C_D(plate 2)	0	0.0455	0.0235

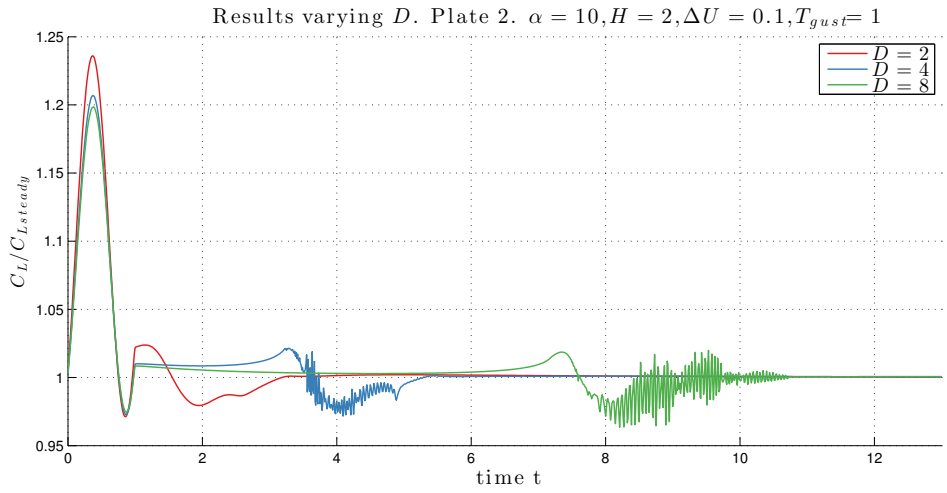


Figure 5.8: $C_L/C_{L_{steady}}$ of the trailing plate for different values of D.

The lift variation induced by a gust is shown in figure 5.8. A larger separation distance D only delays the onset of the plate-wake transient effect on the trailing plate. Its amplitude is practically the same.

Regarding the discretization-induced oscillations seen in figures 5.7 and 5.8, the further apart the plates are, the easier for undesired oscillations to develop. This is probably produced by the wake sheet encountering the second plate on its leading edge in the same line, due to the ground preventing it from moving. On the other hand, for shorter distances, the wake is displaced more by the closer plate vortices and encounters the trailing plate at a more reasonable distance. Anyhow, this behavior is solely caused by the numerical method, and does not represent any true physical interaction.

Influence of the gust maximum velocity increment ΔU

Variations in aerodynamic forces are proportional to the gust intensity. Figure 5.9 represents the response in lift of the trailing plate for two values of ΔU.

It must also be noted that in some cases, $\Delta U = 0.2$ would induce undesired oscillations while a value of 0.1 would not. This is entirely attributable to the numerical method though, and is probably caused by the higher-intensity wake vortices released under the stronger gust.

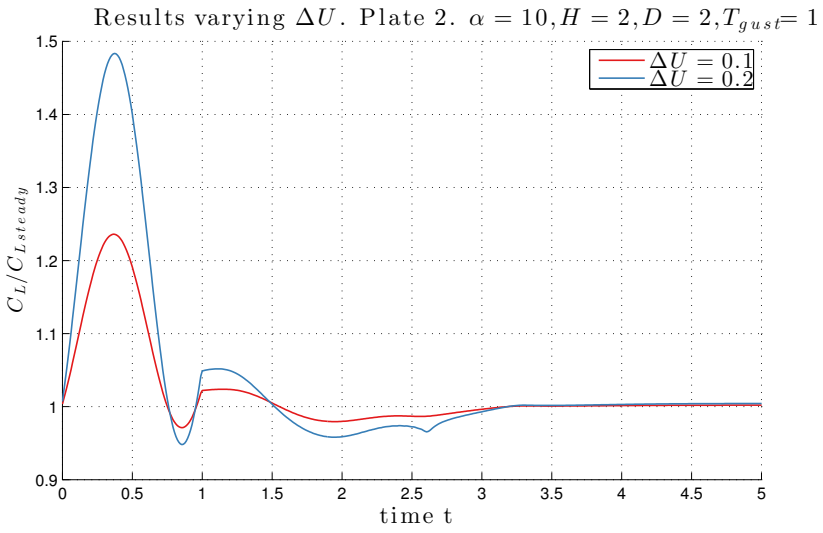


Figure 5.9: $C_L/C_{L_{steady}}$ of the trailing plate for different values of ΔU.

Influence of the gust period T_{gust}

Very quick gusts, such as $T_{gust} = 0.25$, do produce a considerable transient increase in the aerodynamic forces when compared to slower gusts. This is attributable to the unsteady terms of the lift expression (3.56), whose contribution is more visible for the fastest gusts. For slower gusts, the increase in lift is proportional to ΔU. Those cases can be considered as quasi-steady cases, specially for the slowest gusts, like $T_{gust} = 16$.

Figure 5.10 presents some values of C_L for different gust periods. For $T_{gust} = 0.25$ the C_L increases around an 80%, while for other slower periods this increase goes down to about 40%.

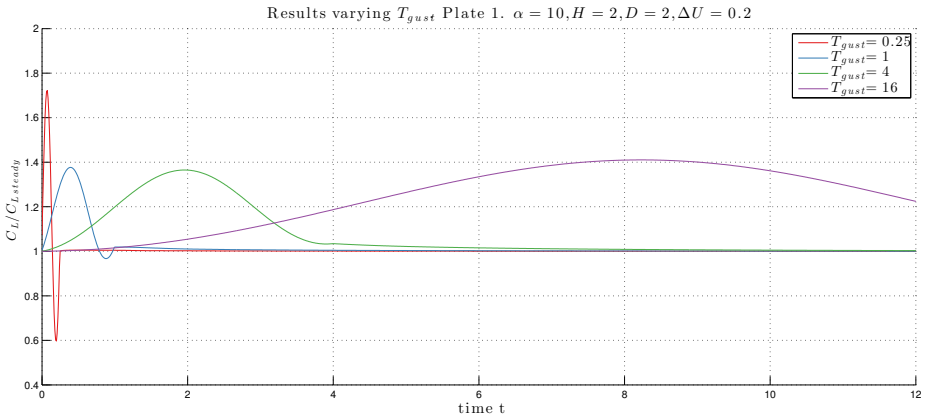


Figure 5.10: $C_L/C_{L_{steady}}$ of the leading plate for different values of T_{gust}.

Chapter 6

Application

In this chapter a plausible application in the design of solar panels is proposed.

A solar panel field constructor may be interested in any aerodynamic forces that may influence the design of the structures. Wind will blow over the panels, which need to be placed as close together as possible at a height from the ground. The ground effect will have an effect in their aerodynamic behavior.

From the structural point of view, the biggest concern will be any possible self-induced oscillations due to aerodynamic forces. Under some conditions, unstable flows can form, for instance due to leading edge separation at very steep angles and a consequent oscillation caused by the leading and trailing edges of an airfoil, which will be shedding vorticity on both sides. However, our model does not account for leading edge separation, so self-induced oscillations are not observed. A possible development to the code would be to incorporate the possibility of leading edge separation, as it has been done for a similar panel code by J. Katz [2].

Still, there lies great interest in the response to gusts of the structure, which may be affected by high or repetitive loads, which are a concern specially for fatigue of materials.

For this preliminary analysis, the data gathered in the computations proposed in Chapter 4 *Procedure* will be used. The gust will be fixed at the highest intensity of $\Delta U = 0.2$, which is appropriate for atmospheric flows near the ground, as explained in Section 3.2.2 *Modelling wind gusts*. The focus of the analysis is on the trailing plate, which sees repetitive forces due to the gust itself and the forces induced by the wake sheet produced by leading airfoils.

The analysis will focus on two aspects: the maximum force that is attained for a given gust and the time it takes for the forces to dissipate. This will be observed in different designs, so to find the most appropriate one. The design variable that will be analyzed is the height from the ground. The angle of attack is kept constant at the positive value of $\alpha = 10°$ and gust periods $T_{gust} = 0.25, 1$ are compared.

The separation between plates D has not been considered for several reasons. First, it is assumed that solar panels need to be built as close as possible. Secondly, the distance between plates would negatively affect the response analysis of the trailing plate, as a larger distance D directly affects the time it takes for the perturbations to arrive. This would introduce inaccurate results as a consequence of comparing geometries that, from the beginning, cannot be compared. For these reasons, D has been fixed to the shortest distance $D = 2$.

6.1 Lift forces

In order to compare the magnitude of the forces, the largest value $C_{L_{MAX}}$ and the second largest value C_{L_2} have been considered. These have been plotted in figure 6.1 for several ground distances H and for the most representative transient gusts, which are $T_{gust} = 0.25$ and $T_{gust} = 1$. Any longer period of the gust makes the problem practically a quasi-steady one.

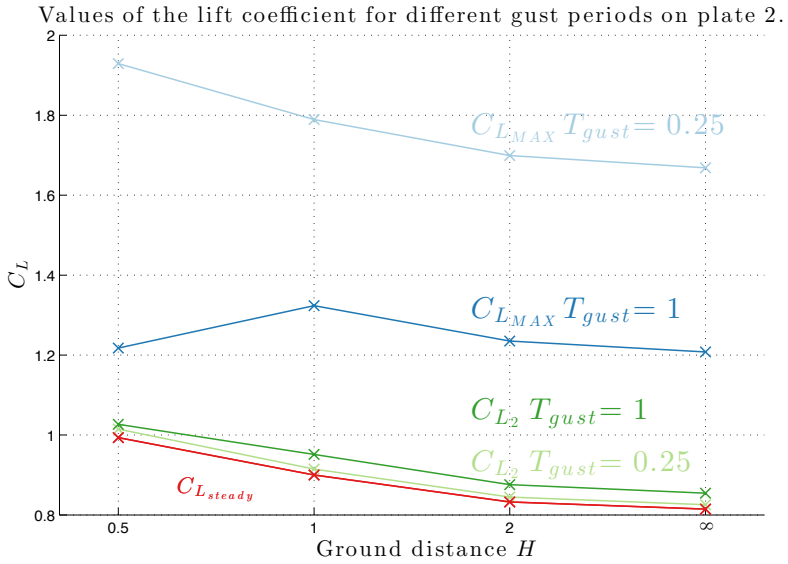


Figure 6.1: Comparison of the peak lift forces for different values of H and T_{gust}.

Figure 6.1 shows the relative strength of the peak in lift due to the gust itself ($C_{L_{MAX}}$) and that caused by the wake of the leading airfoil C_{L_2}, which is practically equal to the steady value of C_L. It needs to be noted that ($C_{L_{MAX}}$) is effectively higher for the shortest gust, as discussed in Chapter 5 *Results*. However, the second peak in the lift force, which is caused by the interaction with the leading plate's wake, is larger for the slower gust. Figure 6.2 of the next section has an illustration of the relative importance of these two maxima in C_L.

Considering the worst case scenario, for $T_{gust} = 0.25$ and $H = 0.5m$, $C_{L_{MAX}} = 1.93$. Assuming a steady wind speed of 8ms^{-1} ($\approx$ 29kmh) and a chord length of 4m, the gust would have a period of $T_{gust} = 0.125$s and the free velocity maximum due to gust would be $U_{\max} = U + \Delta U = 8ms^{-1} + 1.6ms^{-1} = 9.6ms^{-1}$. Then the equivalent force would be, according to (4.19):

$$F_{z_{\max}} \equiv L_{max} = C_{L_{\max}} \frac{1}{2} \rho U c = 1.93 \times (0.5 \times 1.225 \times 8 \times 4) = 37.83N \tag{6.1}$$

where $\rho = 1.225$kg m^{-3} at sea level [6]. This is a noticeable increase with respect to the steady lift, which is calculated in the same manner to be, for the particular case of $H = 0.5$, where $C_{L_{steady}} = 0.99$, equal to $L_{steady} = 19.4N$.

