POLITECNICO DI MILANO

FACOLTÀ DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

Corso di Laurea Magistrale in Mathematical Engineering

A 3D Adaptive Boundary Element Method for Potential Flow with Nonlinear Kutta Condition

Advisor: Prof. Nicola PAROLINI
Co-Advisor: Dott. Andrea MOLA

Author:
Filippo SACCO
Matr. 833678

Anno Accademico 2017 - 2018

Contents

List of Figures

List of Tables

Abstract

Nowadays computational fluid dynamics has reached a high level of importance due to the possibility of applying it in several branches of science and engineering, concerning for example aerodynamics or naval applications. There are, however, some complications arising from the complexity of the problem to be solved. The governing equations of the physical model, namely the Navier-Stokes equations, are computational expensive to solve. One way to avoid this obstacle is provided by simplified flow models. A classical simplified model is based on the potential flow theory, where an elliptic Laplace equation for the flow potential is derived from the Navier-Stokes equations by introducing suitable simplifying hypothesis. The computational complexity is reduced, still keeping the capability of the model to describe the underlying physical reality in relevant operating conditions.

In this work we analyze and implement a Boundary Element Method to solve 2D and 3D potential flows around lifting bodies (airfoils and wings), analyzing different geometrical configurations. This method is powerful because it decreases the dimensionality of the problem transforming it from a 3D problem into a 2D problem defined over the boundary of the domain, thus reducing the degrees of freedom of our model. The Boundary Element Method approach yields a correct treatment of problems in infinite domains, however it works only for small angles of attack, for which the potential flow assumption is adequate. In potential flow models the viscosity is neglected, so that only inviscid features, as lift, can be captured but not viscous effects, as turbulence or boundary layers. The numerical model developed in this thesis is a modification of the program π-`BEM` [1] based on an existing open source finite element library called `deal.II` [2, 3]. The existing solver was able to deal with a potential flow for non lifting bodies in three dimensions. We extended it in order to treat the simulation of lifting bodies also in the two dimensional cases. As the problem in this case becomes nonlinear, a suitable iterative procedure to treat the nonlinearity has been introduced.

A key feature of the model is the possibility to adapt the grid by refining and projecting it over given CAD geometries, so that the input is not the mesh describing the domain, but an initial mesh, the curves and surfaces describing the geometry and a list of parameters that specifies how much and with which strategy the initial mesh should be refined. This possibility is useful in order to refine the grid in the regions where we want to capture better the solution.

A new way to impose the Kutta condition is proposed. This condition states that

the jump of pressure at the trailing edge has to be null. This is a nonlinear condition and different strategies have been proposed in literature such as the recollocation of the trailing edge points upwind or normally to the wake in order to desingularize the integrals at trailing edge and to have distinct nodes at the trailing edge [4], the Kutta condition in term of velocity [5], the decomposition of the velocity at the trailing edge in its components [6] and the linearization of the Kutta condition [7].

In this work, we present a method that is able to compute the pressure at the trailing edge and solve the nonlinear problem, so that neither grid node recollocation nor linearization of the Kutta condition are required.

Introduzione

In questa tesi vogliamo simulare il flusso attorno a una superficie portante in due e tre dimensioni. Il modello che abbiamo deciso di utilizzare è basato sulla teoria di flusso a potenziale. Questa teoria risulta di grande interesse nel campo aerodinamico, ma bisogna porre particolare attenzione quando si introduce la scia e quando si vuole imporre la condizione di Kutta, cosa che ci permette di trattare corpi portanti. Questa condizione, la quale afferma che non c'è salto di pressione al bordo di uscita, viene trattata con uno schema non lineare in modo da poterla imporre. Per l'approssimazione numerica del problema abbiamo utilizzato un metodo agli elementi al contorno (boundary element method, BEM) basato sulla formulazione integrale delle equazioni governanti il modello fisico del problema. Questo metodo risulta particolarmente efficace in quanto la risoluzione del problema non avviene all'interno del dominio, ma sul suo bordo. Grazie a questa potenzialità è possibile trattare casi di flusso esterno dove il dominio ha estensione infinita.

La peculiarità del nostro lavoro consiste nell'imposizione della condizione di Kutta, che viene trattata in modo differente da quanto fatto finora in letteratura [4, 7, 5]. Vedremo come imporre questa condizione ci porti a dover risolvere un sistema di equazioni non lineare, cosa che comporta l'uso di tecniche adeguate per poter ottenere la soluzione del nostro problema. La parte più importante, infatti, è riuscire a imporre la condizione di Kutta così come è enunciata, ovvero che il salto della pressione al bordo d'uscita di un profilo portante debba essere nullo. Essendo questa una condizione non lineare, per imporla dobbiamo utilizzare un metodo di Newton in cui dovremo andare a calcolare il residuo e lo Jacobiano del sistema di equazioni.

Un'altra potenzialità del nostro algoritmo consiste nel potere dare in input una superficie CAD con i suoi bordi e una griglia iniziale. Automaticamente, specificando delle strategie di raffinamento, si ottiene la mesh finale sulla quale viene poi risolto il problema. Questa possibilità fa sì che non dobbiamo creare a parte una mesh per ogni profilo CAD che vogliamo utilizzare nelle simulazioni, ma è invece possibile eseguire l'algoritmo applicandolo a casistiche diverse e automaticamente otteniamo la griglia su cui lavorare. Avendo la possibilità di raffinare la griglia all'interno del nostro algoritmo, è possibile attuare una strategia di raffinamento adattivo, in cui, dopo aver risolto il problema per una mesh poco fitta, essa viene raffinata in alcune zone per poter meglio cogliere l'andamento della soluzione. È possibile iterare questo procedimento per più volte, in modo da riuscire a calcolare soluzioni sempre migliori.

Riportiamo di seguito la struttura dell'elaborato. Nel capitolo 1 viene descritto il modello fisico e il procedimento per passare dalle equazioni di Navier-Stokes a quelle di Laplace e Bernoulli, specificando sotto quali ipotesi questa trasformazione sia possibile. In secondo luogo è illustrato il procedimento per passare dalla formulazione differenziale a quella integrale, in cui la risoluzione del problema passa dal dominio al suo bordo con la conseguente riduzione della dimensionalità. Vengono poi introdotte adeguate condizioni al contorno del profilo e condizioni inerenti la trattazione della scia e del bordo d'uscita, in particolar modo la condizione di Kutta. Vengono dunque analizzate le equazioni finali da risolvere e, una volta ottenuta la soluzione, in che modo ottenere le altre quantità fisiche di nostro interesse come la pressione.
Nel capitolo 2 è trattato il problema presentato nel precedente capitolo introducendone un'opportuna discretizzazione numerica. Sono introdotti quindi opportuni spazi funzionali in cui viene risolto il problema. Senza aver ancora imposto la condizione di Kutta si ottiene così un sistema lineare di equazioni. Nel momento in cui viene imposta la continuità della pressione al bordo d'usicta allora viene illustrato in che modo riuscire a ricavare il gradiente del potenziale e il metodo di Newton che ci permette di imporre la condizione di Kutta. Nelle ultime sezioni viene trattato lo spostamento della scia e il calcolo delle velocità su di essa, dato che compaiono degli integrali ipersingolari di difficile trattazione.
Nel capitolo 3 viene illustrato il procedimento di generazione della griglia sia in due che in tre dimensioni, con particolare attenzione alla trattazione dei nodi doppi e degli hanging nodes.
Nel capitolo 4 vengono presentati i risultati ottenuti in questo lavoro. Dopo aver analizzato le proprietà di convergenza del metodo rispetto alla griglia di calcolo e alla discretizzazione della scia, i risultati sono confrontati con quelli presenti in letteratura. In tre dimensioni vengono presentati anche casi con geometrie più complesse o con la presenza di più corpi.

Il lavoro svolto ha mostrato come il modello proposto è in grado di imporre della condizione di Kutta in modo adeguato e come le potenzialità legate al raffinamento della griglia all'interno dell'algoritmo siano molteplici.
Le evoluzioni di quanto presentato in questo elaborato sono parecchie: la trattazione del problema non stazionario, la soluzione del problema per profili sottili, la parallelizzazione dell'algoritmo, l'introduzione di un metodo ad alto ordine e l'integrazione di questo metodo con l'interazione barca-onde [8] per la simulazione di timoni e derive.

Introduction

In this thesis we want to simulate a flow on a lifting surface in two and three dimensions. The model we decided to use is based on the potential flow theory. This theory is of great interest in the aerodynamical field, but it is necessary to have some attentions on the introduction of the wake and the imposition of the Kutta condition, that will allow to deal with lifting bodies. This condition, that states that the pressure is continuous at the trailing edge, will be treated in a nonlinear way. For the numerical discretization of the problem we use the boundary element method, based on the integral formulation of the equations governing the physical model of the problem. This method is particularly effective because the solution of the problem is not computed in the domain, but on its boundary. Thanks to this potentiality, it will be possible to treat external flow cases where the domain has infinite extension. The peculiarity of our work will be the imposition of the Kutta condition, that will be treated in a different way than what is present in literature [4, 7, 5]. This condition is a nonlinear equation, that it is often linearized in order to have a final linear system of equations. In our work we will impose this condition adopting a fully nonlinear strategy. We will see how imposing this condition will lead us to solve a nonlinear system of equations, involving the use of adequate techniques in order to obtain the solution of the problem. The most important thing, in fact, will be imposing the Kutta condition as it is stated, that is the pressure jump at the trailing edge must be null. Since this is a nonlinear condition, in order to solve it we will use a Newton method in which we will have to specify the residual and the Jacobian of the system of equations. Another potentiality of our algorithm consists in being able to give in input CAD surfaces and edges and an initial grid and automatically. After having specified the refinement strategies in fact, we will obtain the final mesh on which the problem will be solved. This possibility will ensure that we don't have to create previously a mesh for every CAD surface that we want to use in the simulations, but it will be possible to execute the algorithm applying it to different cases and automatically we will obtain the grid on which the solution will be computed. Having the possibility to refine the grid during the execution of our algorithm, it will be possible to use a strategy of adaptive refinement, in which, after having solved the problem with a coarse mesh, the grid will be refined in some regions in order to improve the accuracy of the solution. This procedure can be iterated more times, in order to be able to obtain always better solutions.

State of art

Many fields of engineering benefit from accurate and reliable solvers of Boundary Integral equations. The most noteworthy equations that admit this formulation are the Laplace equation, the Helmholtz equation and the Stokes system. This method has been applied to problems involving hydrodynamic flows ([5, 9, 10, 8, 11, 12, 13, 14]), flows around aerodynamic lifting bodies ([4, 15, 16, 6, 17]), structural mechanics ([18]), electrostatics ([19]), quantum mechanics ([20]) and acoustics ([21]). The most convenient aspect of the Boundary Integral formulation is that the solution at each point of the domain is expressed in term of convolutions between a fundamental solution and the boundary trace of the solution or its normal gradient. This convolution is done on the domain's boundary so the method can reduce the dimensionality of the problem. The associate numerical method is the Boundary Element method which is able to reduce the number of degrees of freedom if compared to the more common Finite Elements or Finite Volumes approaches. The reduction of dimensionality however is limited by the fact that the resulting algebraic system is dense, so the computational cost of assembly and the cost of solving a linear system are $O(n^2)$ flops, since the used method is an iterative GMRES Krylov solver.

In aerodynamic and hydrodynamic applications panel methods have been widely used, since the first work of Hess and Smith (1962) [22]. Different panel methods have been proposed in order to apply them to a variety of engineering aspects. The first methods were based on a velocity formulation in which the boundary condition on the body surface was satisfied through the computation of the velocity. Morino (1974) [6] introduced a panel method based on Green functions, so that the primary unknown is the potential. This formulation is more stable, but with low order panel method, where the potential is assumed constant over a panel, there are significant errors while computing the velocities with finite difference schemes near the trailing edge and at the tip of the lift-generating surface. One improvement to this method was given by Lee et al. in [23, 24] where a higher-order method was implemented employing B-spline basis functions. Since the derivative of the basis functions are known exactly, there is no need to compute the velocity by numerical differentiation, so one limit of the panel method is solved. This procedure however restricts the basis functions to be B-splines. Key ingredient in panel methods, when treating with weakly singular and singular integrals, is shifting the collocation points in order to avoid singular terms as in [25]. In this work there will not be a desingularization of these integrals, but they will be treated with bidimensional Telles, Lachat, Watson or Duffy quadrature formula [26, 27, 28, 29].

When describing a potential flow around aerodynamic lifting bodies, it is necessary to introduce the so-called Kutta condition. This condition, that states that the pressure jump at the trailing edge has to be null, has been widely studied in the literature [30], since its statement is simple but the implications and the possible implementation in a 3D framework are several. In the numerical framework is not easy to impose this condition, since it is nonlinear and because it has to be imposed at the trailing edge,

a physical region where there coexist the body and the wake. In order to avoid the nonlinearity of the problem, some methods introduce a linearization strategy [7, 5, 6] where the null pressure jump is imposed by relations between the velocity's components. In order to avoid the same geometrical location of the trailing edges on the two sides of the wake, some models [4] shift that points upwind or normally to the wake and impose two different boundary integral equations, then, passing to the limit as these points approach the trailing edge, they recover the Kutta condition. Another procedure proposed in [15] consists in expressing the velocity by means of the exact gradients of the basis function in analytical way. This procedure however restricts the treatment only to a limitate number of cases. In our work we show how to impose the Kutta condition in a general way independent on the basis functions, without shifting the trailing edge's nodes and imposing that the pressure jump at the trailing edge should vanish.
One problem arising from the presence of sharp edges is that the normal vectors and so the normal components of the potential gradient have a jump across the interface. For continuous elements a possible solution is the so-called double nodes technique [31] where continuity is preserved on the solution while its normal gradient is allowed to have jump across edges. New challenges come from the developing of Computer Aided Design technologies, in particular the creation of a grid over that geometry specified by user provided files either for FEM [32] or for BEM [33] communities. The techniques used in this work have been presented in [1] and have been extended in this work to 2D configurations. Having the possibility to create a grid in the algorithm, it will be possible to refine the mesh with adaptive strategy as presented in [34].

Thesis structure

The elaborate is structured as follows. In chapter 1 we describe the physical model and the procedure to derive from Navier-Stokes equation the Laplace and Bernoulli equations, specifying under which assumptions this transformation is possible. Then the integral formulation of the potential flow model is derived. We will introduce suitable boundary conditions on the profile and conditions on the treatment of the wake and the trailing edge, in particular the Kutta condition. We will then analyse the final equations that we have to solve and, after having obtained the solution, we will be able to find other physical quantities such as the pressure.
In chapter 2 we will treat the problem presented in the previous chapter bringing it into a numerical framework. We will introduce suitable functional space, in which we will solve our problem, and the collocation method. Without the imposition of the Kutta condition we would obtain a linear system of equations. When we want to impose the pressure continuity at the trailing edge, then we will show how we can obtain the potential gradient and we will illustrate the Newton method that will allow us to impose the Kutta condition. In the last sections we will present how to treat the wake displacement and the computation of the velocities on it, since there will appear

hypersingular integrals of difficult treatment.
In chapter 3 we will present the procedure of grid generation either in two or in three dimensions, with particular attention to the treatment of double nodes and hanging nodes.
In chapter 4 numerical results obtained in this work are presented and discussed. In particular, grid convergence analysis and comparison with available results are used to asses the accuracy of the proposed model.

The work carried out has shown that the imposition of the Kutta condition with this innovative method is adequate and that the refinement of the grid inside the algorithm may have different possible applications.

Chapter 1

Mathematical model: boundary integral equation and Kutta condition

The problem we want to solve consists in finding the velocity and the pressure of the fluid around a lifting body moving at velocity $\mathbf{V}_\infty$. Choosing a reference system fixed with the body, the problem can be recast, as in Figure 1.1, so that the velocity field $\mathbf{V}(\mathbf{x}, t)$ has a farfield condition $\mathbf{V}(\mathbf{x}, t) = \mathbf{V}_\infty$. The wing is oriented so that its elongation direction forms an angle θ (the angle of attack) with respect to wind direction. The equations governing this phenomenon are the Navier-Stokes equations.

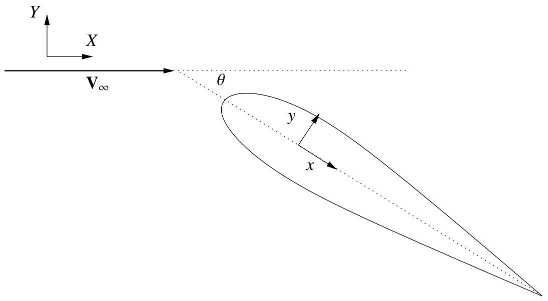


Figure 1.1: 2D visualization of the problem. $\mathbf{V}_\infty$ is the inflow velocity, X and Y is the global coordinates system, x and y is the local coordinates system attached to the body, θ is the angle of attack.

In this chapter we will derive the equations describing the potential flow around a lifting object. Under specific assumptions on the physical model, we will show how the Navier-Stokes equations governing the flow of a Newtonian fluid can be recast into the Laplace equation for the velocity potential field and the Bernoulli equation. This approximation will lead to a simpler problem, which can be useful in the analysis of external flows resulting in a cheaper numerical approach than computational fluid dynamic simulations based on Navier-Stokes.