6.2 Response time

To evaluate the dissipation of gust-induced forces on the system, we need to compare the different responses of different geometries and gust cases. Again, only H and T_{gust} will be compared. Remaining parameters are maintained at $\alpha = 10°$, $D = 2$ and $\Delta U = 0.2$.

For this process, the response in C_L has been fitted to an exponential function of the kind

$$C_L = Ae^{bt}, \tag{6.2}$$

where b is the exponential decay. The exponential decay symbolizes how fast the system can damp the forces induced by the gusts. An example is presented in figure 6.2.

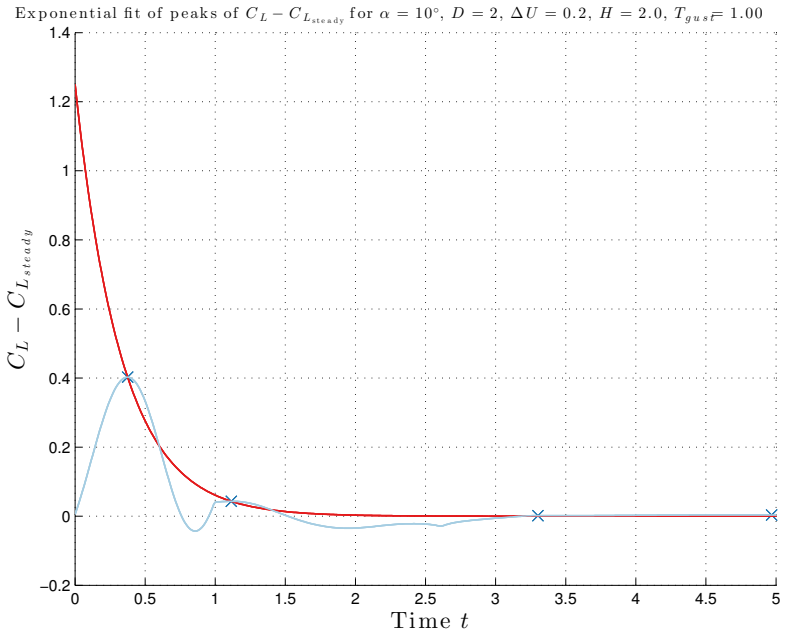


Figure 6.2: Lift response of the trailing plate approximated by an exponential function.

By selecting the peaks in the function C_L, it is possible to create an approximate function of the form (6.2). This extremely basic method proved to fit with very acceptable results all eight models that were tested. The interesting data was the decay b, considered the marker of how efficiently the design will "damp" the unsteady forces. This will allows us to determine which design will undergo milder loads.

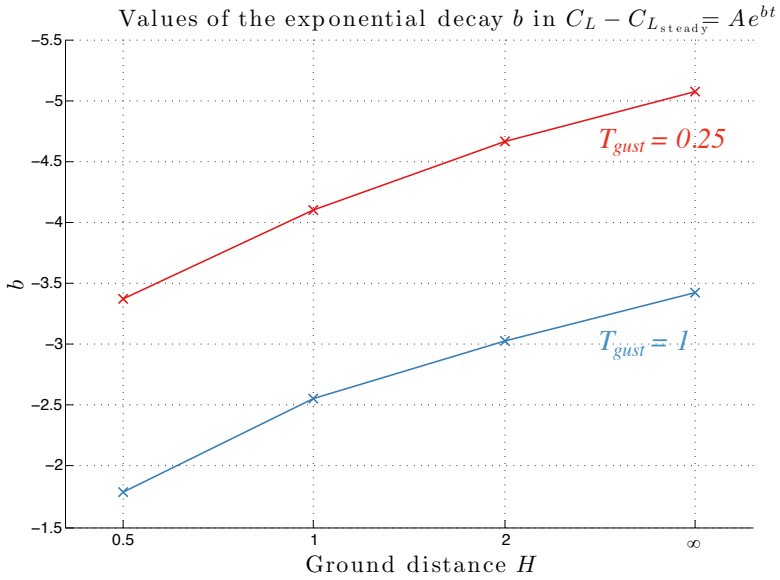


Figure 6.3: Comparison of the decay of C_L for different values of H and T_{gust}. A larger absolute value of b signifies faster decay.

Results are condensed in figure 6.3. It can be observed that for quickest gusts, all H ground distance configurations offer better damping than for the slowest gust. In addition, for both gust periods it seems that the tendency is to have a better "damping" or dissipation of the induced forces for larger ground distances. This is probably due to the effect of the ground on the path of the wake sheet. As discussed in Chapter 5, the ground effectively moves the wake directly towards the trailing airfoils, disabling it from vanishing in the surrounding flow, and intensifying its effect on the trailing plate. Thus, for closer ground distances the lift bump induced by the leading plate is slightly more intense.

6.3 Results of the study for a solar panel field

Eventually, both analyses yielded similar conclusions. The closer the ground distance, the worse the behavior of the system. This is produced by higher loads and worse dissipation of wake perturbations for lower values of the height H. It must be remembered that this model is two-dimensional, and there might be other three dimensional effects of the ground proximity omitted by this method.

In addition, it is recognized that this model could offer more aid in the design of panel structures if it could account for self-induced oscillations. This can only be accomplished by introducing elements that may produce instabilities in the flow, such as leading edge separation. Another feasible improvement to the method would be to couple the solver with a structural model, in order to be able to observe aeroelastic phenomena.

Chapter 7

Conclusion

This chapter lays out the conclusions that have been reached both during the development of the project and after the results have been obtained. An analysis of possible improvements and future development is included as well.

7.1 Outcome of the project

This thesis was conceived as an open-ended research project. The numerical method based on unsteady potential flow has proved to be firstly very efficient, and secondly simple enough to be completely developed in only a few months. Personally, it was very satisfying to program a fully functioning and capable flow solver from the ground up.

Nevertheless, the results of the analysis of flow over flat plates was not as intriguing as it had been expected. The reason for this is probably that common unstable flows, whose study could be considered more practical, can only be simulated when the method allows for instabilities. This project was not aimed at those situations, and it was for this reason limited to more uncomplicated unsteady flows, like discrete gusts.

However, the results of all the cases that were analyzed yielded interesting findings on the interactions of a tandem plate system and the consequences of the ground effect in both steady and unsteady flow situations.

7.2 Validity and applicability of results

Numerical methods based on unsteady potential flow have been in use for a very long time in computational aerodynamics, and its virtues were known beforehand.

In spite of that, it was completely necessary to perform a series of validation tests, which allowed us to understand in deeper detail the interactions in the system, and offered reliable evidences that the numerical solver was giving trustworthy results. Validation tests brought to light an undesired oscillation in the vorticity of the plates intrinsic to the method, which was eventually found to be due to the discretization. Corrective action was taken to mitigate these undesired effects, as explained in Appendix B *Validation of the code*. In conclusion, the validation that was carried out allowed a greater degree of confidence in the results of the analysis.

Furthermore, the results that have been exposed in this project are congruent, and can be related to phenomena visible in real flows. For this reason, the effort dedicated to fine tuning the method through the validation process has been considered successful.

It must be noted again that the results of all analyses of potential flows need to be carefully evaluated, because there exist strong limitations intrinsic to the model. As it has been presented in

Section 3.1.2, which was dedicated to the potential flow theory, many relevant physical phenomena in real flows are omitted in these simulations. Turbulence and boundary layer separation are the most relevant aspects of the flow over flat plates that are omitted in potential flow. Again, there exists a necessity to contemplate further development of the numerical model to account for some of these effects, or otherwise standard potential flow theory may lack validity in some real-world scenarios.

That being said, it was still possible to propose an application of the method to a real-life engineering problem, based on data obtained exclusively from this research project. The method employed is very suitable for the analysis of the gust response of solar panels, and the results are considered useful in such an engineering application. Again, bearing in mind the limitations of the method, and acknowledging the necessary further work to account for all remaining relevant phenomena expected in the real world.

7.3 Potential future development

Expanding the ideas summarized so far in this conclusion, this section lays out possible future developments to the work performed in this thesis.

It is clear that the flow solver that has been designed for this project lacked the ability to address more complex simulations. This can be made possible by introducing enhancements to the numerical model. The most important add-ons are those which would facilitate the simulation of unstable flows.

To begin with, simulation of unstable flows can be achieved by accounting for flow separation. The leading edge separation can be modeled to analyze airfoils at high angles of attack, adding the possibility of self-induced oscillations in the separated flow over a flat plate. The method developed in this thesis does not account for it, since it considers low angles of attack only, but it is possible to use potential flow theory on separated flows if some conditions are fixed, like the separation point [2].

By introducing leading edge separation, the airfoil will shed vorticity at two points, the trailing edge and the leading edge separation point. These two elements can produce oscillations in the nature of the vorticity shedding. Results derived from a self-induced oscillatory motion could be very relevant in aeroelastic applications or in structural fatigue analysis.

Another possible enhancement related to the problem of discrete gusts is to improve the gust model. The model has been designed with the discrete gust incorporated into the free stream velocity, as explained in Chapter 3. However, it is considered feasible to introduce a discrete gust which could be observed by the different fluid particles as it passes through the flow. This could be simulated by a distribution of moving doublet elements, but the validity of such a method is uncertain.

Lastly, minor modifications of the MATLAB program could allow the simulation of flows over different-shaped airfoils. Vortex distributions can account for zero-thickness bodies, and the possibilities of the flow solver can be broadened by applying the numerical method to different geometries.

Appendix A

Derivation of the matrix problem for N plates

The matrix problem that yields the circulation of all vortices in the system needs to undergo some modifications when there are more than one plate. This difficulty is introduced in section 3.3 *The MATLAB program* and the derivation yielding a valid matrix system to solve numerically is presented here.

Let us consider a system of N plates, where each plate has the same number of panels n. Capital letters I,J will refer to one of the N plates, while lower case letters i, j will make reference to one of the n panels in a plate, but also to the panel's associated vortex or collocation point.

We define the matrix $A_{I,J}$ as the matrix of influence coefficients of the vortices j of plate J and its corresponding new wake vortex $\Gamma_{J,W}$ on the collocation points i of plate I. This is a similar matrix to the first term in equation (3.53), but there exists one matrix for each plate's influence on another plate, and the Kelvin condition is not included, as it was in (3.53). The components of $A_{I,J}$ will be influence coefficients of the form $a_{I,i,J,j}$, which using the nomenclature that has just been defined can be expressed as:

$$a_{I,i,J,j} = \bar{v}_{I,i,J,j}(\Gamma_{J,j} = 1) \cdot \bar{n}_{I,i} \tag{A.1}$$

where $\bar{v}_{I,i,J,j} = (u, w)$ is the velocity induced by the vortex j of plate J, defined as $\Gamma_{J,j}$, on the collocation point i of plate I. The value of all influence coefficients $a_{I,i,J,j}$ can be found analogously to the single-plate case as per equations (3.43) and (3.44).

Now the matrix of influence coefficients $A_{I,J}$ can be formulated as:

$$A_{I,J} = \begin{pmatrix} a_{I,1,J,1} & a_{I,1,J,2} & \cdots & a_{I,1,J,n-1} & a_{I,1,J,n} & a_{I,1,J,W} \\ a_{I,2,J,1} & a_{I,2,J,2} & \cdots & a_{I,2,J,n-1} & a_{I,1,J,n} & a_{I,2,J,W} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ a_{I,n,J,1} & a_{I,n,J,2} & \cdots & a_{I,n,J,n-1} & a_{I,1,J,n} & a_{I,n,J,W} \end{pmatrix} \tag{A.2}$$

Then, the strengths of the vortices of each plate J are grouped in a vector $\bar{\Gamma}_J$, which can be represented as:

$$\bar{\Gamma}_J = \begin{pmatrix} \Gamma_{J,1} \\ \Gamma_{J,2} \\ \vdots \\ \Gamma_{J,n} \end{pmatrix} \tag{A.3}$$

In order to have a term for the sum of the strength of the vortices of each plate, the scalar $\Gamma_{J,\Sigma}$ is defined as the sum of the circulation of all n vortices $\Gamma_{J,j}$ of plate J:

$$\Gamma_{J,\Sigma} = \sum_{j=1}^{n} \Gamma_{J,j} \tag{A.4}$$

The term corresponding to the RHS of each collocation point i is the same as presented in (3.51), but must now account for the influence of all the wakes present in the system:

$$RHS_i = -\left[U_\infty(t) + u_W, W_\infty(t) + w_W\right]_i \cdot \bar{n}_i \qquad \text{(3.51 revisited)}$$

hence u_W and w_W hold their same definition: velocity induced by the previous wake vortices of known strength, but there will be as many wakes to consider as airfoils present. The scalar $RHS_{I,i}$ is defined to be the RHS corresponding to collocation point i of plate I. These are grouped in a vector $\overline{RHS}_I$ for notation simplicity:

$$\overline{RHS}_I = \begin{pmatrix} RHS_{I,1} \\ RHS_{I,2} \\ \vdots \\ RHS_{I,n} \end{pmatrix} \tag{A.5}$$

Finally, a system corresponding to any number of plates N can be represented in the following manner:

$$\begin{pmatrix} (A_{1,1}) & (A_{1,2}) & \cdots & (A_{1,N}) \\ \overline{1} & \overline{0} & \cdots & \overline{0} \\ (A_{2,1}) & (A_{2,2}) & \cdots & (A_{2,N}) \\ \overline{0} & \overline{1} & \cdots & \overline{0} \\ \vdots & \vdots & \ddots & \vdots \\ (A_{N,1}) & (A_{N,2}) & \cdots & (A_{N,N}) \\ \overline{0} & \overline{0} & \cdots & \overline{1} \end{pmatrix} \begin{pmatrix} \bar{\Gamma}_1 \\ \Gamma_{1,W} \\ \bar{\Gamma}_2 \\ \Gamma_{2,W} \\ \vdots \\ \bar{\Gamma}_n \\ \Gamma_{N,W} \end{pmatrix} = \begin{pmatrix} \overline{RHS}_1 \\ \Gamma_{1,\Sigma}(t-\Delta t) \\ \overline{RHS}_2 \\ \Gamma_{2,\Sigma}(t-\Delta t) \\ \vdots \\ \overline{RHS}_N \\ \Gamma_{N,\Sigma}(t-\Delta t) \end{pmatrix} \tag{A.6}$$

where $\overline{1}$ and $\overline{0}$ correspond to row vectors of ones and zeros, respectively. The system (A.6) is equivalent to (3.53), and solving for the term with all the circulations provides the solution to the flow field. Note that the main difference between the system for 1 plate and the system for N plates lies in the consideration of N Kelvin conditions (3.49). In order to keep consistency, they appear slightly scattered in the matrix, however their purpose is the same.

An example of a system for 2 plates is presented below in (A.7). It has been considered that each plate has only 3 panels, allowing for a final 8 by 8 matrix that has allowed for a simpler notation where all the scalar terms are directly represented. It relies on the nomenclature defined so far to transform (A.6) into: .

$$\begin{pmatrix} a_{1111} & a_{1112} & a_{1113} & a_{111W} & a_{1121} & a_{1122} & a_{1123} & a_{112W} \\ a_{1211} & a_{1212} & a_{1213} & a_{121W} & a_{1221} & a_{1222} & a_{1223} & a_{122W} \\ a_{1311} & a_{1312} & a_{1313} & a_{131W} & a_{1321} & a_{1322} & a_{1323} & a_{132W} \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ a_{2111} & a_{2112} & a_{2113} & a_{211W} & a_{2121} & a_{2122} & a_{2123} & a_{212W} \\ a_{2211} & a_{2212} & a_{2213} & a_{221W} & a_{2221} & a_{2222} & a_{2223} & a_{222W} \\ a_{2311} & a_{2312} & a_{2313} & a_{231W} & a_{2321} & a_{2322} & a_{2323} & a_{232W} \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \Gamma_{1,1} \\ \Gamma_{1,2} \\ \Gamma_{1,3} \\ \Gamma_{1,W} \\ \Gamma_{2,1} \\ \Gamma_{2,2} \\ \Gamma_{2,3} \\ \Gamma_{2,W} \end{pmatrix} = \begin{pmatrix} RHS_{1,1} \\ RHS_{1,2} \\ RHS_{1,3} \\ \Gamma_{1,\Sigma}(t-\Delta t) \\ RHS_{2,1} \\ RHS_{2,2} \\ RHS_{2,3} \\ \Gamma_{2,\Sigma}(t-\Delta t) \end{pmatrix} \tag{A.7}$$

Appendix B

Validation of the code

Usual validation procedures involve comparing the obtained results with other contrasted solutions. Since potential theory offers known analytical solutions for the flow over a flat plate, these analytical solutions have been taken for the validation of the solver. Relevant computational parameters presented in Section 3.3.2 have also been chosen by contrasting the results obtained, using additionally two-plate and ground-effect cases.

All parameters presented in this section follow the nondimensionalization defined in Chapter 4 *Procedure* and are thus dimensionless quantities.

B.1 Steady State Solution

The reference value of C_L is:

$$C_L = 2\pi \sin\alpha \tag{B.1}$$

which is a result derived from the analytical solution of uniform flow U_∞ at an angle α over a flat plate, using conformal mapping and the Joukowski transformation [1, pp. 128–133]. For one plate without ground effect, the resulting steady state value of C_L is exact, with registered errors of the order of 1×10^{-15}, entirely due to floating-point precision. This is true for any value of N_V (Number of panel elements). Total circulation in a steady situation is thus unaffected by N_V.

At the same time, C_D was exactly zero, expected result that agrees with D'Alembert's paradox. Ensuring that the forces in the simplest possible case are correct not only checks that the parts of the program related to steady problems are correct; it also corroborates that the definition of the force vectors on the airfoil are correct.

B.2 Sudden Acceleration. Wagner's function

When an airfoil is suddenly set into motion from stop to a uniform velocity U, the forces build up with a lag until they reach a steady value. This transition was studied and a solution for two-dimensional incompressible flow was obtained by Wagner, Küssner, von Kármán and Sears, and others [11, pp. 206-210]. The unsteady lift $L(t)$ can be expressed as:

$$L(t) = 2\pi \frac{c}{2} \rho U w \Phi(\tau) \tag{B.2}$$

where w is the downshash $w = U \sin\alpha \approx U\alpha$, the vertical component of the velocity; and $\Phi(t)$ is the so-called Wagner's function, dependent on the nondimensional quantity $\tau = \frac{Ut}{c/2}$. Wagner's function can be approximated by the expression proposed by W. P. Jones [11]:

$$\Phi(\tau) \approx 1 - 0.165e^{-0.041\tau} - 0.335e^{-0.32\tau} \tag{B.3}$$

Comparing the different results to the unsteady lift given by this approximation will allow us to find the best fit. The first objective is to observe the effects that the computational parameters N_V and CFL have on the resulting lift. Several cases were analyzed, and the most relevant figures are presented next.

So far the only known concern dealing with CFL and N_V is the requirement to have a small enough time step Δt so that quick gusts of the order of $T_{gust} = 1/4$ can be appreciated, which was explained in *3.3.2 Relevant Computational Parameters*. This forces a value of CFL as small as possible and a value of N_V as high as possible, but this means increased computational cost from the beginning.

Figures B.1, B.2 and B.3 show that the solution is more accurate the higher the value of N_V, as expected, since more panels are more similar to a theoretical continuous distribution of vorticity.

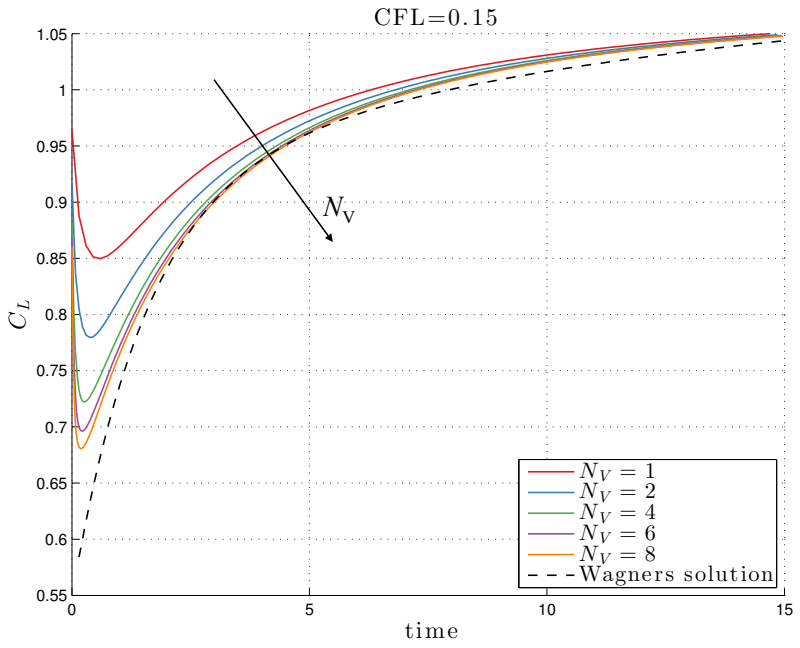


Figure B.1: Response in C_L of one flat plate to a sudden acceleration. CFL= 0.15.

However, surprisingly the best fit is seen for the highest value of CFL = 0.50. A negative aspect of a high CFL is the loss of fast phenomena: in this case at $t = 0$ the lift approaches infinity quite visibly for lower CFL but not for CFL = 0.50. This result does not agree with Wagner but does have theoretical foundation. The unsteady terms of the forces of the lift (3.56) are infinite for the limiting case $t \to 0$. This is indeeed visible the lower the CFL.

So far there is conclusive evidence that higher values of N_V will provide better accuracy. But the best fit can not be determined yet.