Starting from a potential flow, we will derive the boundary integral formulation for the potential model, in which we naturally treat the presence of a wake surface de-

taching from the trailing edge and allowing for the generation of a lifting force. Such formulation allows for the definition of a numerical problem which only requires the discretization of the boundary of the fluid domain [35].
Particular attention will be devoted to the importance of the wake vortex sheet in the potential flow theory for lifting bodies [30]. In particular we will throughly discuss the imposition of the Kutta condition which requires that the wake detaches from the trailing edge and that no pressure discontinuity is observed across the wake. This is of capital importance to avoid the D'Alembert paradox: a pure potential flow cannot produce a lifting force.
The physical assumptions on which the potential flow theory is based are

- irrotational velocity field $\nabla \times \mathbf{V} = 0$;
- incompressible flow $\nabla \cdot \mathbf{V} = 0$;
- negligible body forces;
- negligible viscous forces;
- simply connected fluid domain Ω.

1.1 Derivation of the potential flow equations

We now show how to transform the Navier-Stokes equations into the Laplace and Bernoulli equations defining the potential flow around a body. For a complete treatment, see e.g. [36]. Consider a spatial domain $\Omega \subseteq \mathbb{R}^d$ with $d = 2, 3$ and the incompressible Navier-Stokes equations

$$\frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \nabla)\mathbf{V} + \frac{\nabla p}{\rho} - \nu \nabla^2 \mathbf{V} = \mathbf{g}, \tag{1.1}$$

$$\nabla \cdot \mathbf{V} = 0, \tag{1.2}$$

where $\mathbf{V}$ denotes the velocity of the fluid, p the pressure, ρ the fluid density, ν the kinematic viscosity and $\mathbf{g}$ the external forces for unit of mass, like gravity acceleration. The equation (1.1) is the momentum balance law whereas the equation (1.2) is the mass conservation law that is also a condition of incompressibility.
Under the assumptions of potential flow, we can neglect the viscous term $-\nu\nabla^2\mathbf{V}$ and we obtain the incompressible Euler equations

$$\frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V} \cdot \nabla)\mathbf{V} + \frac{\nabla}{p}\rho = \mathbf{g}, \tag{1.3}$$

$$\nabla \cdot \mathbf{V} = 0. \tag{1.4}$$

As the Reynolds number is defined as $Re = \frac{UL}{\nu}$, where U is the velocity of the fluid with respect to the object, L is the characteristic linear dimension and ν is the kinematic viscosity, assuming negligible viscosity corresponds to take the limit of Reynolds number that goes to ∞. Obviously, neglecting the kinematic viscosity the model will not

be able to describe viscous effects such as turbulence or boundary layers, but it will model only inviscid phenomena.
Moreover in the potential theory we assume that the fluid motion is irrotational, that is $\nabla \times \mathbf{V} = 0$, so there exists a scalar function (called potential) Φ such that $\nabla\Phi = \mathbf{V}$ and the governing equations become

$$\frac{\partial\Phi}{\partial t} + \frac{1}{2}|\nabla\Phi|^2 + \frac{p}{\rho} + \chi = C(t), \tag{1.5}$$

$$\nabla^2\Phi = 0, \tag{1.6}$$

where χ is the external force potential, such that $\mathbf{g} = -\nabla\chi$ and $C(t)$ is a time dependent arbitrary function. Equation (1.5) is called Bernoulli equation and equation (1.6) is the Laplace equation. In particular we want to consider only steady flows and we can consider negligible body forces, since in most aerodynamics problems the gravity force does not play a relevant role, so the Bernoulli equation becomes

$$\frac{1}{2}|\nabla\Phi|^2 + \frac{p}{\rho} = C, \tag{1.7}$$

where C is a time independent constant. We can find the potential Φ solving the Laplace equation, then we obtain the pressure from the Bernoulli equation.
Given the linearity of the gradient and Laplace operator, the potential Φ can be written as $\Phi = \phi_\infty + \phi$, where ϕ denotes the perturbation potential induced by the presence of the wing and $\phi_\infty = \mathbf{V}_\infty \cdot \mathbf{x}$ with $\mathbf{V}_\infty$ the inflow velocity of the fluid and $\mathbf{x}$ the spatial coordinates, thanks to the linearity of the Laplace operator and since $\nabla\phi_\infty = \mathbf{V}_\infty$ and $\nabla^2\phi_\infty = 0$, we obtain

$$\nabla^2\Phi = 0 \Longleftrightarrow \nabla^2\phi = 0, \tag{1.8}$$

$$\frac{1}{2}|\nabla\Phi|^2 + \frac{p}{\rho} = C \Longleftrightarrow \frac{1}{2}|\nabla\phi + \mathbf{V}_\infty|^2 + \frac{p}{\rho} = C. \tag{1.9}$$

1.2 Boundary integral equation

Since the Laplace equation (1.6) admits a fundamental solution, see [37], it can be reformulated in a boundary integral equation, in order to reduce the dimensionality of the problem so that we will be able to use the boundary element method.
We consider a body defined by a bounded open set $B \subset \mathbb{R}^d$ with $d = 2, 3$ with Lipschitz boundary Γ. B is surrounded by another Lipschitz boundary Γ_∞ (non intersecting Γ). We call Ω the open set with $\partial\Omega = \Gamma \bigcup \Gamma_\infty$ as shown in Figure 1.2.

We multiply the Laplace equation by an arbitrary function G and integrate over Ω, thus we obtain this formulation

$$\int_\Omega -\nabla^2\phi G d\Omega = 0 \quad \forall G. \tag{1.10}$$

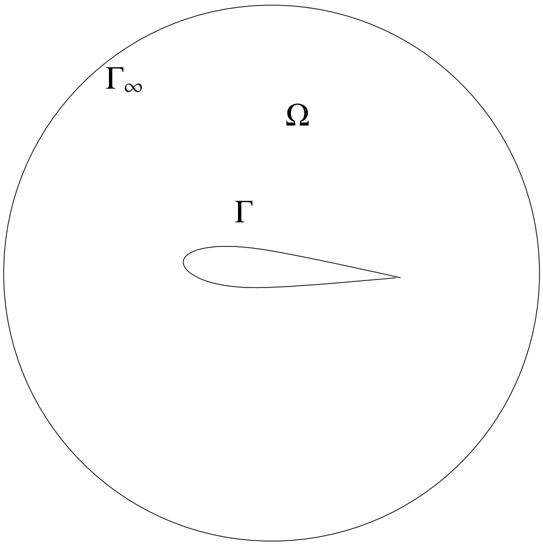


Figure 1.2: Spatial domain.

We integrate by parts twice and we exploit the divergence theorem

$$\int_\Omega \nabla\phi \cdot \nabla G d\Omega - \int_{\partial\Omega} G\frac{\partial\phi}{\partial n} dS = 0 \quad \forall G, \tag{1.11}$$

$$-\int_\Omega \phi\nabla^2 G d\Omega + \int_{\partial\Omega} \phi\frac{\partial G}{\partial n} dS - \int_{\partial\Omega} G\frac{\partial\phi}{\partial n} dS = 0 \quad \forall G, \tag{1.12}$$

where $\frac{\partial\phi}{\partial n} = \nabla\phi \cdot \mathbf{n}$, $\mathbf{n}$ is the outward normal to the domain and $\partial\Omega = \Gamma \cup \Gamma_\infty$.
We impose two conditions on Γ_∞, in order to be able to treat unbounded domains. In particular we want

$$\phi \to 0 \text{ and } \frac{\partial\phi}{\partial n} \to 0, \tag{1.13}$$

so the integral contributions on Γ_∞ becomes null and equation (1.12) becomes

$$-\int_\Omega \phi\nabla^2 G d\Omega + \int_\Gamma \phi\frac{\partial G}{\partial n} d\Gamma - \int_\Gamma G\frac{\partial\phi}{\partial n} d\Gamma = 0 \quad \forall G. \tag{1.14}$$

Now we choose the function $G : \mathbb{R}^d \to \mathbb{R}$ with $d = 2, 3$ such that

$$-\nabla^2 G(\mathbf{x} - \mathbf{y}) = \delta(\mathbf{x} - \mathbf{y}), \tag{1.15}$$

which is the so-called free space Green's function and represents the fundamental solution of the Laplace operator, that is the potential $\Phi(\mathbf{x})$ evaluated in $\mathbf{x}$ given a pointwise source $\delta(\mathbf{x} - \mathbf{y})$ concentrated in $\mathbf{y}$.
In the 2D case, G is given by

$$G(\mathbf{x} - \mathbf{y}) = -\frac{1}{2\pi} \ln(|\mathbf{x} - \mathbf{y}|) \tag{1.16}$$

$$\frac{\partial G}{\partial n}(\mathbf{x} - \mathbf{y}) = \frac{\mathbf{x} - \mathbf{y}}{2\pi|\mathbf{x} - \mathbf{y}|^2} \cdot \mathbf{n}, \tag{1.17}$$

while in the 3D case, we have

$$G(\mathbf{x}-\mathbf{y}) = \frac{1}{4\pi|\mathbf{x}-\mathbf{y}|} \tag{1.18}$$

$$\frac{\partial G}{\partial n}(\mathbf{x}-\mathbf{y}) = -\frac{\mathbf{x}-\mathbf{y}}{4\pi|\mathbf{x}-\mathbf{y}|^3}\cdot\mathbf{n}. \tag{1.19}$$

By substituting the Green's function in (1.14), we obtain

$$\phi(\mathbf{x}) = \int_\Gamma G(\mathbf{x}-\mathbf{y})\frac{\partial\phi}{\partial n}(\mathbf{y})d\Gamma - \int_\Gamma \phi(\mathbf{y})\frac{\partial G}{\partial n}(\mathbf{x}-\mathbf{y})d\Gamma \text{ with } x\in\Omega, \tag{1.20}$$

where the integral terms are only on the body surface, so we have reduced the dimensionality of the problem.
The first term in the right-hand side of (1.20) is referred to as *single layer potential* (SLP) while the second one as *double layer potential* (DLP). If we know the potential $\phi(\mathbf{x})$ and it's normal derivative $\frac{\partial\phi}{\partial n}(\mathbf{x})$ on Γ we can compute the potential in any point $\mathbf{x}$ in the domain Ω. The SLP is continuous across Γ, instead the DLP has a jump discontinuity, in particular when $\mathbf{x}$ approaches a smooth surface Γ from inside Ω

$$\lim_{x\to\Gamma^-}\int_\Gamma \phi(\mathbf{y})\frac{\partial G(\mathbf{x}-\mathbf{y})}{\partial n}d\Gamma = \int_\Gamma^{PV} \phi(\mathbf{y})\frac{\partial G(\mathbf{x}-\mathbf{y})}{\partial n}d\Gamma + \frac{1}{2}\phi(\mathbf{x}), \tag{1.21}$$

where PV means Cauchy Principal Value defined as

$$\int_\Gamma^{PV} \phi(\mathbf{y})\frac{\partial G(\mathbf{x}-\mathbf{y})}{\partial n}d\Gamma := \lim_{\epsilon\to 0}\int_{\Gamma\backslash B_\epsilon} \phi(\mathbf{y})\frac{\partial G(\mathbf{x}-\mathbf{y})}{\partial n}d\Gamma, \tag{1.22}$$

where B_ϵ is the ball of radius ϵ centered in $\mathbf{x}\in\Gamma$. Then equation (1.20) becomes

$$\frac{1}{2}\phi(\mathbf{x}) = \int_\Gamma G(\mathbf{x}-\mathbf{y})\frac{\partial\phi}{\partial n}(\mathbf{y})d\Gamma - \int_\Gamma \phi(\mathbf{y})\frac{\partial G}{\partial n}(\mathbf{x}-\mathbf{y})d\Gamma \text{ with } x\in\Gamma. \tag{1.23}$$

In general, if Γ is not smooth, the right way to write the equation (1.23) is

$$\alpha\phi(\mathbf{x}) = \int_\Gamma G(\mathbf{x}-\mathbf{y})\frac{\partial\phi}{\partial n}(\mathbf{y})d\Gamma - \int_\Gamma \phi(\mathbf{y})\frac{\partial G}{\partial n}(\mathbf{x}-\mathbf{y})d\Gamma \text{ with } x\in\Gamma, \tag{1.24}$$

where α is the Cauchy Principal Value, which is the fraction of the solid angle by which the point $\mathbf{x}$ sees the domain Ω. This is equivalent to consider the trace of equation (1.20). In this case the kernels $G(\mathbf{x}-\mathbf{y})$ and $\frac{\partial G}{\partial n}(\mathbf{x}-\mathbf{y})$ become weakly singular (but integrable) and singular respectively.
For the integrals to be bounded, $\phi\in H^{\frac{1}{2}}(\Gamma)$ and $\frac{\partial\phi}{\partial n}\in H^{-\frac{1}{2}}(\Gamma)$. Equation (1.24) is called boundary integral equation.

1.3 Boundary conditions

For well posedness, the boundary integral equation must be complemented by a suitable set of boundary conditions. We have already seen in equation (1.13) that it's necessary to require that far from the body the perturbation potential and its normal derivative vanish

$$\lim_{|r|\to\infty} \phi(r) = 0 \text{ and } \lim_{|r|\to\infty} \frac{\partial\phi}{\partial n}(r) = 0$$

in particular we require

$$\phi = \mathcal{O}(\|\mathbf{r}\|)^{-1} \text{ or equivalently } \mathbf{V} = \mathcal{O}(\|\mathbf{r}\|)^{-2}$$

otherwise the integration on Γ_∞ would not be zero. These boundary conditions are imposed on Γ_∞.

On Γ we impose the non penetration condition, that means that the flow does not cross the boundary of the solid body, so the normal component of the total velocity of the flow must be null

$$\frac{\partial\Phi}{\partial n} = 0 \to \mathbf{V}_\infty \cdot \mathbf{n} + \nabla\phi \cdot \mathbf{n} = 0 \to \frac{\partial\phi}{\partial n} = -\mathbf{V}_\infty \cdot \mathbf{n}. \tag{1.25}$$

Now we have reduced the initial problem to the simpler one, so we have to solve:
Given $\mathbf{V}_\infty$, find ϕ such that

$$\nabla^2\phi = 0 \quad \text{in } \Omega \tag{1.26}$$

$$\lim_{|r|\to\infty} \phi(r) = 0 \quad \text{on } \Gamma_\infty \tag{1.27}$$

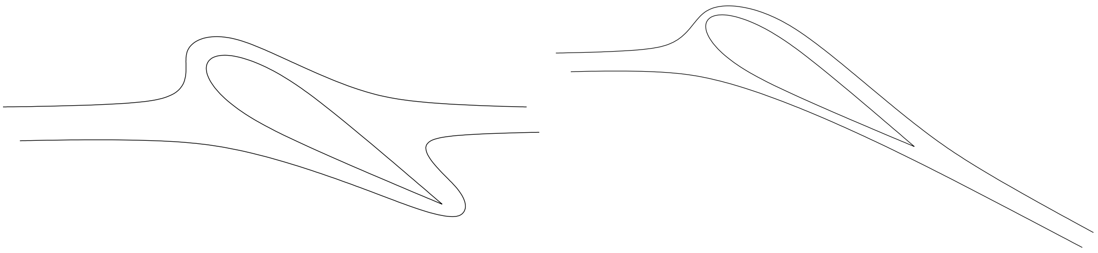
$$\frac{\partial\phi}{\partial n} = -\mathbf{V}_\infty \cdot \mathbf{n} \quad \text{on } \Gamma. \tag{1.28}$$

We stress that the model derived so far is not able to reproduce any kind of lifting effects, see [35]. In section 1.4 we will introduce the Kutta condition and the wake effects to recover such phenomena.

1.4 Wake modeling and Kutta conditions

Even in presence of a lifting body, the solution of this problem has always null circulation (Figure 1.3a) and does not generate lift. This obviously does not represent the physical observations. This problem is called d'Alembert's paradox, according to which for incompressible and inviscid potential flow the aerodynamic force is zero on a body moving at constant velocity.

In the most common practice [38] such problem is overcome thanks to the introduction a wake surface Γ_w in the domain Ω as shown in Figure 1.4. Physically the wake is a thin layer in which high velocity gradients cause a concentration of vorticity and viscous forces [35]. In the lifting body potential theory such vortical layer — which is



(a) Without Kutta condition.

(b) With Kutta condition.

Figure 1.3: Difference between wings without and with the imposition of the Kutta condition.

paramount to lift generation — is introduced as a surface of discontinuity for the potential field. The introduction of this surface makes also the two dimensional domain simply connected.

The wake is usually introduced in the boundary integral formulation as an additional boundary surface Γ_w and we will discuss the spatial collocation of such surface and the boundary conditions prescribed on it. So we divide Γ into Γ_b for the body and Γ_w for the wake. We assume that Γ_b is smooth with a sharp trailing edge and that the wake is a thin line in 2D or surface in 3D detaching from the trailing edge in which is concentrated the vorticity present in the physical wake.

In principle we do not know which is the shape of the wake so its position is an unknown and it influences the solution, so it has be found in a suitable way. In section 2.4 we describe a methodology to find this solution.

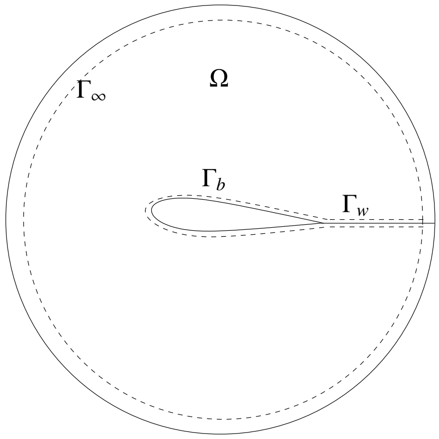


Figure 1.4: 2D spatial domain with the introduction of the wake.

In order to obtain the boundary conditions of the wake, we have to split this thin layer into two boundaries logically distinct, but geometrically identical, so we define

$$\Gamma_{w^+} = \Gamma_w + \epsilon\mathbf{n} \quad \text{and} \quad \Gamma_{w^-} = \Gamma_w - \epsilon\mathbf{n}, \tag{1.29}$$

where $\mathbf{n}$ is the unit normal to Γ_w, such that, when $\epsilon \to 0$ we obtain again Γ_w.
In the same way, the trailing edge is splitted in upper trailing edge TE^+ and lower trailing edge TE^-, two logical distinct points in 2D and curves in 3D, but geometrically identical. So we define

$$TE^+ = TE + \epsilon\mathbf{n} \quad \text{and} \quad TE^- = TE - \epsilon\mathbf{n}, \tag{1.30}$$

such that, when $\epsilon \to 0$ we obtain again the TE, as shown in Figure 1.5.

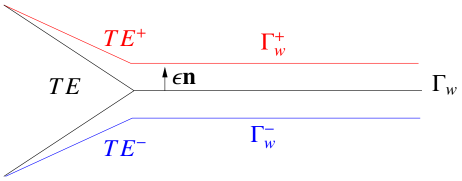


Figure 1.5: Definition of upper and lower trailing edges and Γ_{w^+} and Γ_{w^-}.

A first condition on $\frac{\partial\phi}{\partial n}$ on the upper and lower sides of the wake (obtained applying the mass conservation law across the wake surface) [39] reads

$$\left[\frac{\partial\phi}{\partial n}\right] = 0 \text{ on } \Gamma_w. \tag{1.31}$$

The jump of the normal component of the potential gradient must be null. This ensures there is no mass accumulation on the wake.
On the other hand, we allow discontinuity of the potential

$$\phi(TE^+) \neq \phi(TE^-) \text{ and } \phi(\Gamma_{w^+}) \neq \phi(\Gamma_{w^-}). \tag{1.32}$$

A further condition is derived from the momentum balance law. It states that the pressure must be continuous across the wake

$$p^+ = p^-. \tag{1.33}$$

In order to write this condition in terms of the potential, we pass through the Bernoulli equation (1.5) and we obtain a condition on the Lagrangian derivative of the potential jump on the wake

$$\frac{D_w\,[\phi]}{Dt} = 0, \tag{1.34}$$

where $\frac{D_w[\phi]}{Dt} = \frac{\partial[\phi]}{\partial t} + \mathbf{V}_w \cdot \nabla\,[\phi]$ and $\mathbf{V}_w$ is the wake velocity. In case of steady flows, which are the focus of this work, equation (1.34) implies that the potential jump across the wake must be constant. In particular to fulfill equation (1.34) we can impose that the potential jump across the wake is equal to the one across the trailing edge along a streamline

$$[\phi] = \phi(TE^+) - \phi(TE^-) \text{ on } \Gamma_w. \tag{1.35}$$

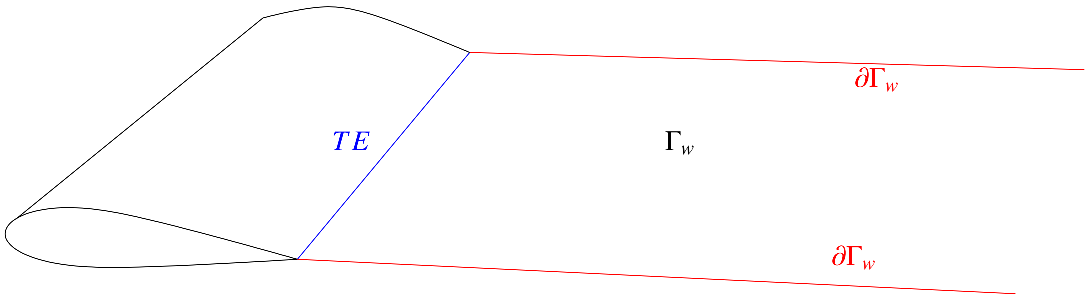


Figure 1.6: Definition of $\partial\Gamma_w$ (red lines) and of the trailing edge (blue line) for 3D problems.

This condition allows to have discontinuity on the velocity across the wake, that is what we see in reality, because the gradient of the velocity is concentrated in the wake, that is only a thin layer in our model, so we can capture this phenomenon as a discontinuity. Moreover equation (1.34) ensures that the wake is a flow surface, id est the wake is tangential to the fluid velocity in each point.

In 3D problems, in order to ensure $\nabla\phi \in L^2_{loc}(\Omega)$ we have to impose the further condition $[\phi]_{\partial\Gamma_w} = 0$ [30]. Where $\partial\Gamma_w$ is intended to be the lateral boundary of Γ_w without considering the trailing edge as shown in Figure 1.6.

The problem now is:

Given $\mathbf{V}_\infty$, find ϕ such that

$$\begin{array}{ll} \nabla^2\phi = 0 & \text{in } \Omega \\ \lim_{|r|\to\infty} \phi(r) = 0 & \text{on } \Gamma_\infty \\ \frac{\partial\phi}{\partial n} = -\mathbf{V}_\infty \cdot \mathbf{n} & \text{on } \Gamma_b \\ \left[\frac{\partial\phi}{\partial n}\right] = 0 & \text{on } \Gamma_w \\ [\phi] = \phi(TE^+) - \phi(TE^-) & \text{on } \Gamma_w \\ [\phi] = 0 & \text{on } \partial\Gamma_w. \end{array} \tag{1.36}$$

It's not sufficient to impose these conditions, since we also have to impose a condition on the trailing edge, that is the Kutta condition: the pressure has to be continuous at the trailing edge.

This condition in the literature is treated in different ways. There are some techniques [4] of recollocation of the geometrical trailing edge points upwind or normally to the wake in order to have two distinct points so that you can impose two different boundary equations, which is not possible if the two points coincide. Another technique [6] is expressing the Kutta condition in term of velocity by decomposing it at the trailing edge and imposing equal components tangential to the wake, that is a sort of linearization of the problem.

In this work we impose the Kutta condition at the trailing edge, letting it be composed by the same geometrical point, with two distinct logical points. The condition is exactly that the pressure jump is null.

So in addition to the equations in (1.36) we add

$$p(TE^+) - p(TE^-) = 0 \text{ on } TE. \tag{1.37}$$

We know from the Bernoulli equation (1.7) that $\frac{p(TE^\pm)}{\rho} = -\frac{1}{2}|\nabla\Phi(TE^\pm)|^2 + C$ so we obtain

$$\begin{aligned}
p(TE^+) - p(TE^-) &= -\frac{1}{2}|\nabla\Phi(TE^+)|^2 + C + \frac{1}{2}|\nabla\Phi(TE^-)|^2 - C \\
&= -\frac{1}{2}|\nabla\Phi(TE^+)|^2 + \frac{1}{2}|\nabla\Phi(TE^-)|^2 \\
&= -\frac{1}{2}|\nabla(\phi(TE^+) + \phi_\infty)|^2 + \frac{1}{2}|\nabla(\phi(TE^+) + \phi_\infty)|^2 \\
&= -\frac{1}{2}|\nabla\phi(TE^+)|^2 - (\nabla\phi(TE^+), \nabla\phi_\infty) - \frac{1}{2}|\nabla\phi_\infty|^2 \\
&\quad +\frac{1}{2}|\nabla\phi(TE^-)|^2 + (\nabla\phi(TE^-), \nabla\phi_\infty) + \frac{1}{2}|\nabla\phi_\infty|^2 \\
&= -\frac{1}{2}(|\nabla\phi(TE^+)|^2 - |\nabla\phi(TE^-)|^2) \\
&\quad -(\nabla(\phi(TE^+) - \phi(TE^-)), \mathbf{V}_\infty).
\end{aligned} \tag{1.38}$$

So the Kutta condition in term of potential becomes

$$-\frac{1}{2}(|\nabla\phi(TE^+)|^2 - |\nabla\phi(TE^-)|^2) - (\nabla(\phi(TE^+) + \phi(TE^-)), \mathbf{V}_\infty) = 0. \tag{1.39}$$

This is a nonlinear condition and requires a suitable numerical treatment to solve the nonlinearity as it will be discussed in section 2.3.

We must point out that the condition of no pressure jump at the trailing edge is not always sufficient. As several numerical simulations at higher angles of attack suggested, we can incur a solution very similar to the one without the wake, as illustrated in Figure 1.7.

As suggested by Bernoulli equation (1.7) Kutta condition only prescribes that the velocity magnitudes on the top and bottom sides of trailing edge assume the same values. Yet when the top and bottom trailing edge velocities have the same norm, but different directions, the Kutta condition is still satisfied. In such cases no significant lift force is observed, and additional treatment is required to obtain a lifting solution. So, to reach higher angles of attack, we impose the additional condition that requires that the longitudinal components of the top and bottom trailing edge velocities are equal, namely

$$\nabla\Phi(TE^+)_x - \nabla\Phi(TE^-)_x = 0, \tag{1.40}$$

where x is the coordinate tangential to the wake in 2D, while in 3D it is the coordinate normal to the trailing edge and tangential to the wake. This condition is stronger than the one that considers only the pressure jump. In 2 dimensional problems, the

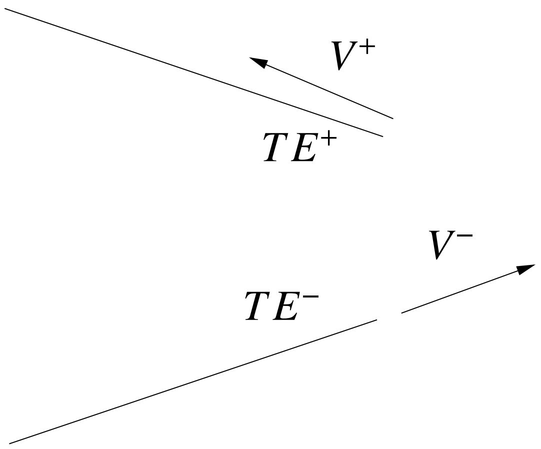


Figure 1.7: Trailing edge velocities for high angle of attack.

satisfaction of equation (1.40) implies that of equation (1.37), since, imposing one component and considering tangent velocities at the trailing edge, the other component is automatically imposed (Figure 1.8). For clarity, if we assign a value to V_x^+, $\mathbf{V}^+$ is automatically fixed since it is tangential to the body. Thus imposing the continuity of the longitudinal component V_x also results in that $V^+ = V^-$ and $V_y^+ = -V_y^-$ (see Figure 1.8).

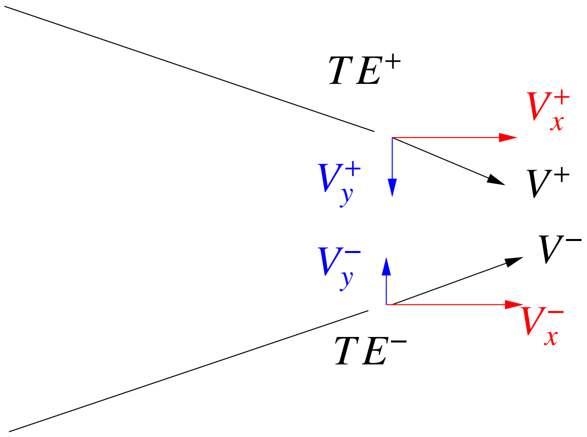


Figure 1.8: 2D trailing edge condition on the velocity's components.

The 3D procedure is the same and the components are shown in Figure 1.9. In this case, imposing equal components along the tangential coordinate to the wake has no specific implication on the other components.
We recall that the trailing edge of the body and of the wake are the same geometrical point, but three distinct logical points. In Figures 1.7,1.8,1.9 and 1.10, we splitted them in order to have a clearer vision.

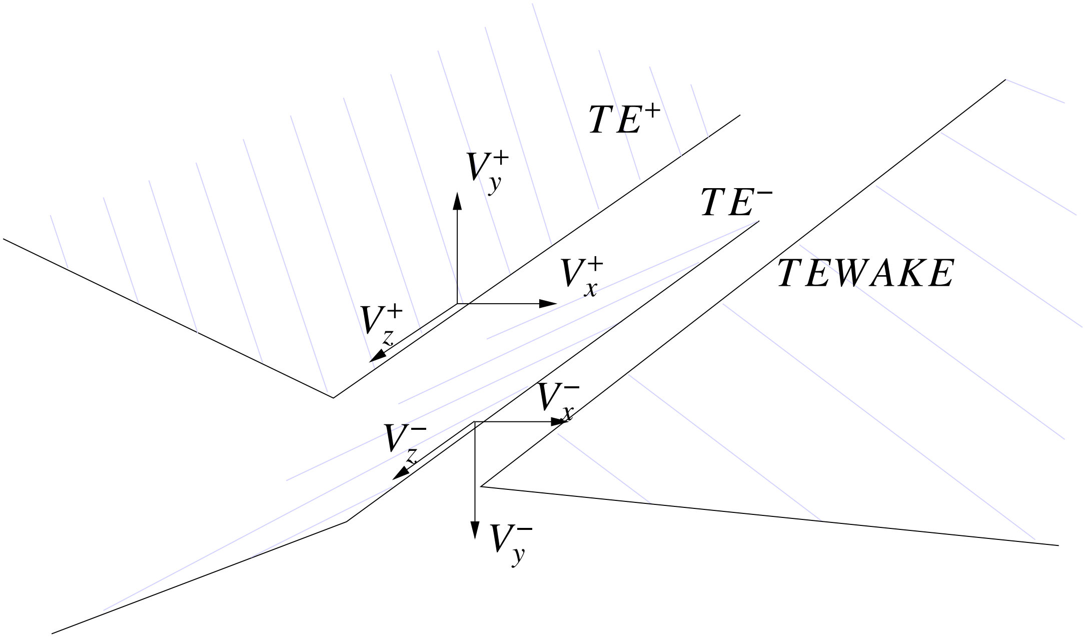


Figure 1.9: 3D trailing edge condition on the velocity's components.

1.5 Final mathematical model

We denote by u the term related to the potential

$$u := \begin{cases} \phi & \text{on } \Gamma_b \\ [\phi] & \text{on } \Gamma_w, \end{cases} \tag{1.41}$$

and with h the term related to its normal derivative

$$h := \begin{cases} \frac{\partial \phi}{\partial n} & \text{on } \Gamma_b \\ \left[\frac{\partial \phi}{\partial n}\right] & \text{on } \Gamma_w. \end{cases} \tag{1.42}$$

Thanks to the Neumann boundary conditions of non penetration on the body and no mass accumulation on the wake, we know the value of h on the boundaries

$$h := \begin{cases} -\mathbf{V}_\infty \cdot \mathbf{n} & \text{on } \Gamma_b \\ 0 & \text{on } \Gamma_w. \end{cases} \tag{1.43}$$

We can rewrite the single layer potential and the double layer potential terms as a function of h and u as follows

$$\begin{aligned}\int_{\Gamma} G\frac{\partial\phi}{\partial n}d\Gamma &= \int_{\Gamma_b} G\frac{\partial\phi}{\partial n}d\Gamma + \int_{\Gamma_{w^+}} G\frac{\partial\phi}{\partial n}d\Gamma + \int_{\Gamma_{w^-}} G\frac{\partial\phi}{\partial n}d\Gamma \\ &= \int_{\Gamma_b} G\frac{\partial\phi}{\partial n}d\Gamma + \int_{\Gamma_w} G\left[\frac{\partial\phi}{\partial n}\right]d\Gamma = \int_{\Gamma} Ghd\Gamma \qquad (1.44)\\ \int_{\Gamma}^{PV} \phi\frac{\partial G}{\partial n}d\Gamma &= \int_{\Gamma_b}^{PV} \phi\frac{\partial G}{\partial n}d\Gamma + \int_{\Gamma_{w^+}}^{PV} \phi\frac{\partial G}{\partial n}d\Gamma + \int_{\Gamma_{w^-}}^{PV} \phi\frac{\partial G}{\partial n}d\Gamma \\ &= \int_{\Gamma_b}^{PV} \phi\frac{\partial G}{\partial n}d\Gamma + \int_{\Gamma_w}^{PV} [\phi]\frac{\partial G}{\partial n}d\Gamma = \int_{\Gamma}^{PV} u\frac{\partial G}{\partial n}d\Gamma. \qquad (1.45)\end{aligned}$$

Now we can recast the original problem (1.36) in the following way:
Given h, find u that satisfies

$$\alpha(\mathbf{x})u(\mathbf{x}) + \int_{\Gamma}^{PV} u(\mathbf{y})\tfrac{\partial G(\mathbf{x}-\mathbf{y})}{\partial n}dy = \int_{\Gamma} G(\mathbf{x}-\mathbf{y})h(\mathbf{y})dy \text{ on } \Gamma_b \qquad (1.46)$$

$$u(\mathbf{x}) = u(TE^+) - u(TE^-) \text{ on } \Gamma_w \qquad (1.47)$$

$$u(\mathbf{x}) = 0 \text{ on } \partial\Gamma_w, \qquad (1.48)$$

where u is also subject to the constraint due to the Kutta condition

$$p(TE^+) - p(TE^-) = 0 \text{ on } TE^+, \qquad (1.49)$$

and in case of high angle of attack to the additional condition

$$\nabla\Phi(TE^+)_x - \nabla\Phi(TE^-)_x = 0 \text{ on } TE^+. \qquad (1.50)$$

If we denote by β the angle between the X axis and the trailing edge longitudinal direction x, as shown in Figure 1.10, Equation (1.50) can be rewritten as

$$[\nabla\Phi(TE^+) - \nabla\Phi(TE^-)]\cdot\begin{bmatrix}\cos(\beta)\\ \sin(-\beta)\\ 0\end{bmatrix} = 0 \text{ on } TE^+. \qquad (1.51)$$

The mathematical problem we have to solve is the following:
Given h, find u that satisfies

$$\begin{gathered}\alpha(\mathbf{x})u(\mathbf{x}) + \int_{\Gamma}^{PV} u(\mathbf{y})\tfrac{\partial G(\mathbf{x}-\mathbf{y})}{\partial n}dy = \int_{\Gamma} G(\mathbf{x}-\mathbf{y})h(\mathbf{y})dy \text{ on } \Gamma_b \qquad (1.52)\\ p(TE^+) - p(TE^-) = 0 \text{ with } x \in TE\\ u(\mathbf{x}) = u(TE^+) - u(TE^-) \text{ on } \Gamma_w\\ u(\mathbf{x}) = 0 \text{ on } \partial\Gamma_w.\end{gathered}$$

There is however another unknown, that is the position of the wake, because the condition on the potential jump on it is valid only if the wake is a flow surface. In order to find its position, we have to compute the velocity in each point of the domain, so that we will be able to align the wake with the flow. We call this procedure wake relaxation and we describe it in detail in section 2.4.