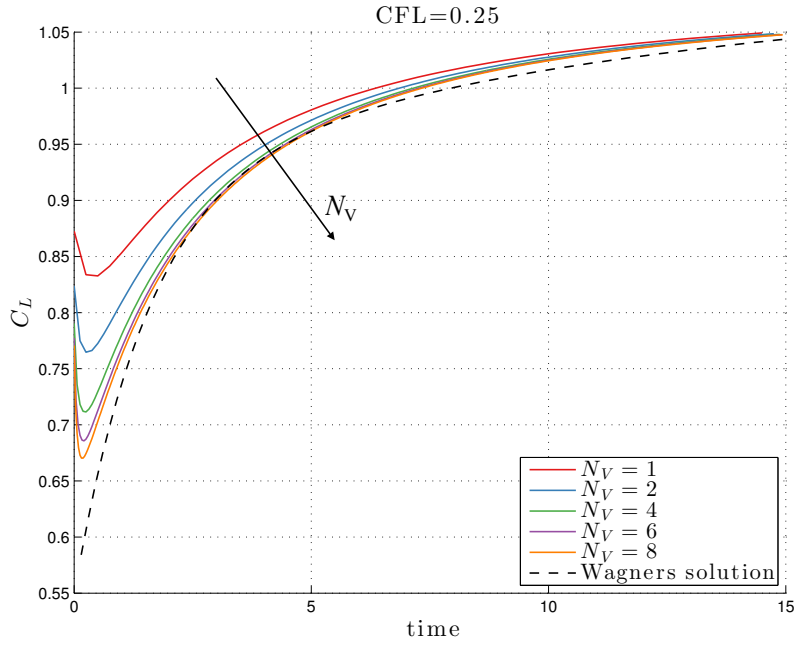


Figure B.2: Response in C_L of one flat plate to a sudden acceleration. CFL= 0.25.

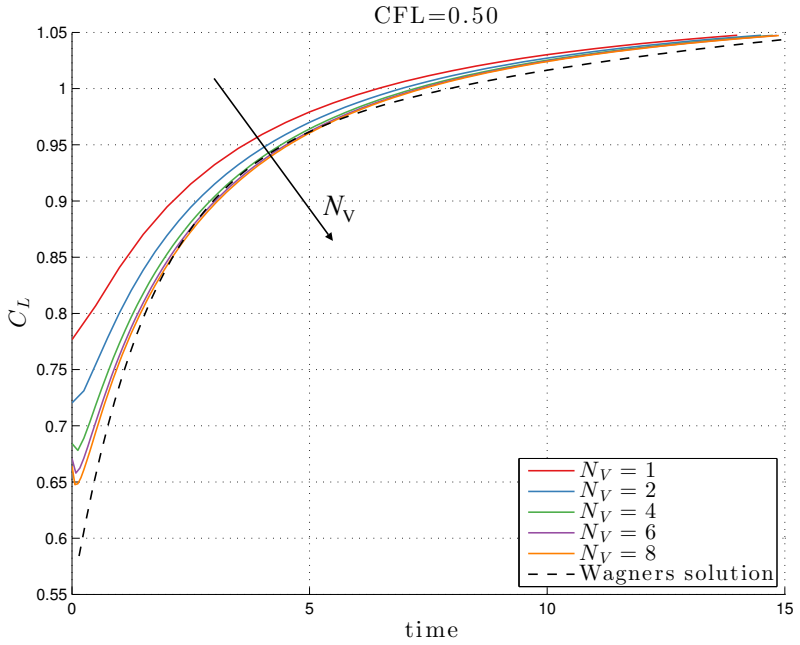


Figure B.3: Response in C_L of one flat plate to a sudden acceleration. CFL= 0.50.

B.3 Numerical instability in discrete gust cases

Preliminary solutions involving discrete wind gusts, with two plates and ground effect yielded disconcerting results. These are shown in figures B.4, B.5 and B.6. The leading plate is labeled plate 1 and the trailing one, plate 2.

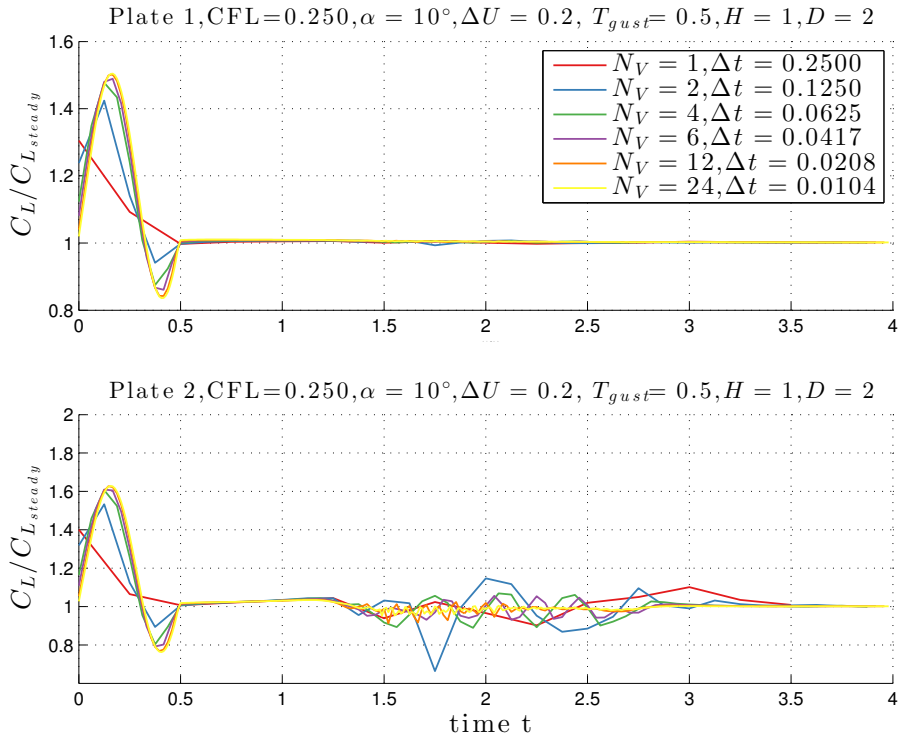


Figure B.4: Lift force on two plates in a gust response. CFL = 0.25 = 1/4

From these figures, it appears that some interaction happening near the trailing plate yields erratic results in the forces of this second plate. These are not supported by any physical reason and must be attributed to the numerical method.

It turns out that these undesired oscillations occur when the leading plate vortices interact in unwanted ways with the panel vortices of the trailing plate. This happens only for some combinations of both geometric parameters and gust parameters. For instance, stronger vortices due to a higher value of ΔU sometimes provoke the appearance of these numerical instabilities. Near-ground cases influence the wake development, avoiding the wake vortices to dissipate vertically and thus provoking that they encounter the trailing edge vortices too close.

Figure B.7 shows the wake development in a case with these undesired oscillations. Undesired interaction of wake vortices with panel vortices can yield erratic results. In the particular case presented in figure B.7, the first wake's vortices run so close to the second panel that the discrete vortices influence becomes too dominant, yielding erratic reads in the lift. In this particular case wake vortices are seen to cross the plate, violating the boundary condition.

It was determined that this numerical instability was caused by leading airfoil's wake vortices and trailing airfoil's panel vortices getting too close and producing undesired interactions, entirely due to the discretization of the airfoils.

In some cases it was even observed that wake vortices would cross the second panel, violating the boundary condition. This can only happen if a wake vortex gets so close to a panel vortex that it can cross the boundary due to their strong vortex-vortex interaction in such proximity.

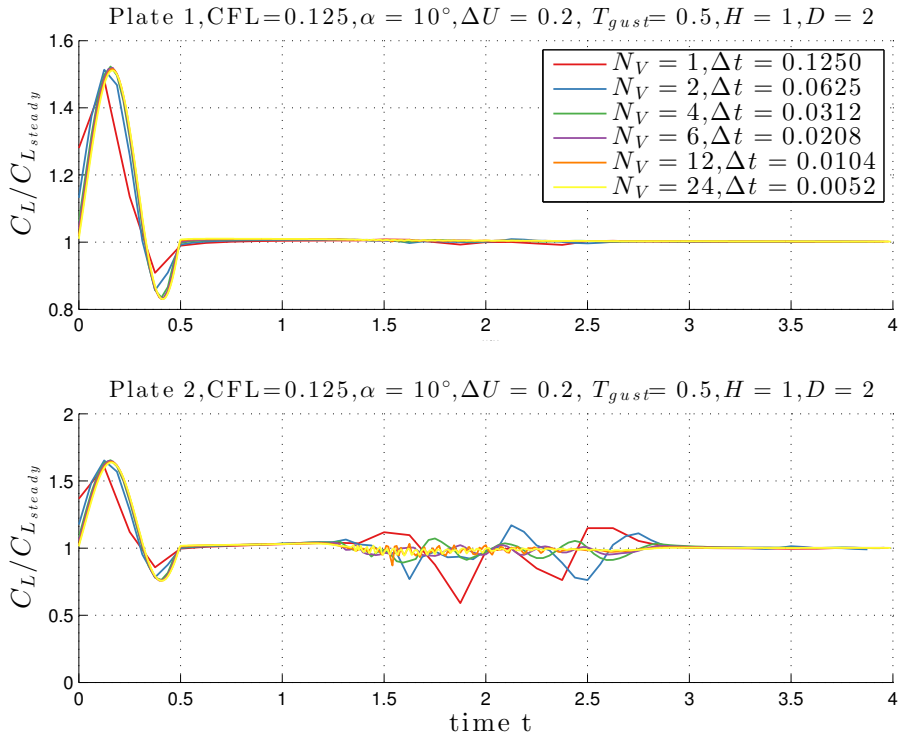


Figure B.5: Lift force on two plates in a gust response. CFL=1/8

Even though steady response was not affected by this phenomenon, these undesired interactions produce noise in the critical moment of the wake-plate interaction, and thus need to be reduced.

It was realized that higher values of N_V reduced the gravity of the instability, as seen in figures B.4, B.5 and B.6. However, if a given case was prone to have stability it was most times inevitable no matter what computational parameters CFL or N_V were chosen. Luckily, the eventual set of parameters allowed most cases to be free of this problem.

After having awareness of the discretization-induced undesired interaction, several sudden acceleration cases were checked again for higher values of N_V. It was determined that $N_V = 24$ provided sufficient damping of the undesired oscillations, good accuracy with respect to the analytical Wagner's solution for the lift, and overall acceptable computational efficiency, as seen in following figures. At this time, CFL was chosen to be CFL=0.25, which, for $N_V = 24$ provides a valid representation of unsteady terms and allows for smooth models of the shortest gust, as presented in Section 3.3.2 *Relevant Computational Parameters* and figure 3.21.

Figure B.8 shows the results of one plate's sudden acceleration for several values of N_V and the factor in $D_W = 0.2\text{–}0.3U_\infty(t)\Delta t$ as well. The choices of the factor in D_W were among 0.2–0.3, as these are the values recommended for this method by J. Katz and A. Plotkin [1], as explained in Section 3.3.2 *Relevant Computational Parameters*. It is observed that a value of 0.2, such that the shedding distance of new wake vortices is $D_W = 0.2U_\infty(t)\Delta t$, provides a more correct result, but not by much. Since this parameter does not have much influence on any other properties of the program, it was chosen to be 0.2 directly due to approximating better Wagner's solution.