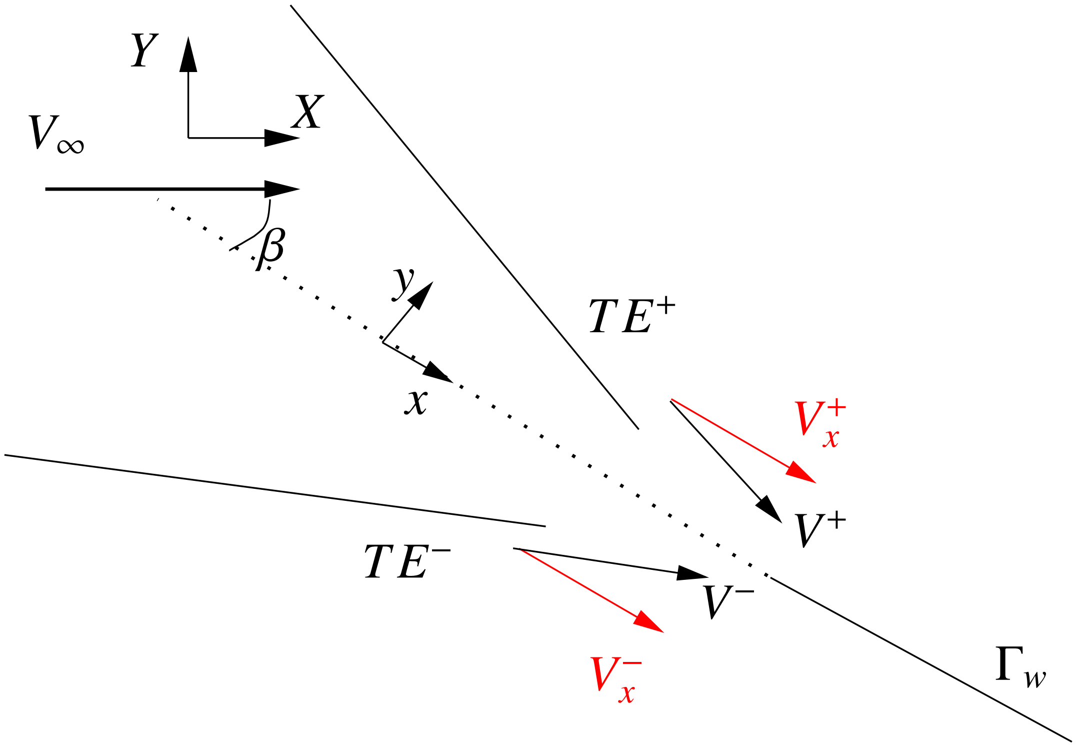


Figure 1.10: Trailing edge condition on the components: the velocities V^+ and V^- calculated in the general coordinates system $X\ Y$ are projected in the local one $x\ y$ in order to match the x components V_x^+ and V_x^-.

1.6 Postprocessing

Once we have computed the solution of the problem (1.53) in terms of the potential, we can compute the velocities on the body, using a gradient recovery strategy on the surface. The velocities field in Ω is calculated taking the gradient of the boundary integral equation (1.20)

$$\begin{aligned} \mathbf{V}(\mathbf{x}) &= \nabla_{\mathbf{x}}\Phi(\mathbf{x}) = \nabla_{\mathbf{x}}\phi(\mathbf{x}) + \mathbf{V}_\infty \\ &= \int_{\Gamma_b} \frac{\partial\phi(\mathbf{y})}{\partial n}\nabla_{\mathbf{x}}G(\mathbf{x}-\mathbf{y})dy - \int_{\Gamma_b}\phi(\mathbf{y})\nabla_{\mathbf{x}}\frac{\partial G(\mathbf{x}-\mathbf{y})}{\partial n}dy \\ &\quad - \int_{\Gamma_w}[\phi(\mathbf{y})]\,\nabla_{\mathbf{x}}\frac{\partial G(\mathbf{x}-\mathbf{y})}{\partial n}dy + \mathbf{V}_\infty, \end{aligned} \tag{1.53}$$

where $\nabla_{\mathbf{x}}$ is the gradient operator applied to a function with respect to the variable $\mathbf{x}$. This formula is not well defined in the presence of edges since a hypersingular term appears in $\nabla_{\mathbf{x}}\frac{\partial G(\mathbf{x}-\mathbf{y})}{\partial n}$ [30].

In the 2D case, $\nabla_{\mathbf{x}} \frac{\partial G(\mathbf{x}-\mathbf{y})}{\partial n}$ is given by

$$\begin{aligned} \nabla_{\mathbf{x}} \frac{\partial G(\mathbf{x}-\mathbf{y})}{\partial n} &= \nabla_{\mathbf{x}} \frac{\partial G(\mathbf{r})}{\partial n} \\ &= -\frac{1}{2\pi|\mathbf{r}|^4} \begin{pmatrix} -r_1^2 + r_2^2 & -2r_1r_2 \\ -2r_2r_1 & -r_2^2 + r_1^2 \end{pmatrix} \mathbf{n}, \end{aligned} \tag{1.54}$$

while in the 3D case, we have

$$\begin{aligned} \nabla_{\mathbf{x}} \frac{\partial G(\mathbf{x}-\mathbf{y})}{\partial n} &= \nabla_{\mathbf{x}} \frac{\partial G(\mathbf{r})}{\partial n} \\ &= -\frac{1}{4\pi|\mathbf{r}|^5} \begin{pmatrix} 2r_1^2 - r_2^2 - r_3^2 & 3r_1r_2 & 3r_1r_3 \\ 3r_2r_1 & 2r_2^2 - r_1^2 - r_3^2 & 3r_2r_3 \\ 3r_3r_1 & 3r_3r_2 & 2r_3^2 - r_1^2 - r_2^2 \end{pmatrix} \mathbf{n}. \end{aligned} \tag{1.55}$$

The presence of the hypersingular term in the integral for the computation of the velocity gives problems in the numerical quadrature. So in order to obtain the velocity on the body and on the wake we will show how to avoid the numerical integration of this term in section 2.5.
We can also obtain the pressure using Bernoulli equation (1.7) and the pressure coefficients using the gradients of the potential calculated on the body

$$c_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty V_\infty^2} = 1 - \frac{|\nabla\Phi|^2}{V_\infty^2}. \tag{1.56}$$

Another coefficient we can calculate is the lift coefficient. The lift L is defined as the component of the force normal to the flow direction, while the lift coefficient C_L is the adimensional quantity.

$$C_L = \frac{L}{\frac{1}{2}\rho_\infty V_\infty^2 S}. \tag{1.57}$$

In 2D simulations, the lift coefficient is the area of the surface between the curve of the pressure coefficient, so it is the integral of the pressure coefficient.

Chapter 2

Numerical method

In this chapter we introduce a discretization of the boundary integral problem (1.46), (1.47) and the Kutta condition (1.49) introduced in the previous chapter. We show how this nonlinear problem can be solved iteratively by computing a sequence of linear problems until convergence.

2.1 Algebraic formulation

We introduce a suitable discretization of the boundary, then we define standard Lagrangian finite elements on Γ as basis functions for the unknown ϕ. The unknown of the problem are the values of ϕ on the respective set of interpolatory points of the Finite Element discretization. For a detailed treatment see [2, 3, 1].
We define a computational mesh defined as a regular decomposition Γ_h of the boundary Γ. Such partition is composed of linear segments (for 2D problems) or quadrilateral cells (for 3D problems). Each element is denoted by K_l and the discrete boundary Γ_h is given by $\Gamma_h = \bigcup K_l$. Regular means that any two cells K_l, K_m only intersects on common faces, edges or vertices, and that there exists a mapping from a reference cell $\hat{K}$ such that the determinant of the Jacobian of this mapping is uniformly bounded away from zero for all cells K_l. In all the computations carried out in this work we provide an extremely coarse grid and then we progressively refine it to obtain a solution with specified accuracy. From the user's perspective, this eases the mesh generation problem, provided that the solver is aware of the underlying geometry on which the mesh should be refined. In the next chapter we will present the procedure that allows the generation of the computational mesh and its refinement on the desired wing geometry.
To put the boundary integral equation into a numerical framework approximating the problem (1.46), we need to find suitable numerical approximation of both the BIE integrals and the unknowns ϕ and $[\phi]$. To start with the unknown functions, let us define the space V_h on Γ_h, such that

$$V_h := \{\phi_h \in C^0(\Gamma_h) : \phi_h|_{K_l} \in \mathcal{Q}^r(K_l), K_l \in \Gamma_h\}, \tag{2.1}$$

where $C^0(\Gamma_h)$ is the space of continuous function over Γ_h, $\mathcal{Q}^r$ is the polynomial space of degree r defined on the cell K_l.

While the Galerkin method appears as the natural choice for FEMs, in the BEM framework such discretization strategy results in a second integration of weakly singular kernels. We know how to treat the singular integrals that naturally appear in the BIE using specific quadrature formulas. Yet, a second integration would increase the computational complexity of the BEM. Therefore we consider a common approach for the solution of boundary integral equations in the engineering community, which is represented by the collocation boundary element method [40]. The collocation method consists in replacing the continuous function ϕ with its numerical approximation $\phi_h \in V_h$ which represents the discretized potential in the finite dimensional space, and collocating the BIE on a number of points equal to the number of unknowns. When linear finite elements are selected, such collocation points coincide with the nodes of the numerical quadrilateral grid (Figure 2.1). Another possible choice would be to collocate the equations close to, but not on the boundary, in such a way the integrals appearing in equation (1.24) would not be singular at all. However, if the domain presents sharp corners or cusps, the point selection may not be straight-forward. In this work, we collocate these points on the boundary Γ_h and on the trailing edge in order to deal with the Kutta condition exactly. In our method there will be the possibility of having two logical distinct points but geometrically identical at the trailing edge, so that we will be able to impose two conditions, namely the boundary integral equation and the Kutta condition.

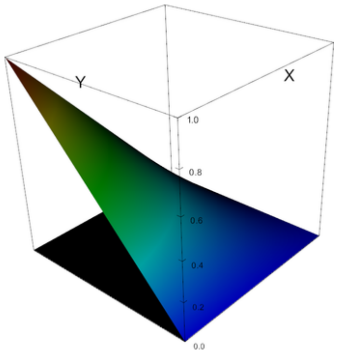


Figure 2.1: Linear finite element on quadrilateral cell.

Let $\psi_j(\mathbf{x})$ be the basis functions of V_h such that $V_h = \mathrm{span}\{\psi_j\}_{j=1}^N$, where N denotes the dimension of the finite dimensional space, such that $N = N_b + N_w$ where the numbers of degrees of freedom are N_b on the body and N_w on the wake respectively, then for any $\phi_h \in V_h$ we have

$$\phi_h(\mathbf{x}) = \sum_{j=1}^{N} \phi_j \psi_j(\mathbf{x}), \tag{2.2}$$

so that ϕ_h is uniquely identified by the vector $\phi = \{\phi_j\}$. In this framework, the degrees of freedom are the values of the potential at the collocation points $\mathbf{x}_i$ with $i = 1, ..., N_b$, on the body, and the boundary integral equation becomes

$$\begin{aligned} \alpha(\mathbf{x}_i)\phi_h(\mathbf{x}_i) + \int_{\Gamma_b} \phi_h(\mathbf{y})\tfrac{\partial G(\mathbf{x}_i-\mathbf{y})}{\partial n}dy + \int_{\Gamma_w} \phi_h(\mathbf{y})\tfrac{\partial G(\mathbf{x}_i-\mathbf{y})}{\partial n}dy = \\ \int_{\Gamma_b} G(\mathbf{x}_i - \mathbf{y})\tfrac{\partial \phi_h(\mathbf{y})}{\partial n}dy \text{ if } \mathbf{x}_i \in \Gamma_b, \end{aligned} \tag{2.3}$$

where, on Γ_w, ϕ_h is the potential jump across the wake and the contribution in the right-hand side of the jump of $\frac{\phi_h(\mathbf{y})}{\partial n}$ is null for equation (1.31).
In the discrete form, equation (2.3) becomes

$$\begin{aligned} \alpha(\mathbf{x}_i) \textstyle\sum_{j=1}^{N} \phi_j\psi_j(\mathbf{x}_i) + \sum_{l=1}^{K_b} \sum_{j=1}^{N} \int_{K_l} \phi_j\psi_j(\mathbf{y})\tfrac{\partial G(\mathbf{x}_i-\mathbf{y})}{\partial n}dS \\ + \textstyle\sum_{l=1}^{K_w} \sum_{j=1}^{N} \int_{K_l} \phi_j\psi_j(\mathbf{y})\tfrac{\partial G(\mathbf{x}_i-\mathbf{y})}{\partial n}dS \ = \\ \textstyle\sum_{l=1}^{K_b} \sum_{j=1}^{N} \int_{K_l} G(\mathbf{x}_i - \mathbf{y})\phi_j\tfrac{\partial\psi_j(\mathbf{y})}{\partial n}dS \ \text{ if } \mathbf{x}_i \in \Gamma_b, \end{aligned} \tag{2.4}$$

where $\phi_h(\mathbf{x}_i)$ in (2.3) is replaced with its discrete counterpart $\sum_{j=1}^{N} \phi_j\psi_j(\mathbf{x}_i)$ in (2.4) according to (2.2) with N as the number of degrees of freedom. The integral on Γ_b in (2.3) is replaced with the summation of the K_b contributions obtained from the integrals on the body cells. The same argument explained for the integrals on Γ_b holds for the discretization of the integral on Γ_w. The integral of a function $f(\mathbf{y})$ over each cell is performed in the numerical framework according to

$$\int_{K_l} f(\mathbf{y})dS = \sum_{q=1}^{N_q} f(\mathbf{x}_q)J_l^q. \tag{2.5}$$

The integration over each body cell is carried out through the sum of the discrete values on the N_q quadrature nodes of the integral argument $f(\mathbf{x}_q)$. Each element of such sum is multiplied by J_q^l. It represents the value of the determinant of the local Jacobian of the mapping from the reference cell $\hat{K}$ to the cell $K_l \in \Gamma_h$ at each quadrature node $\mathbf{x}_q$ and its corresponding quadrature weight. For the complete discussion of the mapping, its Jacobian and the quadrature weights, we refer to [3]. The integrals on the reference cell, that in discrete form is the summation on N_q, are performed using different quadrature schemes. When the collocation point does not lie inside the cell we simply use bidimensional Gauss scheme, if the collocation point is inside the current cell we need a different strategy since the kernels G and $\frac{\partial G}{\partial n}$ are weakly singular, in particular we use bidimensional Telles quadrature formula [26].
We can now write the linear system of equations

$$(\mathbf{A} + \mathbf{N})\phi = \mathbf{b}, \tag{2.6}$$

where

$$\mathbf{A}_{ij} = \alpha(\mathbf{x}_i)\psi_j(\mathbf{x}_i) \tag{2.7}$$

$$\mathbf{N}_{ij} = \sum_{l=1}^{K_b}\sum_{q=1}^{N_q} \psi_j(\mathbf{x}_q)\frac{\partial G(\mathbf{x}_i - \mathbf{x}_q)}{\partial n}J_q^l + \sum_{l=1}^{K_w}\sum_{q=1}^{N_q} \psi_j(\mathbf{x}_q)\frac{\partial G(\mathbf{x}_i - \mathbf{x}_q)}{\partial n}J_q^l \tag{2.8}$$

$$\mathbf{b}_i = \sum_{l=1}^{K_b}\sum_{q=1}^{N_q} G(\mathbf{x}_i - \mathbf{x}_q)\frac{\partial \psi_j(\mathbf{x}_q)}{\partial n}J_q^l. \tag{2.9}$$

Matrices $\mathbf{A}$ and $\mathbf{N}$ belong to $\mathbb{R}^{N_b \times N}$ and vector $\mathbf{b} \in \mathbb{R}^{N_b}$. In particular, choosing the collocation points $\mathbf{x}_i$ as the support points of the nodal basis function $\psi_i(\mathbf{x})$, we obtain $\psi_j(\mathbf{x}_i) = \delta_{ij}$, where δ_{ij} is the delta of Kronecker. This choice simplifies the structure of the matrix $\mathbf{A}$, in fact it becomes diagonal.

$$\begin{aligned} \mathbf{A}_{ii} &= \alpha(\mathbf{x}_i) \\ \mathbf{A}_{ij} &= 0 \text{ with } i \neq j. \end{aligned} \tag{2.10}$$

We also know, see [8], that for the internal problem

$$\alpha(\mathbf{x}_i) = -\int_{\Gamma_b} \frac{\partial G(\mathbf{x}_i - \mathbf{y})}{\partial n} dy, \tag{2.11}$$

thus for external problem we obtain

$$\alpha(\mathbf{x}_i) = 1 - \int_{\Gamma_b} \frac{\partial G(\mathbf{x}_i - \mathbf{y})}{\partial n} dy = 1 - \sum_{l=1}^{K_b}\sum_{q=1}^{N_q} \frac{\partial G(\mathbf{x}_i - \mathbf{x}_q)}{\partial n}J_q^l. \tag{2.12}$$

Considering that $\sum_{j=1}^{N} \psi_j(\mathbf{y}) = 1$ for Lagrangian elements, matrix $\mathbf{A}$ can be computed summing up the columns of matrix $\mathbf{N}$

$$\mathbf{A}_{ii} = 1 - \sum_j \mathbf{N}_{ij}. \tag{2.13}$$

Equation (2.6) is satisfied on the body, that is for the ϕ_i corresponding to $\mathbf{x}_i \in \Gamma_b$.
The treatment of external flow for non lifting bodies was already implemented in π-`BEM`. We now discuss the imposition of the nonlinear Kutta condition on the nodes at the trailing edge, and describe both the resultings nonlinear problem and the Newton method adopted for its resolution. All such aspects of implementation are fundamental for the potential flow simulations past lifting bodies and were implemented in the framework of the present master thesis work.
On the wake we have to impose other conditions for the degrees of freedom associated to the collocation points $\mathbf{x}_i$ with $i = 1...N_w$. We assume now that the wake is a flow surface, then in section 2.4 we will show which is the procedure in order to ensure

this assumption. On Γ_w we have to impose the analogous of condition (1.47) in the algebraic form

$$[\phi]_i = \phi(TE_i^+) - \phi(TE_i^-) \text{ with } i : \mathbf{x}_i \in \Gamma_w, \tag{2.14}$$

or equivalently

$$[\phi]_i - \phi(TE_i^+) + \phi(TE_i^-) = 0 \text{ with } i : \mathbf{x}_i \in \Gamma_w, \tag{2.15}$$

where the notation TE_i refers to the corresponding node on the trailing edge of the node with index i, id est $\mathbf{x}_i$ belongs to the streamline leaving TE_i.

On the boundary of the wake $\partial\Gamma_w$ excluding the trailing edge we have the same of equation (1.48) in algebraic form

$$[\phi]_i = 0 \text{ with } i : \mathbf{x}_i \in \partial\Gamma_w. \tag{2.16}$$

These conditions (2.15) and (2.16) on the wake in discrete form generate a matrix $\mathbf{C} \in \mathbb{R}^{N_w \times N}$ with $\mathbf{C}_{ii} = 1$ and for the $\mathbf{x}_i \in \Gamma_w$ we have $\mathbf{C}_{ij} = -1$ for $j = TE_i^+$ and $\mathbf{C}_{ij} = 1$ for $j = TE_i^-$ and the other entries null, while for the $\mathbf{x}_i \in \partial\Gamma_w$ all the other entries are null. The right-hand side related to the wake conditions has null entries. We can generate the full matrix describing the system of equations as:

$$\mathbf{Z} = \begin{bmatrix} \mathbf{A} + \mathbf{N} \\ \mathbf{C} \end{bmatrix}$$

and the full right-hand side as:

$$\mathbf{d} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}$$

where $\mathbf{Z} \in \mathbb{R}^{N \times N}$ and $\mathbf{d} \in \mathbb{R}^N$. In order to solve this linear system of equations, since the matrix $\mathbf{A} + \mathbf{N}$ and so matrix $\mathbf{Z}$ is non symmetric and dense, we use an iterative GMRES Krylov solver. We here remark that such iterative solver is also used to solve all the other linear systems involved in our numerical methodology. In particular, the linear systems resulting from the gradient recovery strategy (equation (2.23)) and the linear system needed to invert the Jacobian in the Newton iterations (equation (2.42)). Both parts of the overall numerical model will be illustrated in the next sections.

On the lower trailing edge we have to impose the boundary integral equation. In this case, since at the trailing edge we have two geometrically identical, but logically distinct, points in writing such boundary integral equation we have to consider an angle α for both the ϕ at upper and lower trailing edge sides as shown in Figure 2.2.

So equation (1.46) becomes

$$\alpha(TE^+)\phi(TE^+) + \alpha(TE^-)\phi(TE^-) + \int_{\Gamma_b} \phi(\mathbf{y})\frac{\partial G(\mathbf{x}-\mathbf{y})}{\partial n}dy + \int_{\Gamma_w} [\phi]\,(\mathbf{y})\frac{\partial G(\mathbf{x}-\mathbf{y})}{\partial n}dy = \int_{\Gamma_b} G(\mathbf{x}-\mathbf{y})\frac{\partial\phi(\mathbf{y})}{\partial n}dy \text{ with } x \in TE^-. \tag{2.17}$$

On the trailing edge we have to impose the Kutta condition on the upper side, instead

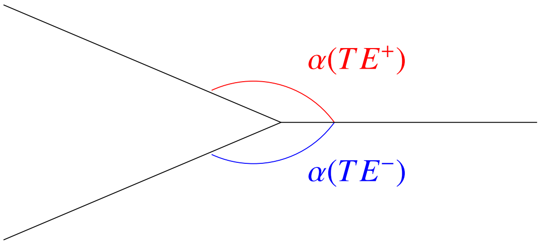


Figure 2.2: BIE at the trailing edge considering the right angle.

on the lower side we can use equation (2.6) with the correction discussed in equation (2.17), so it is enough to modify the rows of the linear system. In particular the angle α calculated on the trailing edge using equation (2.13) is the total angle seen by the trailing edge, so we have to divide by two and put the result in $\mathbf{A}_{ii}$ and in $\mathbf{A}_{ij}$ with j the corresponding index of the upper trailing edge associated with the lower trailing edge of index i.
The system of equations of the discretized problem is linear, but when introducing the Kutta condition nonlinear equations will be added.

2.2 Kutta condition

We cannot write equation (1.39) in the algebraic form, since the gradients of the basis functions ψ_j are not defined in the collocation points $\mathbf{x}_i$ but only inside a single cell of the mesh, in fact across the cells the gradients are discontinuous. So we have to recover the value of the ϕ gradient on the nodes and then calculate the value of equation (1.39). We want to find a function $\nabla\Phi$ that is the L_2-projection of the ϕ gradient we have on the cells $\nabla\Phi_c$, so for each component $k = 1, \ldots, \mathrm{d}$ we solve

$$(\nabla\Phi[k], v) = (\nabla\Phi_c[k], v) \quad \forall v \in V_h. \tag{2.18}$$

Using the basis functions, we can write $\nabla\Phi(\mathbf{x})[k] = c_i[k]\psi_i(\mathbf{x})$ and we choose as test functions again the basis functions ψ_j so that we obtain

$$c_i[k](\psi_i, \psi_j) = (\nabla\Phi_c[k], \psi_j) \quad \forall j, \tag{2.19}$$

where the gradient of the total potential can be split in the two contributions deriving from the gradient of the perturbation potential and the inflow velocity of the fluid.

$$\nabla\Phi_c[k] = \nabla\phi_c[k] + \mathbf{V}_\infty[k]. \tag{2.20}$$

Moreover the gradient of the potential is the sum of two contributions, one normal to the cell, the other tangential

$$\nabla\phi_c[k] = \nabla_S\phi_c[k] + \nabla\phi_c[k] \cdot \mathbf{n}, \tag{2.21}$$

where ∇_S is the superficial gradient operator.
Using again the basis functions we can write the gradient of the potential on a cell

$$\nabla\Phi_c[k] = \sum_{i=1}^{N}(\phi_i \nabla_S \psi_i[k] + \frac{\partial \phi_i}{\partial n}[k]\psi_i) + \mathbf{V}_\infty[k]. \tag{2.22}$$

In algebraic form equation (2.18) becomes

$$\mathbf{M}\mathbf{c}[k] = \mathbf{r}[k], \tag{2.23}$$

where $\mathbf{c}[k]$ is the vector with components $c_i[k]$ and $\mathbf{M}$ is a mass matrix

$$\mathbf{M}_{ij} = \int_\Gamma \psi_i \psi_j dS = \sum_{l=1}^{K}\sum_{q=1}^{N_q} \psi_i(\mathbf{x}_q)\psi_j(\mathbf{x}_q)J_q^l. \tag{2.24}$$

In particular using Lagrangian elements, we do not have to integrate all over the domain Γ, but it is sufficient to integrate over the cells having that share the degree of freedom with index i, since otherwise the product of the basis functions is null.
The right-hand side is given by

$$\begin{aligned} \mathbf{r}_i[k] &= \int_\Gamma \left(\textstyle\sum_{i=1}^{N}(\phi_i \nabla_S \psi_i[k] + \frac{\partial \phi_i}{\partial n}[k]\psi_i) + \mathbf{V}_\infty[k]\right)\psi_j dS \\ &= \textstyle\sum_{l=1}^{K}\sum_{q=1}^{N_q}\left(\sum_{i=1}^{N}\left(\phi_i \nabla_s \psi_i[k](\mathbf{x}_q) + \frac{\partial \phi_i}{\partial n}[k]\psi_i(\mathbf{x}_q)\right) + \mathbf{V}_\infty[k]\right)\psi_j(\mathbf{x}_q)J_q^l. \end{aligned} \tag{2.25}$$

Solving the linear system (2.23) we obtain the coefficients of the L_2-projection of the gradient.
Once we have recovered each component of the gradient, we can construct the full gradient, hence we can calculate the pressure on each node.

$$p = \frac{1}{2}|\nabla\Phi|^2 = \frac{1}{2}\left(\sum_{k=1}^{\mathrm{d}} \nabla\Phi[k]^2\right). \tag{2.26}$$

In particular we can use the procedure described to evaluate the pressure on the upper and lower sides of the trailing edge, once the potential vector $\phi = \{\phi_j\}$ is assigned. But in order to determine the correct potential ϕ, we need to impose the Kutta condition and find the potential resulting in a null pressure jump at the trailing edge. As Kutta condition is quadratic, the result in nonlinear problem is solved by means of Newton iteration scheme.

2.3 Newton method for the nonlinear algebraic problem

We want to impose the condition of null pressure jump at the trailing edge. We have seen in section 2.2 how to evaluate the pressure values on the trailing edge, once $\phi =$

$\{\phi_j\}$ is assigned. The correct ϕ value is then the one satisfying the nonlinear system $\mathbf{f}(\phi) = \mathbf{0}$. To find such value, we resort to Newton iterations.
Given an initial guess ϕ_0, we seek for the increment $\Delta\phi$ such that $\mathbf{f}(\phi_0 + \Delta\phi) = \mathbf{0}$. If we use a first order approximation of this equation, we obtain $\mathbf{f}(\phi_0 + \Delta\phi) = \mathbf{f}(\phi_0) + \mathbf{f}'(\phi_0)\Delta\phi = \mathbf{0}$, that leeds to solve this iterative system of equations

$$\mathbf{f}'(\phi_n)\Delta\phi = -\mathbf{f}(\phi_n) \tag{2.27}$$

$$\phi_{n+1} = \phi_n + \Delta\phi. \tag{2.28}$$

We first find the increment $\Delta\phi$ then we update the value of ϕ_{n+1} and continue this procedure until $\mathbf{f}(\phi_{n+1})$ is small, in particular until the L^2-norm of $\mathbf{f}(\phi_{n+1})$ is less than a certain threshold.
In order to solve iteratively the Newton scheme, we use the package `NOX` (Nonlinear Object-Oriented Solutions) of the library `Trilinos` [41], that is designed to solve large-scale systems of nonlinear equations. This package needs two main functions: the first is the evaluation of $\mathbf{f}(\phi)$, the second is the result of the product between $\mathbf{f}'(\phi)$ and the increment $\Delta\phi$ given in input.
We have to define the residual $\mathbf{f}(\phi)$ for the problem as follows

$$\mathbf{f}(\phi) = \begin{cases} (\mathbf{A} + \mathbf{N})\phi - \mathbf{b} & \text{on } \Gamma_b \text{ and } TE^- \\ [\phi] - \phi(TE^+) + \phi(TE^-) & \text{on } \Gamma_w \\ [\phi] & \text{on } \partial\Gamma_w \\ p(TE^+) - p(TE^-) & \text{on } TE^+. \end{cases} \tag{2.29}$$

In order to solve the problem also with high angle of attack, on the upper trailing edge we can also impose

$$\nabla\Phi(TE^+)_x - \nabla\Phi(TE^-)_x = 0 \quad \text{on } TE^+. \tag{2.30}$$

The Jacobian matrix is given by $\mathbf{J} = \mathbf{f}' = \frac{\partial \mathbf{f}}{\partial \phi}$. We have to take the residual and differentiate it by the variable ϕ. On the body and the lower trailing edge the function $\mathbf{f}$ is a linear function of ϕ, so the Jacobian is simply

$$\mathbf{J} = (\mathbf{A} + \mathbf{N}) \quad \text{on } \Gamma_b \text{ and } TE^-, \tag{2.31}$$

on the wake we impose the same constraint as in the residual

$$\mathbf{J}_{ii} = 1 \quad \mathbf{J}_{iTE^+} = -1 \quad \mathbf{J}_{iTE^-} = 1 \text{ for nodes } i \text{ on } \Gamma_w, \tag{2.32}$$

on the lower trailing edge we have to impose

$$\mathbf{J} = \frac{\partial p(TE^+)}{\partial \phi} - \frac{\partial p(TE^-)}{\partial \phi}, \tag{2.33}$$

or alternatively

$$\mathbf{J} = \frac{\partial \nabla\Phi(TE^+)_x}{\partial \phi} - \frac{\partial \nabla\Phi(TE^-)_x}{\partial \phi}. \tag{2.34}$$

We want to know

$$\frac{\partial p(TE)}{\partial \phi} = \frac{\partial \frac{1}{2}\left(\sum_{k=1}^{d} \nabla\Phi[k]^2\right)}{\partial \phi} = \frac{1}{2}\sum_{k=1}^{d} \frac{\partial \nabla\Phi[k]^2}{\partial \phi} = \sum_{k=1}^{d} \nabla\Phi[k]\frac{\partial \nabla\Phi[k]}{\partial \phi}. \tag{2.35}$$

The values of $\nabla\Phi[k]$ are obtained by the gradient recovery procedure introduced in section 2.2, while we do not have the values of $\frac{\partial \nabla\Phi}{\partial \phi}$, because we have only numerical values of the gradient at the trailing edge. We need to exploit a method that can bring back the derivatives of the gradients. Using the package `Sacado` of the library `Trilinos`, that is a set of automatic differentiation tools, we can build the residual in (2.23) bringing back all the derivatives of the gradient's components. In particular, while we compute the right-hand side as shown in equation (2.25), we differentiate this formula by the variable ϕ. Differentiating equation (2.23) on both sides, we obtain for each component k

$$\mathbf{M}\mathbf{J}_{\nabla\Phi}[k] = \mathbf{J}_{rhs}[k], \tag{2.36}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{J}_{\nabla\phi}$ is the Jacobian of the gradient that we want to know and $\mathbf{J}_{rhs}$ is the Jacobian of the right-hand side of equation (2.25) that we have built by differentiation. The most important thing is that it is not necessary to solve this equation for the matrix $\mathbf{J}_{\nabla\Phi}$, because we do not need to know it. Our goal is to calculate the product between the Jacobian matrix $\mathbf{J}$ and the increment $\Delta\phi$. So multiplying equation (2.36) by the increment $\Delta\phi$, we obtain

$$\mathbf{M}\mathbf{J}_{\nabla\Phi}[k]\Delta\phi = \mathbf{J}_{rhs}[k]\Delta\phi. \tag{2.37}$$

The term $\mathbf{J}_{\nabla\Phi}[k]\Delta\phi$ is what effectively we need to put in the function of `NOX`. So this equation leeds to

$$\mathbf{J}_{\nabla\Phi}[k]\Delta\phi = \mathbf{M}^{-1}\mathbf{J}_{rhs}[k]\Delta\phi. \tag{2.38}$$

In order to solve this system, the `NOX` routine needs the scalar product function between a given increment $\Delta\phi$ and the matrix $\mathbf{M}^{-1}\mathbf{J}_{rhs}[k]$. So, in our implementation of such function, we first multiply the increment $\Delta\phi$ by the Jacobian matrix $\mathbf{J}_{rhs}[k]$ to obtain $\mathbf{t} = \mathbf{J}_{rhs}[k]\Delta\phi$. Then to evaluate the action of $\mathbf{M}^{-1}$ on $\mathbf{t}$ we solve the linear system $\mathbf{M}\mathbf{y} = \mathbf{t}$, the result of which is the jacobian matrix vector product required by `NOX` program.