It is also seen in figure B.8 that $N_V = 48$ provides a slightly fitter result. However, the result is not really appreciable, and doubling the number of panel vortices does increase dramatically

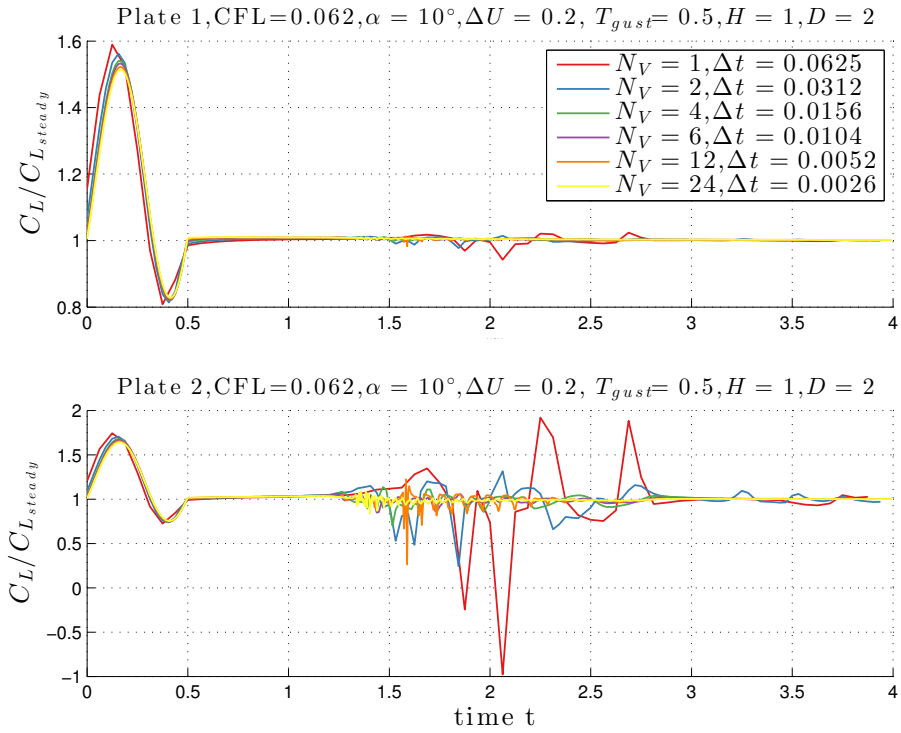


Figure B.6: Lift force on two plates in a gust response. CFL=1/16

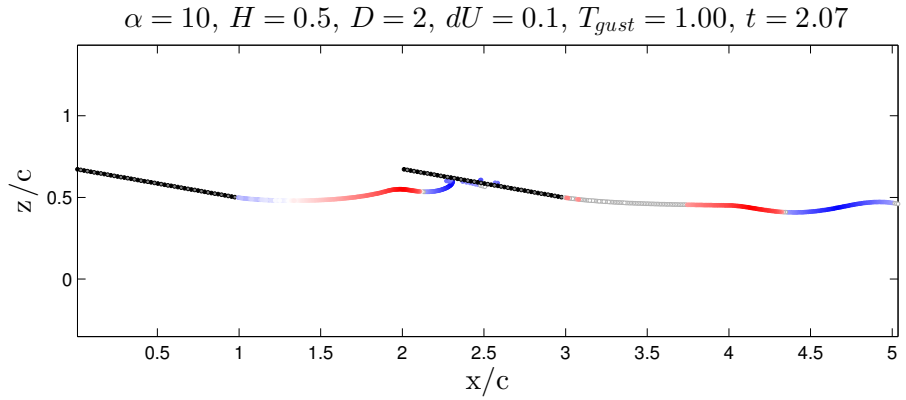


Figure B.7: Undesired interaction of wake vortices with panel vortices.

the computational time. In addition, $N_V = 24$ already provides an acceptable behavior of the undesired oscillations described in previous paragraphs, and presented in figure B.9 again for updated values of N_V.

Since the difference in results from choosing $N_V = 24$ or $N_V = 48$ would be negligible, even when the number of panels are being doubled, $N_V = 24$ is determined valid and its error due to providing a less continuous circulation distribution than a higher N_V is negligible. $N_V = 24$ is also the better choice due to computational efficiency.

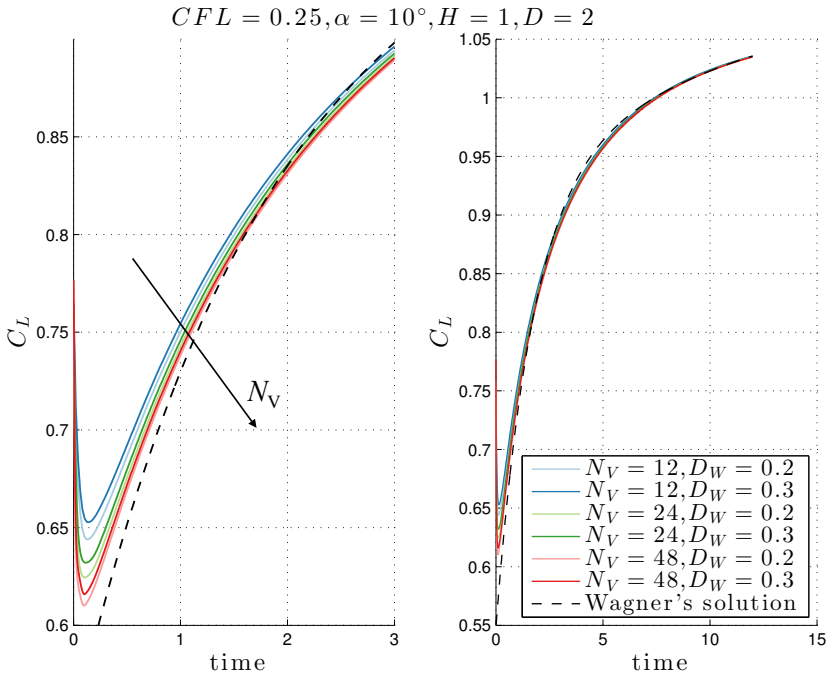


Figure B.8: A plate's response to sudden acceleration. Influence of N_V and the D_W factor. The left panel shows an inset of the right panel.

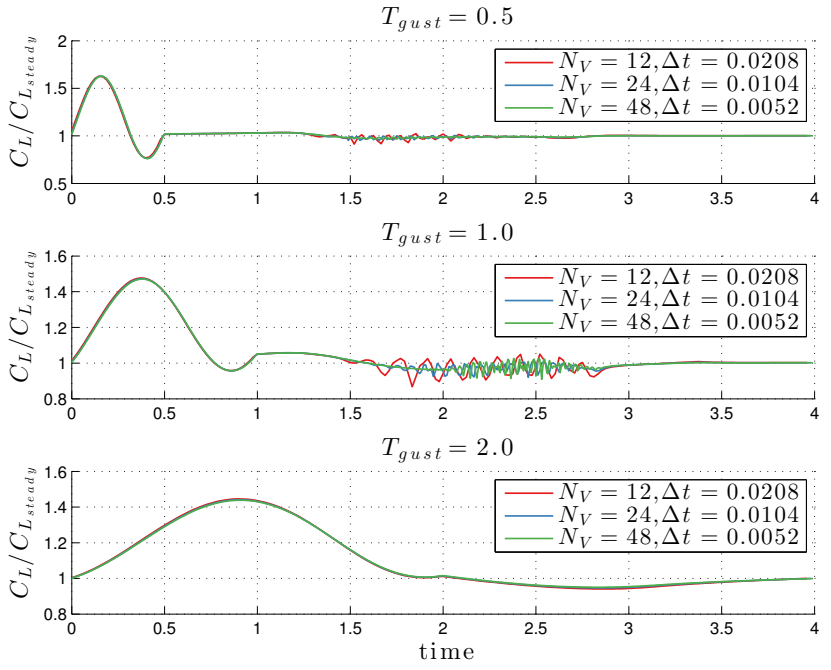


Figure B.9: Trailing plate's response to sudden acceleration of two plates in ground effect. Influence of N_V and T_{gust} for $\alpha = 10°$, $H = 1$ and $D = 2$.

Eventually the solver's results have been validated, and the best computational parameters have been determined. These choices optimize the accuracy of the results that will be obtained.