In particular if we consider the imposition of the Kutta condition expressed in equation (2.33), the Jacobian is

$$\mathbf{s} = \sum_{k=1}^{\mathrm{d}} diag\left(\nabla\phi[k]\right)\mathbf{M}^{-1}\mathbf{J}_{rhs}[k]\Delta\phi, \tag{2.39}$$

where $diag\left(\nabla\phi[k]\right)$ is the diagonal matrix such that $diag\left(\nabla\phi[k]\right)_{ii} = \nabla\phi[k]_i$.

Otherwise if we impose equation (2.34) the Jacobian is

$$\mathbf{s} = \mathbf{M}^{-1}\mathbf{J}_{rhs}[1]\Delta\phi\cos(\beta) + \mathbf{M}^{-1}\mathbf{J}_{rhs}[2]\Delta\phi\sin(-\beta). \tag{2.40}$$

So we can summarize that the product between the Jacobian matrix and the increment is

$$
\mathbf{J}\Delta\phi = \begin{cases} (\mathbf{A}+\mathbf{N})\Delta\phi & \text{on } \Gamma_b \text{ and } TE^- \\ \Delta\phi_i - \Delta\phi_{TE^+} + \Delta\phi_{TE^-} & \text{on } \Gamma_w \\ \Delta\phi & \text{on } \partial\Gamma_w \\ \mathbf{s}_{TE^+} - \mathbf{s}_{TE^-} & \text{on } TE^+, \end{cases} \tag{2.41}
$$

where $\mathbf{s}_{TE^+}$ and $\mathbf{s}_{TE^-}$ are the components of the $\mathbf{s}$ vector at the indices referring the degrees of freedom on the upper and lower trailing edges respectively.
`NOX` will solve iteratively these equations

$$
\mathbf{J}(\phi_n)\Delta\phi = -\mathbf{f}(\phi_n) \tag{2.42}
$$

$$
\phi_{n+1} = \phi_n + \Delta\phi. \tag{2.43}
$$

In particular also the internal equation (2.42) is solved iteratively with a GMRES Krylov solver by this package in order to obtain the increment, in fact the program needs the effect of the Jacobian matrix on a vector, not its inverse.
The routine of the algorithm combined with `NOX` tools is presented in the pseudocode Algorithm 1.

Algorithm 1 Newton's algorithm

```
Newton loop:
    Assemble mass matrix $\mathbf{M}$
    Assemble Jacobian matrices $\mathbf{J}_{rhs}[k]$
    Recover the gradient and the pressure $\mathbf{Mc}[k] = \mathbf{r}[k]$
    Compute residual $\mathbf{f}(\phi)$
    if residual < threshold then
        Exit Newton loop
    else
        Jacobian loop $\mathbf{J}\Delta\phi = -\mathbf{f}(\phi)$
    Update potential value $\phi_{n+1} = \phi_n + \Delta\phi$
```

2.4 Wake relaxation routine

In section 1.4 we pointed out that the position of the wake is an unknown of the problem. At the stage of the solution of the nonlinear problem in section 2.3, we only impose the conditions (1.35) and (1.31). There is indeed no guarantee that the shape guessed for the wake results in a streamline or flow surface. Thus, we iteratively adjust the wake position and repeat the solution of the nonlinear system until the wake position is stable.

The first problem is how to move the wake to guarantee that it is aligned to the streamlines detaching from the trailing edge. To impose this, we move the nodes of the mesh according to the following iterative process

$$\mathbf{x}_{i+1}^{new} = \mathbf{x}_i + \nabla\phi(\mathbf{x}_i) + \mathbf{V}_\infty, \tag{2.44}$$

that means that the new position of the node downwind $\mathbf{x}_{i+1}$ is the one of the previous node translated by the velocity of the flow (Figure 2.3).

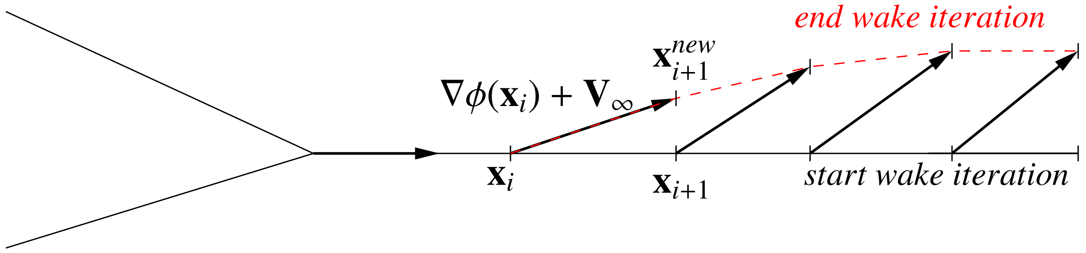


Figure 2.3: Wake relaxation routine in 2D.

In this way we can ensure that, after some time steps, the nodes of the wake leaving the trailing edge will lay on a streamline, so that condition (2.14) is satisfied. This routine continues until all the wake is aligned with its local velocity, so when the L^2-norm of the difference between the nodes' position at consecutive time steps is less than a certain threshold.

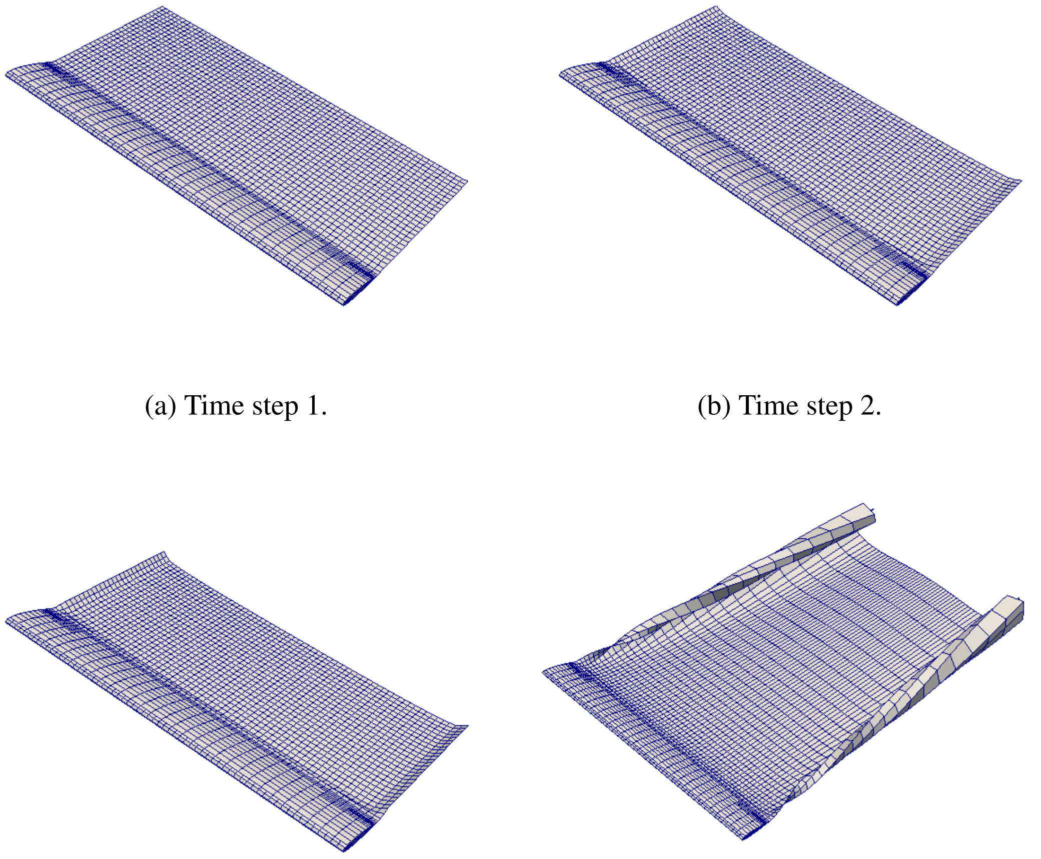
We cannot calculate the value of the potential gradient on the wake using the gradient recovery strategy, since we do not know the value of the potential and its normal derivative. In fact the solution of the BEM problem only results in the potential jump $[\phi]$ on the wake. So we need to exploit equation (1.53). We recall that this formula has no meaning in presence of edges, so we cannot use it to calculate the velocity on the trailing edge. Moreover the presence of a hypersingular term makes it difficult to deal with. We will see in section 2.5 how we treated it.

Since the trailing edge belongs to the body, here we know the potential and its normal derivative, so we can use the gradient recovery strategy. So, for the position of the first nodes of the wake after the trailing edge, we have to plug the mean value of the gradients of the potential calculated at the upper and lower trailing edges in equation (2.44), namely

$$\nabla\phi(\mathbf{x}_{TE}) = \frac{1}{2}(\nabla\phi(\mathbf{x}_{TE^+}) + \nabla\phi(\mathbf{x}_{TE^-})). \tag{2.45}$$

This last condition ensures that the wake leaves the body exactly with the mean angle between upper and lower trailing edges.

In 3D simulation, we compute the velocities on each strip of nodes and move the first nodes downwind according to equation (2.44), while the other nodes downstream are moved according only to the wind direction $\mathbf{V}_\infty$ as shown in Figure 2.4.



(a) Time step 1.

(b) Time step 2.

(c) Time step 3.

(d) Final time step.

Figure 2.4: Wake relaxation routine in 3D.

2.5 Singular and hypersingular terms

In this section we show how we treat the singular and hypersingular terms coming from the boundary integral formulation. There are some problems arising in the integration terms with $\mathbf{x} - \mathbf{y}$ at the denominator while $\mathbf{y}$ approaches $\mathbf{x}$.

In equation (2.3), there are the terms with $G(\mathbf{x}-\mathbf{y})$ that goes to infinity as $\mathbf{y}$ approaches $\mathbf{x}$, but we need to integrate them. The term $\frac{\partial G}{\partial n}$, that is $\nabla G \cdot \mathbf{n}$, is bounded, even though ∇G is more singular than G. This happens since we have that ∇G and $\mathbf{n}$ become orthogonal, because the gradient is tangential to the surface, so their scalar product is 0. Conversely G becomes singular, since $\frac{1}{4\pi|\mathbf{x}-\mathbf{y}|} = \frac{1}{4\pi|\mathbf{r}|} \to \infty$ as $|\mathbf{r}| \to 0$. To avoid this problem, we can pass in polar coordinates, so that we obtain $|\mathbf{r}|$ from the Jacobian determinant so that the $|\mathbf{r}|$ at the denominator is simplified [27, 28, 29].

Another integration problem arises from (1.53), due to the presence of the hypersingular term $\nabla_{\mathbf{x}} \frac{\partial G(\mathbf{x}-\mathbf{y})}{\partial n}$. This term gives problems when calculating the velocity in equation (1.53), in particular only in the integral concerning the wake surface $\int_{\Gamma_w} [\phi(\mathbf{y})] \nabla_{\mathbf{x}} \frac{\partial G(\mathbf{x}-\mathbf{y})}{\partial n}$. This happens since, in order to know the velocities at the nodes of the wake, the integration variable $\mathbf{y}$ reaches $\mathbf{x}$.

We know the following result [35]

$$\nabla_{\mathbf{x}} \int_{\Gamma_w} [\phi(\mathbf{y})] \frac{\partial G(\mathbf{x}-\mathbf{y})}{\partial n} dS = -\nabla_{\mathbf{x}} \times \int_{\Gamma_w} \gamma(\mathbf{y}) G(\mathbf{x}-\mathbf{y}) dS - \nabla_{\mathbf{x}} \times \int_{\partial \Gamma_w} G(\mathbf{x}-\mathbf{y}) [\phi(\mathbf{y})] \, d\mathbf{l}, \tag{2.46}$$

where $d\mathbf{l}$ is the counter-clockwise direction of Γ_w with respect to $\mathbf{n}$ and

$$\gamma(\mathbf{y}) = \mathbf{n} \times \nabla_S [\phi(\mathbf{y})] . \tag{2.47}$$

We cannot use this formulation to calculate the velocity since the first term remains hypersingular. In order to avoid this integral, if we assume $[\phi(\mathbf{y})] = [\phi]$ constant, we have $\boldsymbol{\gamma} = \mathbf{0}$ because the surface gradient of the potential is null [5]. So we can write

$$\nabla_{\mathbf{x}} \int_{\Gamma_w} [\phi] \frac{\partial G(\mathbf{x}-\mathbf{y})}{\partial n} dS = -[\phi] \nabla_{\mathbf{x}} \times \int_{\partial \Gamma_w} G(\mathbf{x}-\mathbf{y}) d\mathbf{l} = -[\phi] \int_{\partial \Gamma_w} \nabla_{\mathbf{x}} G(\mathbf{x}-\mathbf{y}) \times d\mathbf{l}. \tag{2.48}$$

This integral can be computed, since when $\mathbf{y}$ approaches $\mathbf{x}$ on $\partial \Gamma_w$ the terms $\nabla_{\mathbf{x}} G(\mathbf{x}-\mathbf{y})$ and $d\mathbf{l}$ become parallel, so their vector product is null.

In this work we consider $[\phi]$ constant on mesh cells between two streamlines. Doing so equation (2.48) in numerical form becomes

$$\nabla_{\mathbf{x}} \int_{\Gamma_w} [\phi] \frac{\partial G(\mathbf{x}-\mathbf{y})}{\partial n} dS = \sum_{i=1}^{N_s} \sum_{j=1}^{N_w} [\phi]_i \int_{l_j} \nabla_{\mathbf{x}} G(\mathbf{x}-\mathbf{y}) \times d\mathbf{l}, \tag{2.49}$$

where N_s is the number of trailing edge cells and N_w is the number of streamline cells. We underline that the hypothesis that $[\phi]$ is constant on the wake is used only in this calculation to obtain the velocity, otherwise $[\phi]$ has the same behaviour of ϕ on the body.

2.6 Final simulation algorithm

The pseudocode presented in Algorithm 2 summarizes the complete numerical procedure.

The convergence criterion adopted for the external loop is based on the L^2-norm of the difference of potentials of two iterations that should be lower than a given threshold. The same is done on the position of the wake. The initial guess for the Newton loop is the solution of $\mathbf{Z}\phi = \mathbf{d}$ for the first iteration of the while loop, while for the following

Algorithm 2 Complete algorithm

```
 1: Initialization
 2: Read domain and parameters from external files
 3: Create mesh
 4: while Not converged do
 5:     Compute normals at nodes $\mathbf{n}$
 6:     if First time in while loop then
 7:         Assemble system matrices $\mathbf{A}$ and $\mathbf{N}$ and right-hand side $\mathbf{b}$
 8:         Impose constraints in matrix $\mathbf{C}$
 9:         Assemble final matrix $\mathbf{Z}$ and final right-hand side $\mathbf{d}$
10:         Solve the system without Kutta condition $\mathbf{Z}\phi = \mathbf{d}$
11:     else
12:         Update matrix $\mathbf{Z}$
13:     Newton loop in Algorithm 1
14:     Update wake
```

iterations the initial guess is the solution at the previous step.

The row 12 of the algorithm is done in order to reduce the computational cost. Every wake cycle, we modify the position of the wake. This influences the matrices of the boundary integral equations. Since matrix $\mathbf{Z}$ has the following structure:

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z_{bb}} & \mathbf{Z_{bw}} \\ \mathbf{Z_{wb}} & \mathbf{Z_{ww}} \end{bmatrix}$$

where $\mathbf{Z_{bb}}$ takes into account the contribution of the body on the body itself, $\mathbf{Z_{bw}}$ is the contribution of the wake on the body and so on. Since the wake must satisfy equation (2.14), $\mathbf{Z_{wb}}$ and $\mathbf{Z_{ww}}$ do not change, because every row has a diagonal 1 and on the corresponding upper and lower trailing edges -1 and $+1$, respectively. Moreover, even the matrix $\mathbf{Z_{bb}}$ remains the same, since the contribution of the body is not changed. So it is not necessary to assemble all the matrix $\mathbf{Z}$, but only the section $\mathbf{Z_{bw}}$, that is the influence of the wake on the body. In particular we do not change the matrix $\mathbf{A}$ that is diagonal so it does not compare in $\mathbf{Z_{bw}}$, but we modify only the matrix $\mathbf{N_{bw}}$.

An analogous argument is used for the updating the right-hand side. We have

$$\begin{bmatrix} \mathbf{d_b} \\ \mathbf{d_w} \end{bmatrix} = \begin{bmatrix} \mathbf{D_{bb}} & \mathbf{D_{bw}} \\ \mathbf{D_{wb}} & \mathbf{D_{ww}} \end{bmatrix} \begin{bmatrix} \frac{\partial \phi}{\partial n}_{\mathbf{b}} \\ \frac{\partial \phi}{\partial n}_{\mathbf{w}} \end{bmatrix}$$

Since we have $\mathbf{d_w} = \mathbf{0}$ due to equation (2.14) and $\frac{\partial \phi}{\partial n}_{\mathbf{w}} = \mathbf{0}$ for the mass conservation across the wake (1.31), the only part of matrix $\mathbf{D}$ that contributes to the right-hand side is $\mathbf{D_{bb}}$ that does not change, so we do not have to modify the right-hand side.

In order to give an idea of which is the computational effort for the simulations, in table 2.1 we present some computational times for the simulation in 2D and 3D. In 2D the simulation is done on a NACA 0012 profile with 3083 degrees of freedom (2059 on the body, 1024 on the wake). In 3D the simulation is done on a NACA 0015 wing with 4700 degrees of freedom (2815 on the body, 1885 on the wake).

simulation	body dofs	wake dofs	assemble matrix	Newton loop	wake routine	update matrix
2D	2059	1024	90 s	32 s	50 s	15 s
3D	2815	1885	210 s	450 s	845 s	35 s

Table 2.1: Analysis on the computational cost for 2D and 3D simulations expressed in seconds. We specify the number of degrees of freedom on the body and on the wake. The analysis is done on the time spent in assembling the system matrix, in solving the Newton problem, in updating the wake position and in updating the system matrix.

We can see how updating the system matrix, instead of assembling it from the beginning, implies a significant reduction of the computational times. The most expensive loop even for 2D and 3D simulations is the wake relaxation routine in order to align it as a flow surface.

Chapter 3

Mesh generation

The development of Computer Aided Design (CAD) technologies has posed new challenges on the construction of a mesh that well approximates the surface geometry. With CAD modelers it is possible to represent surfaces with complicated topology and geometry through composite parametric surfaces. These types of surfaces consist of smoothly parametrized patches that meet at their common boundaries (curves and vertices). The approximation of these surfaces should guarantee that the topology and geometric features of the original surface are preserved, the number of elements with respect to the desired resolution has to be small, the elements have to be nicely shaped. In this work the goal has been to generate a curvature-adapted surface mesh refining an extremely coarse initial grid. Using this strategy of meshing the result will be a denser anisotropic mesh where the curvature is high, while nearly flat regions there will be almost equilateral elements. When creating new nodes fulfilling the quality requirements, they will be projected on the surfaces, so that the values for the local surface normals are obtained by the underlying CAD. The method used in our algorithm is based on the projectors illustrated by Dassi, Mola and Si in [42].

A common method for generating grids on CAD surfaces is placing the new nodes on the surface in a parametric way on CAD patches. This approach is not applicable when there are small gaps or overlaps among the patches, but also even in presence of a continuous junction between two patches the mesh can present non conformal elements. In order to avoid these problems, in [42] a surface projector is implemented which queries the CAD data structure to obtain the surface projection of the newly generated points. In practice such projection is realized by intersecting a straight line passing through the point and directed along the specified direction with each of the surface patches. The projected point is selected as the closest intersection with respect to the original point between the line and the surface. In particular the direction used is the normal.

The grid generated with this strategy will not have nodes located on the patches junctions. Moreover, when the grid is refined near an edge, the new nodes are projected on the curve by an arc-projector, that identifies the point splitting the arc at a specified length fraction.

In the resulting algorithm it is possible to refine automatically, using different strategies, a mesh over a CAD file in 2 and 3 dimension using the library `OpenCASCADE` that allows us to deal with CAD surfaces and curves. So the user gives an input parameter file specifying the CAD curves and surfaces to be imported, the refinement strategies to adopt and how many cycles of refinement to carry out. The algorithm is also able to deal with hanging and double nodes (the latter ones being important in the BEM discretization on sharp corners such as the trailing edge).

The 3D meshing strategy already present in π-`BEM` has been extended in this work to treat 2D cases. In addition, we also introduced the management of the wake surface and the identification of the trailing edge nodes. The trailing edge and the wake surface are specified in the user's parameters with an identification flag. It is possible to identify the upper and lower trailing edges and of the wake in order to be able to impose all the conditions for the solution of the problem: the Kutta condition for the upper trailing edge, the BIE (2.17) for the lower trailing edge , the condition (2.14) on the potential jump for the trailing edge wake. The nodes of the wake will be related to the corresponding nodes at the trailing edge, in order to to satisfy the condition of same potential jump on a streamline of equation (2.14). In particular we ensure that in the wake there are no hanging nodes, so that the identification of a streamline required by the relaxation routine procedure results easier.

3.1 2D refinement

For two dimensional geometries the refinement strategy is based either on a global refinement or on the curvature.

The first strategy simply consists in splitting all the elements of the mesh in two.

In the second strategy we first identify the unrefined elements in high curvature regions, as those for which the angle between the cell normal and that of the neighbouring cells is more than a given threshold. For example, considering the situation sketched in Figure 3.1, if $\arccos(\mathbf{n}_2 \cdot \mathbf{n}_1) > \alpha_{threshold}$ we flag the elements for refinement. After all the flags are set, all the segments representing the cells are split in two halves, and the new nodes are projected on the CAD curve of the airfoil.

One problem arising when using these refinement strategies is that two neighbour elements can have very different dimensions. In this case the largest element is selected for refinement.

In two dimensions, in order to obtain $P_{n,1}$ (respectively $P_{n,2}$), the new location is obtained as that of the points that splits the curvilinear distance of points P_1 and P_2 (respectively P_2 and P_3) alon (Figure 3.1).

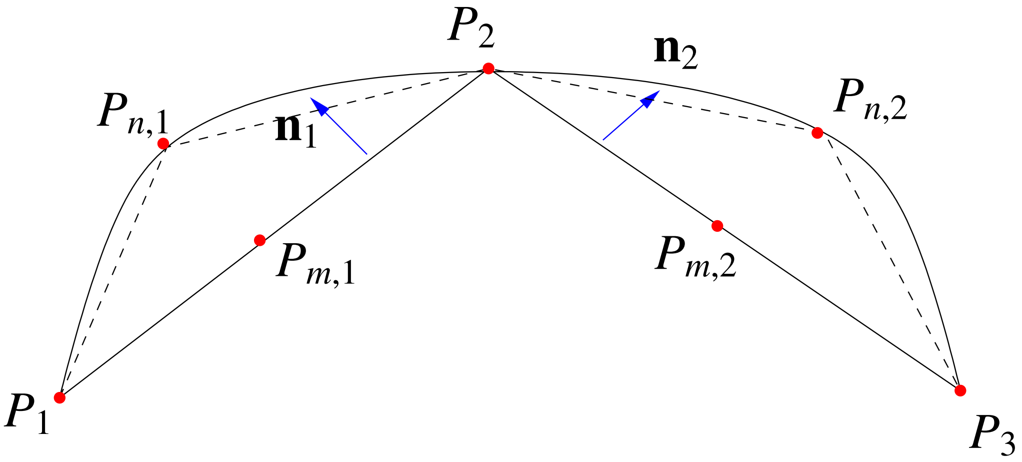


Figure 3.1: 2D mesh.

3.2 3D refinement

In 3D meshing the refinement strategies employed are, as before, global refinement and curvature based refinement. Along with these, an anisotropic refinement strategy based on the aspect ratio has also been employed.
The first strategy consists in dividing all the cells of the mesh in four smaller cells, by bisection (Figure 3.2).
The strategy based on the curvature consists in calculating through an estimate of the mean curvature of the underlying CAD surface within the cell. To obtain such estimate, the CAD geometry is interrogated to obtain the maximum local curvature in all the four nodes of each cell (wich lie on the surface). The average of such maximum local curvature values is computed and the cell is flagged for refinement if the curvature is larger than a certain threshold.
In the third strategy we divide the cells that have aspect ratio larger than a given threshold. Since the aim of such refinement strategy is to obtain cells with low elongation, the cells are only split along their direction of maximum elongation. The aspect ratio is here computed as the biggest ratio between two consecutive edges of the cell. If a low threshold is set, the resultant cells will be similar to squares. Otherwise, they are rectangles of prescribed maximum ratio between the dimensions.
At each refinement step, the new nodes are initially generated on the quadrilateral elements, as shown in Figure 3.2. The new vertices are then projected on the surface along the direction normal to the mesh cells.
One problem arising from this type of meshing is that, if we have two neighbour identical cells and we refine only one of them, we obtain a cell which area is a quarter of the other one. This jump between the dimensions of the two cells can produce some numerical errors in the computations. So such jumps need to be avoided in critical regions, in which strong solution gradients require finer resolutions.

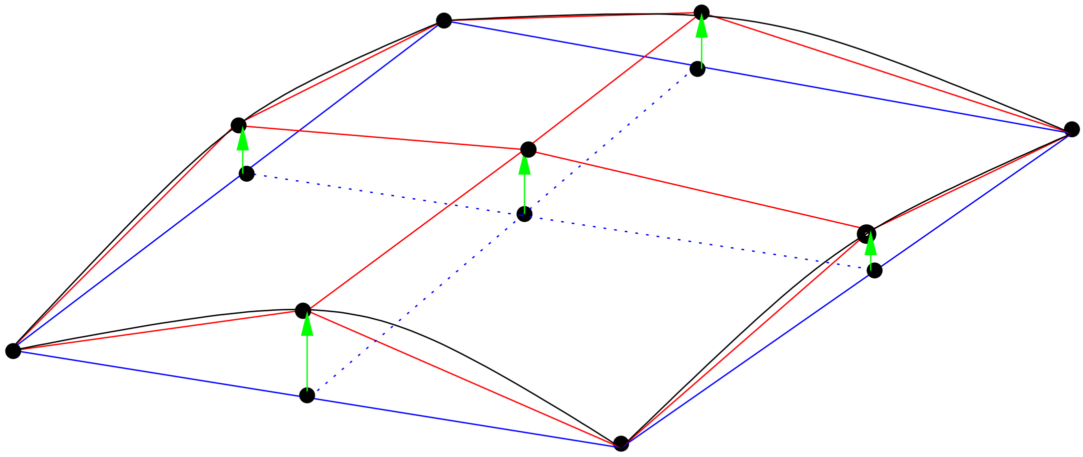


Figure 3.2: 3D mesh refinement of a quadrilateral cell. The new nodes are projected onto the parametric geometry specified by means of a CAD file.

3.3 Hanging and double nodes

During the meshing it is possible to create hanging nodes. These nodes appear when we refine a cell, but not its neighbour (Figure 3.3). In this case all the quantities we calculate on the hanging nodes will be imposed equal to the mean of the quantities evaluated on the two neighbour nodes. For clarity, the potential evaluated in P_3 will not result from a BIE collocated in P_3, but from the imposition of the constraint in the BEM linear system $\phi(P_3) = \frac{\phi(P_1)+\phi(P_2)}{2}$. The same constraint is enforced for $\frac{\partial\phi}{\partial n}$, $\nabla\phi$, the pressure p.

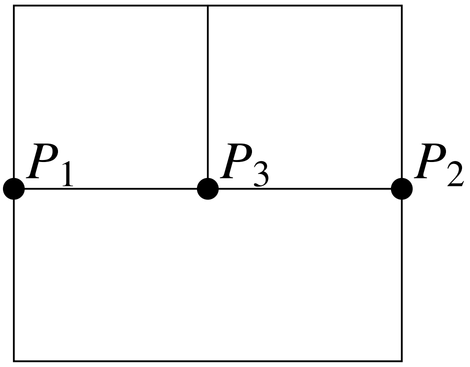


Figure 3.3: Hanging nodes.

Double nodes appear on the contrary when two cells belong to two different surfaces, but have a common edge on a curve (Figure 3.4). In this case we have two logically different vertices which lie in the same geometric position. This results in the duplication of the degrees of freedom on the edges, which results in the possibility to impose more than one condition at the trailing edge as we have already seen in section 2. As for the nodes of the other edges, the constraint imposed is that the potential calculated on one side of the edge is equal on the other one. Namely, considering Figure 3.4, the potential evaluated in P_2 will be $\phi(P_2) = \phi(P_1)$. In this case, due to the fact that we have different normal vectors, we will have different $\frac{\partial\phi}{\partial n}$.

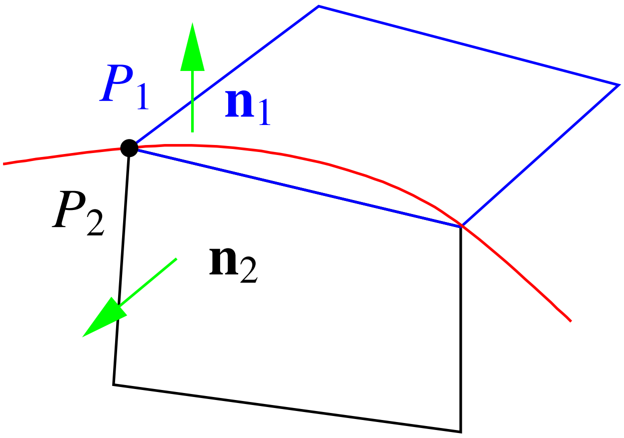


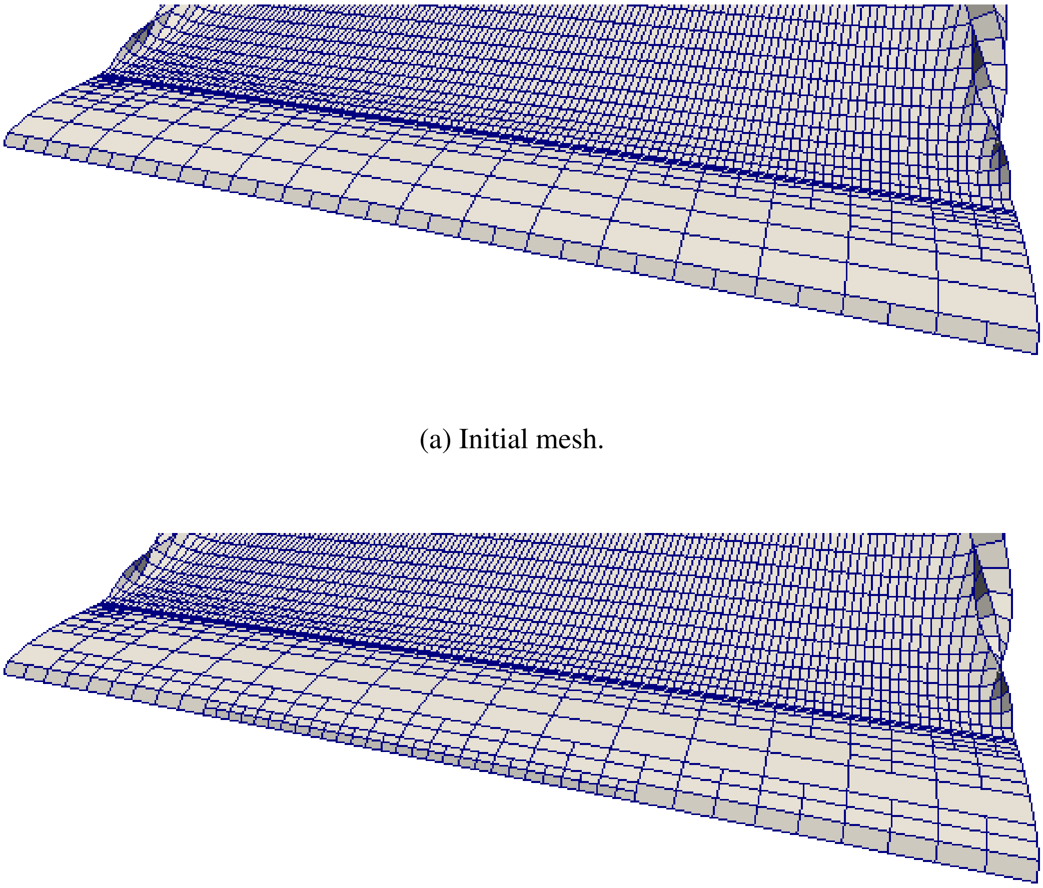
Figure 3.4: Double nodes.

3.4 Adaptive refinement

To further exploit the mesh refinement capability described in the previous section, we have also considered a strategy for the adaptive refinement of the grid based on the solution. In order to select the cells to refine, we adopted a strategy based on an error estimator on the pressure coefficient. The error estimator requires the evaluation of the gradient of the pressure coefficient on each cell and compute the jump of this quantity between two neighbour cells. Where this difference is large, the error is large and the cells are refined, otherwise not. If it is necessary to refine, the selected cell is divided in four smaller cells. The error indicator introduced by Kelly, Gago, Zienkiewicz and Babuska in [43] approximates the error on each cell by integrating the jump of the gradient solution along the faces of each cell,

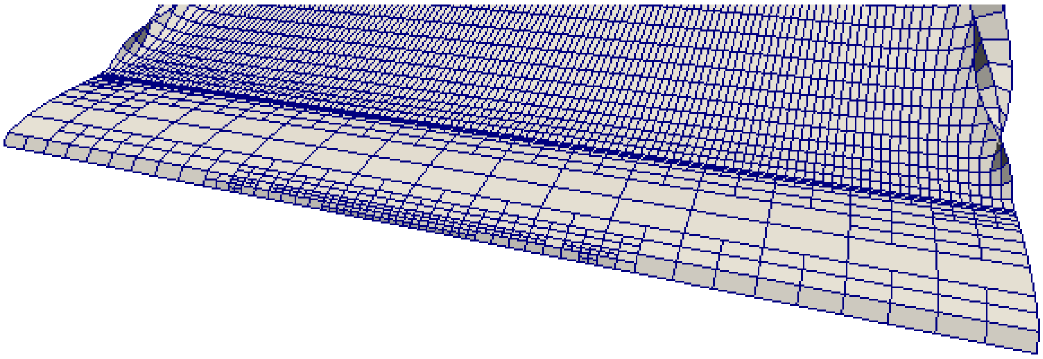
$$\eta_K^2 = \sum_{F\in\partial K} \frac{h_F}{2p_F} \int_{\partial K_F} \left[\frac{\partial u_h}{\partial n}\right]^2 do, \tag{3.1}$$

where η_K is the error estimator on the cell K, h_F is the face diagonal, p_F is the maximum polynomial degree of adjacent elements, $\left[\frac{\partial u_h}{\partial n}\right]$ is the jump of the gradient of the prescribed quantity at the face, the summation is done on the cell's faces and the integration is computed on the boundary edges of the current cell. This operator was already implemented in `deal.II` [2, 3] but it is used in this work to compute the error on the pressure coefficient only for the cells on the body. In particular we select the cells that contributes as a certain percentage on the global error set in input by the user. In Figure 3.5 it is shown the procedure on a NACA wing, where the mesh is refined at the leading edge where high gradients of the pressure coefficient are present. The adaptive refinement concentrates in the middle of the leading edge, since in the lateral parts of the wing there are lower gradients of velocity and pressure.