Appendix C

The MATLAB source code

This appendix presents the source code of the solver. It has been entirely written for this thesis, based on the scheme presented in Section 3.3 *The MATLAB program*.

```
function [Cl,Cd,Cm_LE,Cl_max,Cd_max,Cm_LE_max,time_vector,ut,Cl_steady] = gustsolver(alphadeg,h,d,
    dUgust,Tgust,n_vortices,final_time,CFL_factor,n_real_plates,ground_effect,steady_solution,
    plots_on,IT_PIC)
% GUSTSOLVER Calculates forces on N flat plates subject to a wind gust.
% GUSTSOLVER employs unsteady potential theory to solve the flow field
% around N flat plates near the ground, which are subject to an incoming
% 1-cos shaped gust of amplitude dUgust and period Tgust.
% The outputs for both plates are Cl, Cd, Cm_LE, Cl_max, Cd_max, Cm_LE_max,
% time_vector, ut (freestream velocity vector) and Cl_steady.
% The inputs are the angle of attack, the height from the ground, the
% distance between plates' leading edges, dUgust and Tgust period for the
% gust, the number of vortices of each plate, the time to run the
% calculation and the Courant_Friedrichs_Lewy condition (CFL factor).
% Additionally n_real_plates can be set to any number of plates, default
% being 2.
% Two more additional arguments can be added as zeros to deactivate the
% ground effect and/or the steady state initialization respectively. In
% case steady_state is set to zero, the analysis will consider sudden
% acceleration of the system of plates from 0 to 1 at time zero. Steady
% cases can also be run setting Tgust to 0. An additional fourth argument
% plots_on can be set to 1 to show a representation of the system at each
% time step (This consumes a lot of time). A fifth additional argument
% IT_PIC allows the program to save a PDF picture of the wake development
% every IT_PIC iterations. If it is not defined, no pictures are saved.
%
% As an example: run "[Cl,Cd,Cm_LE,Cl_max,Cd_max,Cm_LE_max,time_vector,ut,Cl_steady]=gustsolver
    (10,1,2,0.2,1,12,5,0.25,2,1,1,1,50);".
%
%
% Author: Alvaro Martin Sampayo / alvaro.m.sampayo@alumnos.uc3m.es
%
% This program was developed as part of a Bachelor's thesis: "Analysis of
% the Effect of Gusts on an Array of Plates with Ground Effect".
%
% Universidad Carlos III de Madrid (UC3M). June 2014.
%

%% PROGRAM SET UP
close all

% FUNCTION'S OPTIONAL INPUTS
if exist('plots_on','var')==1
    scrsz = get(0,'ScreenSize');
    fhandle_plots_on=figure('Position',[1 1 scrsz(3) scrsz(4)]);
else
    plots_on=0;% 1 or 0 Introduces progress plots on all timesteps (very time-consuming)
end
if exist('n_real_plates','var')==1
    if n_real_plates<=0
        error('Please select a natural number for the number of real plates.')
    end
else
    n_real_plates=2; %This function considers 2 plates if not specified, but any number can be
        chosen.
end
if exist('ground_effect','var')==1
else
    ground_effect=1;
end
if exist('steady_solution','var')==1
else
    steady_solution=1;
```

```
59  end
60  if exist('IT_PIC','var')==1
61  else
62      IT_PIC=0;%Effectively no figure output
63  end
64  %% GEOMETRY AND COMPUTATION PARAMETERS
65  h(1:n_real_plates)=h;
66  c(1:n_real_plates)=1; %Normalized
67  alphadeg(1:n_real_plates)=alphadeg;
68  alpha=alphadeg.*pi/180;
69  d(1:n_real_plates-1)=d:d:d*(n_real_plates-1); %spaces any number of plates d from each other.
70  d=[0,d];%d(1)=0 so that the first plate is located at x=0 in the fixed frame.
71
72  u_start=1; %Normalized
73
74  if ground_effect~=0 % Automatic set up of mirror plates if ground_effect is not deactivated.
75      n_plates=2*n_real_plates;
76      h(n_plates/2+1:n_plates)=-h(1:n_plates/2);
77      c(n_plates/2+1:n_plates)=c(1:n_plates/2);
78      alpha(n_plates/2+1:n_plates)=-alpha(1:n_plates/2);
79      d(n_plates/2+1:n_plates)=d(1:n_plates/2);
80  else
81      n_plates=n_real_plates;
82  end
83
84  c_cell=c./n_vortices; %The characteristic length for the computation is the cell length.
85  dt=c_cell(1)/u_start*CFL_factor;
86  n_iterations=ceil(final_time/dt); %Rounded up for non integer results.
87
88  if steady_solution~=0 % Sets up gust profile when steady_solution is not deactivated.
89      if Tgust>0
90          ut=zeros(1,n_iterations);
91          for it=1:floor(Tgust/dt)
92              ut(1+it)=u_start+ (dUgust/2*(1-cos(2*pi*(it*dt)/Tgust)));
93          end
94          ut(1)=u_start;
95          ut((it+2):n_iterations)=u_start;
96      elseif Tgust==0 %%if Tgust==0 then solves for a steady case.
97          ut(1:n_iterations)=u_start;
98      end
99
100 else %If steady_solution==0, then solves for a sudden acceleration.
101     ut(1:n_iterations)=u_start;
102 end
103 dx_new_wake_vortex=0.2*u_start*dt; %Distance new wake vortex is shedded from the trailing edge.
        Typically 0.2~0.3*U*dt
104 max_vortices_wake=n_iterations; %Facilitates wake management - to use in further versions with
        better memory management through vortex reduction
105
106 %% CONSTANTS
107 rho=1.225; % [mass / length^3] 1.225 kg /m^3 ISA at SEA LEVEL
108 %Rho does not actually influence any variable, as all results are given as
109 %dimensionless coefficients.
110
111 %% SPACE ALLOCATION
112 %main variables
113 vortices_g=zeros(n_plates,n_vortices);
114 vortices_x=zeros(n_plates,n_vortices);
115 vortices_z=zeros(n_plates,n_vortices);
116 coll_x=zeros(n_plates,n_vortices);
117 coll_z=zeros(n_plates,n_vortices);
118 wake_g=zeros(n_plates,max_vortices_wake);
119 wake_x=zeros(n_plates,max_vortices_wake);
120 wake_z=zeros(n_plates,max_vortices_wake);
121 % calculation of forces variables
122 Lx=zeros(n_real_plates,n_iterations-1);
123 Lx_steady=zeros(n_real_plates,1);
124 L=zeros(n_real_plates,n_iterations-1);
125 D=zeros(n_real_plates,n_iterations-1);
126 L_prime=zeros(n_real_plates,n_vortices);
127 D_prime=zeros(n_real_plates,n_vortices);
128 Cl=zeros(n_real_plates,n_iterations-1);
129 Cd=zeros(n_real_plates,n_iterations-1);
130 Cm_LE=zeros(n_real_plates,n_iterations-1);
131 u_press=zeros(n_plates,n_vortices);%Values of velocity at each vortex stored across iterations for
        pressure calculation.
132 w_press=zeros(n_plates,n_vortices);
133 %WRU (Wake roll-up) displacements variables
```

```
u_wru=zeros(n_plates,max_vortices_wake);
w_wru=zeros(n_plates,max_vortices_wake);

% Now program starts.

%% DISCRETIZATION. GENERATION OF GEOMETRY

% locates vortices_x, vortices_z, coll_x, coll_z on the fixed
% reference frame.

for i_plate=1:n_plates %discretizes all plates' vortices and collocation points
    for i_vortex=1:n_vortices %locates  vortices and collocation points on plate
        vortices_x(i_plate,i_vortex)=(c_cell(i_plate)*(i_vortex-1)+c_cell(i_plate)*(1/4))*cos(alpha(
            i_plate))+d(i_plate); %AT QUARTER CHORD
        vortices_z(i_plate,i_vortex)=h(i_plate)+sin(alpha(i_plate))*c(i_plate)-(vortices_x(i_plate,
            i_vortex)-d(i_plate))*tan(alpha(i_plate));
        coll_x(i_plate,i_vortex)=(c_cell(i_plate)*(i_vortex-1)+c_cell(i_plate)*(3/4))*cos(alpha(
            i_plate))+d(i_plate); %AT THREE QUARTERS CHORD
        coll_z(i_plate,i_vortex)=h(i_plate)+sin(alpha(i_plate))*c(i_plate)-(coll_x(i_plate,i_vortex)
            -d(i_plate))*tan(alpha(i_plate));
    end
end

%% Calculation of influence coefficients.
%Calculates the influence of vortices on existing vortices plus the
%influence of the newly shed vortex on the remaining vortices.
inf_coeffs=zeros(n_plates,n_vortices,n_plates,n_vortices);%space allocation
inf_coeffs_latest_wake=zeros(n_plates,n_vortices,n_plates);
for i_plate=1:n_plates %Loop to obtain influence coefficients
    for i_coll=1:n_vortices %collocation point loop
        for i_plate_vor=1:n_plates %collocation i_col on plate i_plate is affected by all plates, so
             i_plate_vor goes over them.

            %influence of new wake vortex of gamma 1 on
            %all collocation points
            [u_ic,w_ic]=VOR2D(coll_x(i_plate,i_coll),coll_z(i_plate,i_coll),(d(i_plate_vor)+(c(
                i_plate_vor))*cos(alpha(i_plate_vor))+dx_new_wake_vortex),h(i_plate_vor),1);
            inf_coeffs_latest_wake(i_plate,i_coll,i_plate_vor)=dot([u_ic,w_ic],[sin(alpha(i_plate)),
                cos(alpha(i_plate))]);

            %each plate considered to influence considered here. Loop
            %over vortices substituted by vectorial VOR2Dvv. VOR2Dvv
            %is especially employed here since we are interested on
            %the influence of each vortex on our collocation point,
            %not in the sum of their influences yet. VOR2D will do
            %that.
            [u_ic,w_ic]=VOR2Dvv(coll_x(i_plate,i_coll),coll_z(i_plate,i_coll),vortices_x(i_plate_vor
                ,:),vortices_z(i_plate_vor,:),1);
            inf_coeffs(i_plate,i_coll,i_plate_vor,:)=dot([u_ic;w_ic],[ones(1,n_vortices)*sin(alpha(
                i_plate));ones(1,n_vortices)*cos(alpha(i_plate))],1);%vectorial dot along dimension
                1
            %end
        end
    end
end
clear u_ic w_ic i_coll i_plate_vor i_vor

%inf_coeffs must be transformed into a super 2-dim matrix (ICM). which
%includes vortices being shedded onto the wake as well.
%
% Large Influence Coefficients Matrix. Concatenation of existing influence
% coefficients into a usable matrix containing all body vortices plus all
% newly shed wake vortices. Includes Kelvin condition equations: one for
% each plate system (circulation of 1 plate + its shedded wake must remain
% constant.) Steady Solver right below uses a simplified IC matrix.

%% STEADY STATE SOLVER
if steady_solution~=0 %Same procedure as unsteady ICM in the block below.
    Asteadyaux=zeros(n_vortices,n_vortices); %space allocation
    for i_plate_done=1:n_plates %Loop to obtain matrix
        Asteady=zeros(n_vortices,n_vortices);%clear and temporal space allocation for this loop
        for i_plate_does=1:n_plates
            if i_plate_does==1
                for dim1=1:n_vortices %Concatenation loop
                    for dim2=1:n_vortices
                        Asteady(dim1,dim2)=inf_coeffs(i_plate_done,dim1,i_plate_does,dim2);
                    end
                end
```

```
202             elseif i_plate_does>1
203                 for dim1=1:n_vortices %Concatenation loop
204                     for dim2=1:n_vortices
205                         Asteadyaux(dim1,dim2)=inf_coeffs(i_plate_done,dim1,i_plate_does,dim2);
206                     end
207                 end
208                 Asteady=[Asteady,Asteadyaux];
209             end
210         end
211         if i_plate_done==1
212             ICMsteady=Asteady;
213         elseif i_plate_done>1
214             ICMsteady=[ICMsteady; Asteady];
215         end
216     end
217     clear Asteady Asteadyaux
218     RHSsteady=zeros(size(ICMsteady,1),1);%Preallocating RHSsteady vector for speed
219
220     for i_plate=1:n_plates
221         RHSsteady((n_vortices*(i_plate-1)+1):(n_vortices*i_plate))=-u_start*sin(alpha(i_plate));
222     end
223
224     %Solution of the steady state problem. Introducing t=0 plate vorticities
225     aux_g=ICMsteady\RHSsteady; %Matrix is singular to working precision? console warning
226     for i_plate=1:n_plates %This loop saves results into data matrices.
227         vortices_g(i_plate,:)=aux_g(((i_plate-1)*(n_vortices)+1):(i_plate*n_vortices));
228     end
229     clear aux_g ICMsteady RHSsteady
230
231     %% Steady forces calculation
232     for i_plate=1:n_real_plates
233         for i_vortex=1:n_vortices
234             [u,w]=VFIELD(vortices_x(i_plate,i_vortex),vortices_z(i_plate,i_vortex),u_start,
                    vortices_g,vortices_x,vortices_z,wake_g,wake_x,wake_z,0); %limit_vortices_wake
                    preset to zero (so that VFIELD doesn't take wakes)
235
236             L_prime(i_plate,i_vortex)=rho*u*vortices_g(i_plate,i_vortex);
237             D_prime(i_plate,i_vortex)=-rho*w*vortices_g(i_plate,i_vortex);
238
239             %calculation of moment.
240             %Force (D*i + L*k) dot normal vector times distance.
241             x=c_cell(i_plate)*(i_vortex-1)+(1/4)*c_cell(i_plate);
242             Lx_steady(i_plate)=Lx_steady(i_plate) +(dot([D_prime(i_plate,i_vortex),L_prime(i_plate,
                    i_vortex)],[sin(alpha(i_plate)),cos(alpha(i_plate))] ))*x;
243         end
244         L_steady(i_plate)=sum(L_prime(i_plate,:));
245         D_steady(i_plate)=sum(D_prime(i_plate,:));
246     end
247     Cl_steady=L_steady./(rho*.5*(u_start^2)*c(i_plate));
248     Cd_steady=D_steady./(rho*.5*(u_start^2)*c(i_plate));
249     Cm_LE_steady=Lx_steady./(rho*.5*(u_start^2)*(c(i_plate))^2); %at LE
250 end %end of steady state solver
251
252 %% Unsteady ICM calculation
253 Aaux=zeros(n_vortices,n_vortices+1);%space allocation
254 for i_plate_done=1:n_plates %Loop to obtain ICM (Influence Coeff Matrix)
255     A=zeros(n_vortices,n_vortices+1);%clear and temporal space allocation for this loop
256     for i_plate_does=1:n_plates
257         if i_plate_does==1
258             %concatenation doesn't like 4Dim mixed with 3Dim. Fixed
259             %with concatenation loop.
260             for dim1=1:n_vortices %Concatenation loop
261                 for dim2=1:n_vortices
262                     A(dim1,dim2)=inf_coeffs(i_plate_done,dim1,i_plate_does,dim2);
263                 end
264                 A(dim1,n_vortices+1)=inf_coeffs_latest_wake(i_plate_done,dim1,i_plate_does);
265             end
266         elseif i_plate_does>1
267             for dim1=1:n_vortices %Concatenation loop
268                 for dim2=1:n_vortices
269                     Aaux(dim1,dim2)=inf_coeffs(i_plate_done,dim1,i_plate_does,dim2);
270                 end
271                 Aaux(dim1,n_vortices+1)=inf_coeffs_latest_wake(i_plate_done,dim1,i_plate_does);
272             end
273             A=[A,Aaux];
274         end
275     end
276     %Includes embedded Kelvin condition for this plate
```

```
277     Kelvinaux=zeros(1,size(A,2)); %This row will be all zeros except for the same plate itself, for
            which it will be ones.
278     Kelvinaux( (1+(n_vortices+1)*(i_plate_done-1)) : (n_vortices+1)*i_plate_done )=1;
279
280     if i_plate_done==1
281         ICM=[A; Kelvinaux];
282     elseif i_plate_done>1
283         ICM=[ICM; A; Kelvinaux];
284     end
285 end
286 clear A Aaux Kelvinaux dim1 dim2 i_plate_does
287 RHS=zeros(size(ICM,1),1);%Preallocating RHS vector for speed
288
289 %% Prepares plots:
290 if ground_effect~=0
291     n_plot_plates=n_plates/2;
292 else
293     n_plot_plates=n_plates;
294 end
295 if plots_on==1
296     figure(1)
297 end
298 reverseStr=''; %Used in progress message displayed in the console.
299
300
301 %% MAIN TIME LOOP
302
303 for IT=1:n_iterations % TIME LOOP STARTS HERE.
304
305     if plots_on~=1 % Displays a progress message if there are no plots.
306         progress_msg=sprintf('\n Computation Running. Iteration %i / %i \n',IT,n_iterations);
307         fprintf([reverseStr, progress_msg]);
308         reverseStr=repmat(sprintf('\b'), 1, length(progress_msg));
309     end
310
311     %% Saving gamma plates for pressure calculation (time-derivative of phi)
312     vortices_g_prev=vortices_g;
313
314     %% SHEDDING OF NEW WAKE VORTICES
315     for i_plate=1:n_plates
316
317         %Body-Fixed RF is our frame of reference.
318         if IT>1 %VORTICES MOVED TO MAKE ROOM AT 1 FOR THE NEW ONE (FIFO)
319             wake_x(i_plate,2:end)=wake_x(i_plate,1:(end-1));
320             wake_z(i_plate,2:end)=wake_z(i_plate,1:(end-1));
321             wake_g(i_plate,2:end)=wake_g(i_plate,1:(end-1));
322         end
323         %New wake vortex created. Circulation yet unknown
324         wake_x(i_plate,1)=d(i_plate)+(c(i_plate))*cos(alpha(i_plate))+dx_new_wake_vortex;%dx_ solo
                aqui, goes with the flow see Katz p.412
325         wake_z(i_plate,1)=h(i_plate);
326         wake_g(i_plate,1)=0;
327     end
328     if IT>max_vortices_wake
329         limit_vortices_wake=max_vortices_wake;
330     else
331         limit_vortices_wake=IT;
332     end
333
334     %% MOMENTARY RHS VECTOR CALCULATION (for this iteration)
335     %This vector is known, only needs to be found. Same size every
336     %iteration. Preallocated outside time loop, overwritten every time.
337
338     %Includes RHS of normal velocities and embedded Kelvin condition eqns.
339     for i_plate_done=1:n_plates
340         for i_coll_done=1:n_vortices
341             u_w=0;%clear to zero
342             w_w=0;%clear to zero
343             for i_wake_does=1:n_plates
344                 %  vectorial computation on 2:limit_vortices_wake because newly shed zero-gamma
                        vortex is at 1, so skipped.
345                 [u_aux,w_aux]=VOR2D(coll_x(i_plate_done,i_coll_done),coll_z(i_plate_done,i_coll_done
                        ),wake_x(i_wake_does,2:limit_vortices_wake),wake_z(i_wake_does,2:
                        limit_vortices_wake),wake_g(i_wake_does,2:limit_vortices_wake));
346                 u_w=u_w+u_aux;
347                 w_w=w_w+w_aux;
348             end
```

```
349             RHS((i_plate_done-1)*(n_vortices+1)+i_coll_done)=dot([-(ut(IT)+u_w),-w_w],[sin(alpha(
                    i_plate_done)),cos(alpha(i_plate_done))]);
350         end
351         %Kelvin Condition (1 per plate):
352         if IT==1 && steady_solution==0 %if first iteration and no steady solution exists, total
                circulation of previous step is zero, thus it is omitted, as the space had been
                allocated as zeros before.
353         else
354             RHS(i_plate_done*(n_vortices+1))=sum(vortices_g(i_plate_done,:));
355         end
356     end
357
358     %% SOLUTION OF MATRIX PROBLEM. FINDING VORTICITY AND STORING IT
359
360     aux_g=ICM\RHS; % Matrix problem is solved and circulations are found.
361
362     for i_plate=1:n_plates %This loop saves results into data matrices.
363         vortices_g(i_plate,:)=aux_g(((i_plate-1)*(n_vortices+1)+1):((i_plate-1)*(n_vortices+1)+
                n_vortices)); %Vectorized version of a loop.
364         wake_g(i_plate,1)=aux_g(i_plate*(n_vortices+1)); %new wake point at 1
365     end
366
367     %Now post-processing can be done. Velocities and Loads can be
368     %calculated. The vorticity of the body has been determined for this IT
369     %time step. The newly shed wake point vorticity as been found as well.
370
371     %% POSTPROCESSING - CALCULATION OF PRESSURES AND LOADS
372     if IT==1 %We need information of previous timesteps, as pressures for a certain IT time are
            found when timestep is IT+1 (EXPLICIT EULER)
373         for i_plate=1:n_real_plates
374             for i_vortex=1:n_vortices
375                 [u,w]=VFIELD(vortices_x(i_plate,i_vortex),vortices_z(i_plate,i_vortex),ut(IT),
                        vortices_g,vortices_x,vortices_z,wake_g,wake_x,wake_z,limit_vortices_wake);
376                 u_press(i_plate,i_vortex)=u;
377                 w_press(i_plate,i_vortex)=w;
378             end
379         end
380     end
381     if IT>1
382         u_press_prev=u_press;
383         w_press_prev=w_press;
384         for i_plate=1:n_real_plates
385             for i_vortex=1:n_vortices
386                 [u,w]=VFIELD(vortices_x(i_plate,i_vortex),vortices_z(i_plate,i_vortex),ut(IT),
                        vortices_g,vortices_x,vortices_z,wake_g,wake_x,wake_z,limit_vortices_wake);
387                 u_press(i_plate,i_vortex)=u;
388                 w_press(i_plate,i_vortex)=w;
389                 %with sum of vorticities up to current vortex
390                 L_prime(i_plate,i_vortex)=rho*u_press_prev(i_plate,i_vortex)*vortices_g_prev(i_plate
                        ,i_vortex) + rho*c_cell(i_plate)*cos(alpha(i_plate))*sum((vortices_g(i_plate,1:
                        i_vortex)-vortices_g_prev(i_plate,1:i_vortex)))/dt;
391                 D_prime(i_plate,i_vortex)=-rho*w_press_prev(i_plate,i_vortex)*vortices_g_prev(
                        i_plate,i_vortex) + rho*c_cell(i_plate)*sin(alpha(i_plate))*sum((vortices_g(
                        i_plate,1:i_vortex)-vortices_g_prev(i_plate,1:i_vortex)))/dt;
392                 %calculation of moment.
393                 %(Force (D*i + L*k) dot-product normal vector) times distance.
394                 x=c_cell(i_plate)*(i_vortex-1)+(1/4)*c_cell(i_plate);
395                 Lx(i_plate,IT-1)=Lx(i_plate,IT-1) +(dot([D_prime(i_plate,i_vortex),L_prime(i_plate,
                        i_vortex)],[sin(alpha(i_plate)),cos(alpha(i_plate))] ))*x;
396
397             end
398             L(i_plate,IT-1)=sum(L_prime(i_plate,:));
399             D(i_plate,IT-1)=sum(D_prime(i_plate,:));
400             Cl(i_plate,IT-1)=L(i_plate,IT-1)./(rho*.5*(u_start^2)*c(i_plate));
401             Cd(i_plate,IT-1)=D(i_plate,IT-1)./(rho*.5*(u_start^2)*c(i_plate));
402             Cm_LE(i_plate,IT-1)=Lx(i_plate,IT-1)./(rho*.5*(u_start^2)*(c(i_plate))^2); %at LE
403         end
404     end
405
406     %% WAKE ROLLUP
407     %First the movements are calculated. After all wake
408     %displacements are found, then they are moved.
409     for i_plate=1:n_plates
410         for index_wake=1:limit_vortices_wake
411             [u_wru(i_plate,index_wake), w_wru(i_plate,index_wake)]=VFIELD(wake_x(i_plate,index_wake)
                    ,wake_z(i_plate,index_wake),ut(IT),vortices_g,vortices_x,vortices_z,wake_g,wake_x,
                    wake_z,limit_vortices_wake);
412
```

```
413         end
414     end
415     for i_plate=1:n_plates
416         wake_x(i_plate,1:limit_vortices_wake)=wake_x(i_plate,1:limit_vortices_wake)+u_wru(i_plate,1:
                limit_vortices_wake)*dt;%Vectorized version of a loop
417         wake_z(i_plate,1:limit_vortices_wake)=wake_z(i_plate,1:limit_vortices_wake)+w_wru(i_plate,1:
                limit_vortices_wake)*dt;%
418     end
419
420     %% PROGRESS PLOTS TO FILE
421     %Saves a progress figure every IT_PIC iterations.
422
423     if rem(IT,IT_PIC)==0
424         scrsz = get(0,'ScreenSize');
425         fhandle=figure('Position',[1 1 scrsz(3) scrsz(4)*.45]);
426         for iplate=1:n_plot_plates
427             colorpalette=redblue(limit_vortices_wake);
428             [~,wake_indices_ordered]=sort(wake_g(iplate,1:limit_vortices_wake));
429             %wake_order is an array of indices, ordered so that the
430             %corresponding values in wake_g to those ordered indices go from
431             %lowest to largest.
432             plot(vortices_x(iplate,:),vortices_z(iplate,:),'k--o','MarkerSize',2,'MarkerFaceColor'
                    ,[0 0 0])
433             hold on %Hold on here means that plot above deletes previous step.
434             plot(coll_x(iplate,:),coll_z(iplate,:),'k--o','MarkerSize',2,'MarkerFaceColor',[0.5 0.5
                    0.5])
435             for i_wakepoint=1:limit_vortices_wake %Takes all wake points
436                 %Now points are plotted in order, from lowest vorticity to
437                 %highest. This is why wake_indices_ordered(i_wakepoint) is
438                 %taken as index of wake_g, since this will fetch them
439                 %automatically ordered by their value.
440                 if abs(wake_g(iplate,wake_indices_ordered(i_wakepoint))) <1/12*max(wake_g(iplate,:))
                        %Cheat so that vortices with near-zero vorticity show up in white.
441                     plot(wake_x(iplate,wake_indices_ordered(i_wakepoint)),wake_z(iplate,
                            wake_indices_ordered(i_wakepoint)),'o-','MarkerSize',2,'MarkerFaceColor',[1
                            1 1],'MarkerEdgeColor',[.7 .7 .7])
442                 else
443                     plot(wake_x(iplate,wake_indices_ordered(i_wakepoint)),wake_z(iplate,
                            wake_indices_ordered(i_wakepoint)),'o-','MarkerSize',2,'MarkerFaceColor',
                            colorpalette(i_wakepoint,:),'MarkerEdgeColor',colorpalette(i_wakepoint,:))
444                 end
445             end
446         end
447         time=dt*(IT-1);
448         title(sprintf('$\\alpha=%i$, $H=%.1f$, $D=%i$, $dU=%.1f$, $T_{gust}=%.2f$, $t=%.2f$',
                alphadeg(1),h(1),d(2),dUgust,Tgust,time),'Fontsize',16,'interpreter','latex')
449         xlabel('x/c','Fontsize',16,'interpreter','latex')
450         ylabel('z/c','Fontsize',16,'interpreter','latex')
451         if ground_effect~=0
452             ylim([0.5*h(1),1.2*h(1)+c(1)*sin(alpha(1))])
453         end
454         axis equal
455         figname=sprintf('Figure-A%.1f_H%.1f_D%.1f_dUgust%.1f_Tgust%.1f_Tf%.3f_T%.3f.pdf',alphadeg(1)
                ,h(1),d(2),dUgust,Tgust,final_time,time);
456         saveas(fhandle,figname,'pdf')
457         close
458     end
459
460
461     %% PROGRESS PLOTS
462     if plots_on==1 %Plots the progress of the wake, and the plate vorticity.
463         figure (fhandle_plots_on)
464         subplot(311) %progress of wake in geometry
465         for iplate=1:n_plot_plates
466             colorpalette=redblue(limit_vortices_wake);
467             [~,wake_indices_ordered]=sort(wake_g(iplate,1:limit_vortices_wake));
468             %wake_order is an array of indices, ordered so that the
469             %corresponding values in wake_g to those ordered indices go from
470             %lowest to largest.
471             plot(vortices_x(iplate,:),vortices_z(iplate,:),'k--o','MarkerSize',6,'MarkerFaceColor'
                    ,[0 0 0])
472             hold on %Hold on here means that plot above deletes previous step.
473             plot(coll_x(iplate,:),coll_z(iplate,:),'k--o','MarkerSize',6,'MarkerFaceColor',[0.5 0.5
                    0.5])
474             for i_wakepoint=1:limit_vortices_wake %Takes all wake points
475                 %Now points are plotted in order, from lowest vorticity to
476                 %highest. This is why wake_indices_ordered(i_wakepoint) is
477                 %taken as index of wake_g, since this will fetch them
```

```
478                 %automatically ordered by their value.
479                 if abs(wake_g(iplate,wake_indices_ordered(i_wakepoint))) <1/20*max(wake_g(iplate,:))
                          %Cheat so that vortices with near-zero vorticity show up in white.
480                     plot(wake_x(iplate,wake_indices_ordered(i_wakepoint)),wake_z(iplate,
                            wake_indices_ordered(i_wakepoint)),'o-','MarkerSize',5.5,'MarkerFaceColor'
                            ,[1 1 1],'MarkerEdgeColor',[.7 .7 .7])
481                 else
482                     plot(wake_x(iplate,wake_indices_ordered(i_wakepoint)),wake_z(iplate,
                            wake_indices_ordered(i_wakepoint)),'o-','MarkerSize',5.5,'MarkerFaceColor',
                            colorpalette(i_wakepoint,:),'MarkerEdgeColor',colorpalette(i_wakepoint,:))
483                 end
484             end
485         end
486         title('Position of vortices')
487         legend('vortices','collocation points','wake vortices')
488         hold off
489         if ground_effect~=0
490             ylim([0.5*h(1),1.2*h(1)+c(1)*sin(alpha(1))])
491         end
492         axis equal
493
494         subplot(312) %Vorticity of plates
495         hold off
496         for iplate=1:n_plot_plates
497             plot(vortices_x(iplate,:),vortices_g(iplate,:),'b*')
498             hold on
499         end
500         title('Vorticity distribution on the panels')
501         xlabel('X-position')
502         ylabel('Vortex vorticity')
503         for iplate=1:n_plot_plates %Wake vorticity
504             subplot(3,n_plot_plates,(n_plot_plates)*2+iplate)
505             plot(1:max_vortices_wake,-wake_g(iplate,:),'g')
506             title('Wake vorticity')
507             xlabel('Wake index')
508             ylabel('Vorticity [-\Gamma]')
509         end
510     end %end of IF plots==ON
511
512 end % MAIN TIME LOOP END
513 %%time vector and velocity as output. Size reduced as forces available till
514 %%IT-1 only
515 time_vector=0:dt:dt*(n_iterations-2); %-1 for (IT-1) and -1 because starts at 0 not dt.
516 ut(end)=[]; %Reduces size of ut eliminating last position.
517 %% Post Process Data before saving
518 Cl_max=max(Cl,[],2); %Takes maximum along dim2, thus finding for each plate
519 Cd_max=max(Cd,[],2);
520 Cm_LE_max=max(Cm_LE,[],2);
```

```
1  function [u,w] = VOR2D(x,z,x1,z1,vor_g)
2  %VOR2D Calculates influence of vortex located at (x1,z1) on point (x,z)
3  %   VOR2D Calculates the influence of vortices with positions x1, z1 and
4  %   intensities vor_g (These may be vectors) on point with coordinates
5  %   (x,z). VOR2D returns the components of the velocity u and w.
6
7      RX=x-x1;%distance from affected point x z
8      RZ=z-z1;
9      R=sqrt(RX.^2+RZ.^2);
10
11     V=0.5./pi.*vor_g./R;
12     u=V.*(RZ./R);
13     w=V.*(-RX./R);
14
15 self_indices=find(R<=0.001); %Self-induced velocity eliminated
16     u(self_indices)=0;
17     w(self_indices)=0;
18
19     u=sum(u);
20     w=sum(w);
21 end
```

```
1  function [u,w] = VOR2Dvv(x,z,x1,z1,vor_g)
2  %VOR2Dvv Calculates influence of vortices located at (x1,z1) on point (x,z)
3  %   VOR2Dvv (for vector to vector) Calculates the influence of vortices
4  %   with positions x1, z1 and intensities vor_g (These may be vectors) on
5  %   point with coordinates (x,z). VOR2Dvv returns the components of the
6  %   velocity u and w that EACH vortex on the x1, z1 array produces. Not
```

```
%   their sum, as VOR2D does.
%
% See VOR2D

    u=0.0;
    w=0.0;
    RX=x-x1;%distance from affected point x z
    RZ=z-z1;
    R=sqrt(RX.^2+RZ.^2);

    V=0.5./pi.*vor_g./R;
    u=V.*(RZ./R);
    w=V.*(-RX./R);


self_indices=find(R<=0.001); %Self-induced velocity eliminated
    u(self_indices)=0;
    w(self_indices)=0;
end
```

```
function [u,w] = VFIELD(x,z,Uinf,vortices_g,vortices_x,vortices_z,wake_g,wake_x,wake_z,
    limit_vortices_wake)
%VFIELD Calculates the local velocity at x, z in a potential flow field
%with free-stream velocity Uinf, and affected by thin bodies and wakes,
%represented by vortices_g/x/z and wake_g/x/z.
%   VFIELD uses VOR2D to assemble a loop that considers all vortices on the
%   game (plates + wakes) and returns the velocity that the is to be found
%   at a point x, z. In order to achieve this the freestream velocity has
%   to be known.

%Extracting parameters
n_plates=size(vortices_g,1);
u=0;
w=0;

for plate=1:n_plates
 % plates influence
        [uu,ww]=VOR2D(x,z,vortices_x(plate,:),vortices_z(plate,:),vortices_g(plate,:));
        u=u+uu;
        w=w+ww;
 % wakes influence
        [uu,ww]=VOR2D(x,z,wake_x(plate,1:limit_vortices_wake),wake_z(plate,1:limit_vortices_wake),
            wake_g(plate,1:limit_vortices_wake));
        u=u+uu;
        w=w+ww;
end
u=u+Uinf;
end
```

Additionally, a colormap function shared by Adam Auton [12] on the MATLAB Central File Exchange has been used for coloring figures.

Appendix D

Budget

This appendix presents an estimate of the total costs of the research project. The elements that have been accounted for are:

- Hardware costs
- Software costs
- Personnel expenses

Costs associated to resources related to the academic institution have not been estimated, such as the personnel cost of the advisor or the cost of use of university facilities.

For the hardware, the personal computer has been priced at 1229€, sale price of an entry line MacBook Pro according to Apple's Spanish website. Its amortization cost related to the project has been estimated according to:

$$C = \frac{\text{months of use}}{\text{lifespan}} \times \text{sale price} \times \% \text{ dedication} \tag{D.1}$$

The computer has been used for 6 months, and its estimated lifespan is 5 years or 60 months. Percentage of dedication to the project has been 100%. Final costs associated to hardware amount thus to 122.90€.

Software costs amount to 500€, which is the cost of an individual academic-use MATLAB license in Spain, according to the official website of the company that commercializes it, the MathWorks.

Personnel expenses are composed of engineering salary and displacement costs. Displacement costs account for 90 displacements to the academic institution, priced at 8€ each. They add up to 720€.

The base annual engineering salary for an Engineering Degree holder in Spain is, according to the Spanish "XVI Convenio colectivo nacional de empresas de ingeniería y oficinas de estudios técnicos" which establishes basic labor conditions, of 17,038.62€ [13]. Considering 6 months of full-time dedication would amount to 8,519.31€.

Final estimated costs of the project are synthesized in the following table. Total cost is **9,862.21€**.

Hardware costs	122.90€	Personal computer
Software costs	500.00€	MATLAB license
Personnel expenses	720.00€	Displacements
	8,519.31€	Engineering hours
Total cost	9,862.21€	

Bibliography

[1] J. Katz and A. Plotkin, *Low-speed aerodynamics*, vol. 13. Cambridge University Press, 2001.

[2] J. Katz, "A discrete vortex method for the non-steady separated flow over an airfoil," *Journal of Fluid Mechanics*, vol. 102, pp. 315–328, 1981.

[3] H. Aziz and R. Mukherjee, "Unsteady aerodynamics of suddenly accelerated multiple airfoils," *Proceedings of the 13th Asian Congress of Fluid Mechanics*, 2010.

[4] A. L. Sánchez, "Introduction to fluid mechanics." University lecture notes, 2011.

[5] E. L. Houghton, P. Carpenter, S. Collicott, and D. Valentine, *Aerodynamics for Engineering Students*. Elsevier, 2012.

[6] B. Etkin and L. D. Reid, *Dynamics of flight: stability and control*. Wiley New York, 1996.

[7] A. L. Sánchez, "Fluid mechanics." University lecture notes, 2012.

[8] J. R. Holton and G. J. Hakim, *An introduction to dynamic meteorology*. Elsevier Academic Press, 2004.

[9] R. J. LeVeque, *Finite volume methods for hyperbolic problems*, vol. 31. Cambridge University Press, 2002.

[10] "The mathworks inc, matlab ©: Programming fundamentals." Retrieved January 2014 from `http://www.mathworks.com/help/pdf_doc/matlab/matlab_prog.pdf`, 2013.

[11] Y.-c. Fung, *An introduction to the theory of aeroelasticity*. Courier Dover Publications, 2002.

[12] A. Auton, "Red blue colormap." MATLAB Central File Exchange. Retrieved June 2014 from `http://www.mathworks.com/matlabcentral/fileexchange/25536-red-blue-colormap`, 2009.

[13] "Resolución 16097 de 29 de septiembre de 2011 de la direccion general de trabajo. "xvi convenio colectivo nacional de empresas de ingeniería y oficinas de estudios técnicos"." Boletín Oficial del Estado, 13 de octubre de 2011. Retrieved June 2014 from `http://www.boe.es/boe/dias/2011/10/13/pdfs/BOE-A-2011-16097.pdf`., 2011.