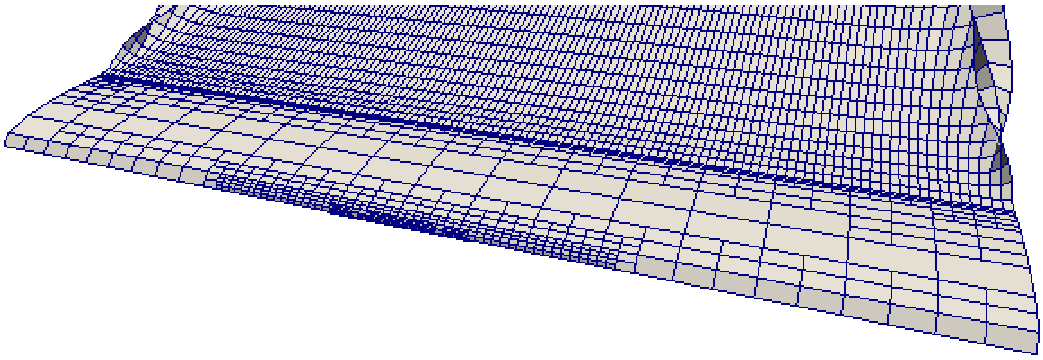


(a) Initial mesh.

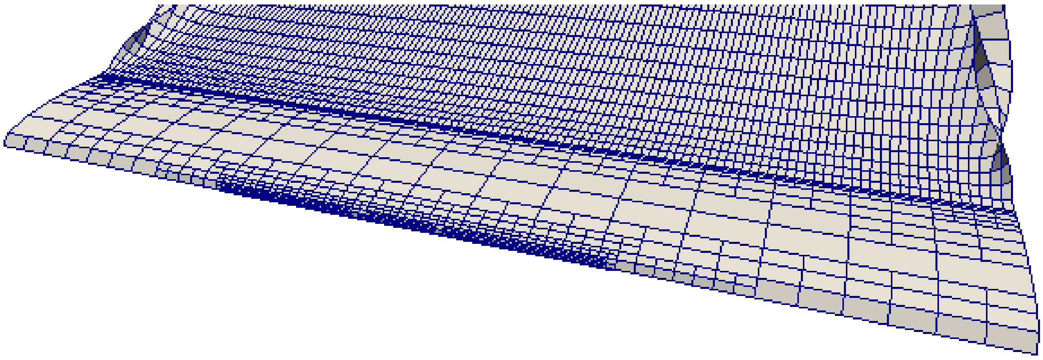
(b) Adaptive refinement cycle 1.



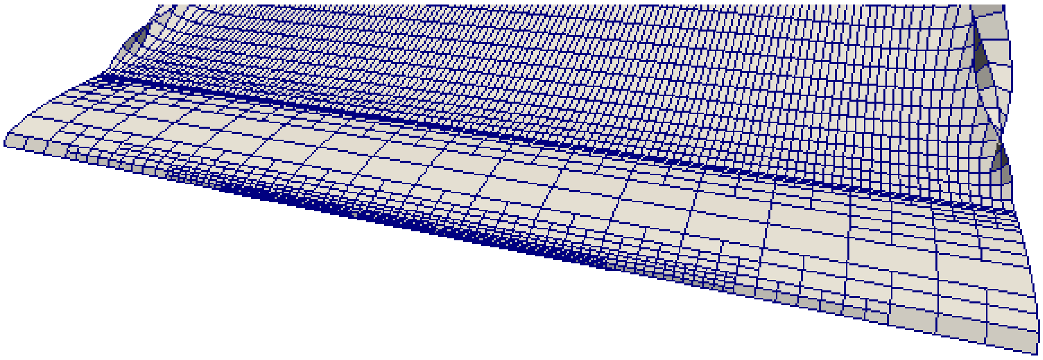
(c) Adaptive refinement cycle 2.



(d) Adaptive refinement cycle 3.



(e) Adaptive refinement cycle 4.



(f) Adaptive refinement cycle 5.

Figure 3.5: Mesh adaptivity over a rectangular wing with NACA 0015 airfoil. Starting from the mesh in plot a) we refine the grid according to the error estimator until we obtain the mesh in plot f).

Chapter 4

Numerical results

In this chapter we will present the numerical results obtained in this work to provide an adequate assessment of the accuracy mathematical and numerical model introduced in the previous chapters. Such results will be compared to equivalent results obtained with other numerical models and with experimental data available in the literature.
Most of the results here reported will refer to 2D airfoils and 3D wing shapes obtained as extrusions of airfoil sections. For such reason, we here recall some aeronautic terminology used in the following analysis (Figure 4.1). For each section of the wing the chord is the imaginary straight line joining the leading and the trailing edge. The chord length c is the distance on such line between leading and trailing edge. For each airfoil section, a mean (or camber) line can be identified along with a distribution of thickness. We indicate the maximum airfoil thickness with d. In 3D wing the maximum extension s in the direction normal to the airfoil sections is indicated as span. Finally we define the angle of attack α as the angle between the asymptotic speed $\mathbf{V}_\infty$ and the chord. The aspect ratio is defined as $AR = s/c$.

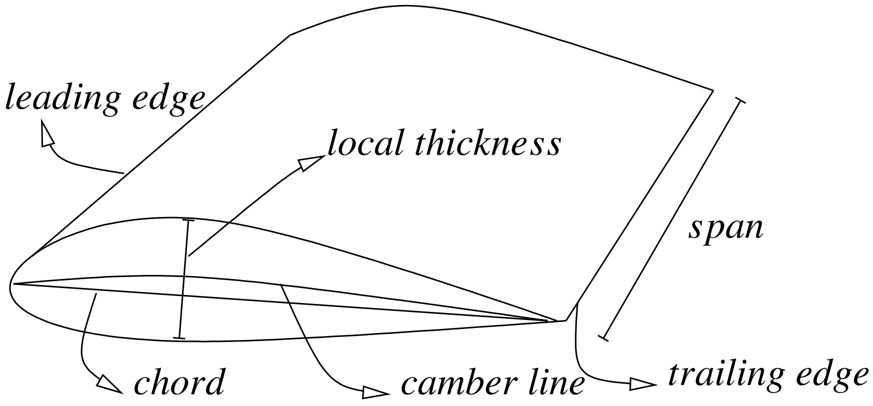


Figure 4.1: This figure shows the different parameters defining chord, camber line, local thickness and span.

4.1 2D results

In this section we will first present some analysis on the influence of grid geometrical and wake parameters on the model predictions. After such assessment, we will proceed comparing the model results with corresponding literature results.

4.1.1 Grid convergence analysis

In this paragraph we report the results of analysis on the convergence of the solution with respect to the mesh size on the body, or, in alternative, as a function of the degrees of freedom.

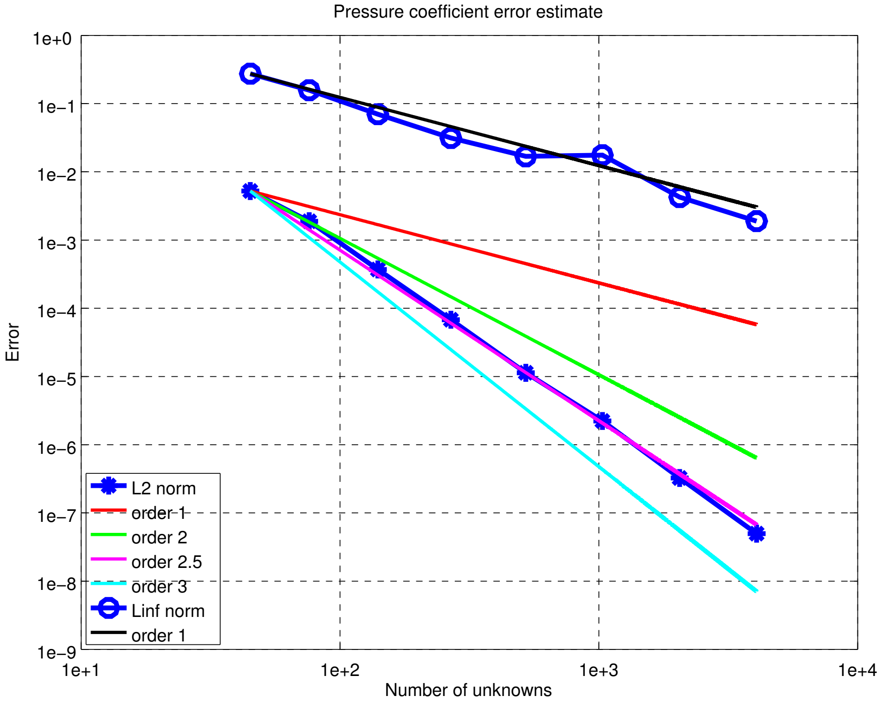


Figure 4.2: Pressure coefficient convergence as the number of degrees of freedom of the airfoil grid is increased. The circled blue line indicates the L_∞-norm of the difference between the c_p obtained for the current solution and the solution with 8152 degrees of freedom. For reference, order 1 (black line) convergence rate curves is also included. The asterisks and the corresponding blue line represent the L_2-error as a function of the unknowns. As in the previous case, the error is computed as the L_2-norm of the difference between the solution at the current grid and that at 8152 degrees of freedom. For reference, order 1 (red line), 2 (green line), 2.5 (violet line) and 3 (cyan line) convergence rate curves are also included.

Such analysis has been carried out considering the flow past a NACA 0012 profile with angle of attack $\alpha = 6°$. The study is done comparing solutions obtained with numbers of unknowns increasing from 45 to 8152. For the present investigation, the number of unknowns is intended as the number of degrees of freedom on the body. Hence, we do not take into account the number of unknowns of the wake. Also for this test, due to the lack of an analytic solution, we define the error as the difference between c_p for the best available grid and that for the other ones. With L_2- and L_∞-errors we indicate the L_2- and L_∞-norms of the error defined in such way.

Figure 4.2 illustrates the convergence of L_2- and L_∞-norms of the pressure coefficient errors as a function of the number of degrees of freedom. We must remark that we do not take into account the last cells at the trailing edge, since, if the grid becomes too much refined in that region, the pressure coefficient starts to oscillate and assume non physical values. As pointed out in [1], this might be related to specific problems related to the singular quadrature implementation in the Laplace solver employed.

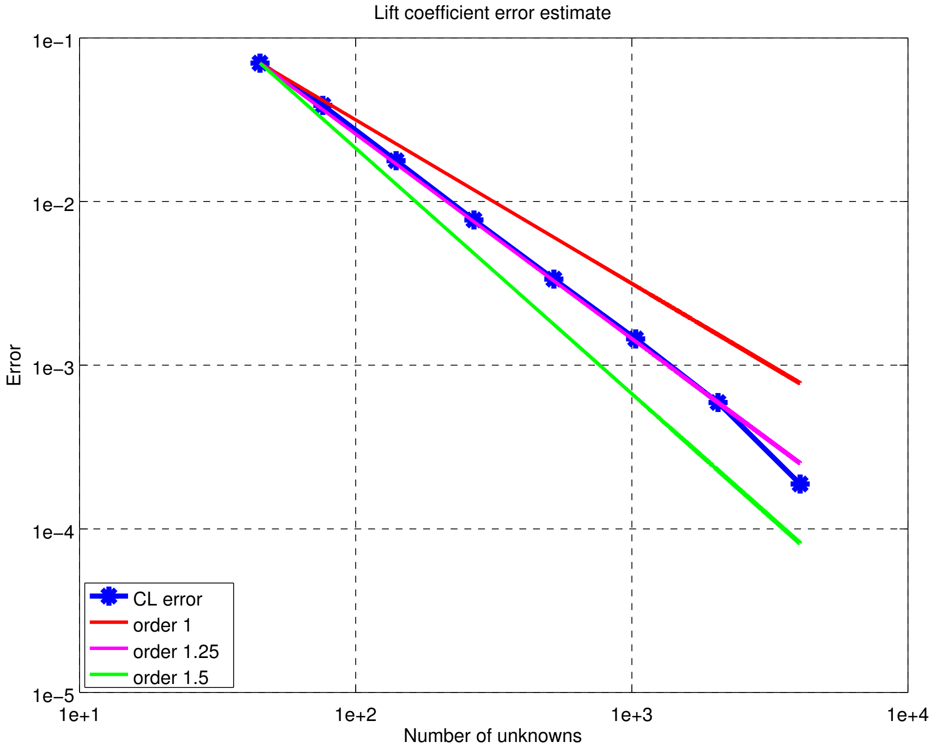


Figure 4.3: Lift coefficient convergence as the number of degrees of freedom of the airfoil grid is increased. The asterisks and the corresponding blue line represent the error as a function of the unknowns. The error presented is evaluated as the difference between the C_L at the current grid and that at 8152 degrees of freedom. For reference, order 1 (red line), 1.25 (green line) and 1.5 (violet line) convergence rate curves are also included.

We can see that the rate of convergence of the L_2-error (blue asterisks) fits the trend of the violet line that represents the convergence order of 2.5, so the L_2-error is $o((\frac{1}{n})^{2.5})$ with n number of degrees of freedom on the body. The L_∞-error's convergence (circled blue line) follows the black line that represents the linear convergence rate, so the L_∞-error is $o(\frac{1}{n})$. We remark that, since in these analysis we used a global refinement strategy, the number of degrees of freedom doubles at each refinement cycle and the length of each element is halved.

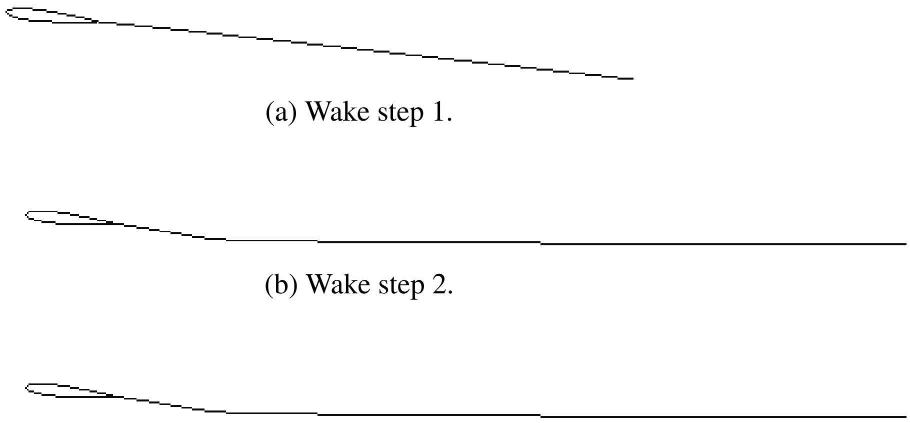
The plot suggests that an L_2-error of 10^{-2} is reached with a simulation with only 45 degrees of freedom. If we want instead to have $L_\infty = 10^{-2}$ we have to require a finer mesh, which reaches more than 1000 degrees of freedom.

A similar analysis on the convergence of the lift coefficient C_L is shown in Figure 4.3. Here the error is evaluated as the difference between the C_L with 8152 degrees of freedom and C_L obtained with coarser grids.

The rate of convergence of the lift coefficient (blue dotted line) fits the violet line that represents the convergence order of 1.25, so the error on the lift coefficient is $o((\frac{1}{n})^{1.25})$. In this case, an error of 10^{-2} is reached for 400 degrees of freedom.

4.1.2 Wake iterations analysis

A first investigation aimed at assessing how many iterations in 2D simulations are required to reach convergence of the wake geometry. This qualitative analysis has been carried out on the geometry of a NACA 0012 airfoil and is presented in Figure 4.4.



(a) Wake step 1.

(b) Wake step 2.

(c) Final wake step.

Figure 4.4: NACA 0012 airfoil and wake profile for $\alpha = 6°$ at different wake relaxation steps. The upper figure presents the geometry at the first step. The middle figure depicts the wake after two steps, when the Newton problem has been solved two times. The lower figure presents the wake configuration at convergence.

By a qualitative standpoint, we can see that at the second iteration the wake has already reached convergence as the second and final iteration look identical. Figure 4.4c also shows how the wake leaves the body tangentially with respect to the trailing edge orientation. Far from the body it instead aligns to the wind direction $\mathbf{V}_{\infty}$, thus, as expected the wake seems to assume the shape of a streamline.

4.1.3 Length wake analysis

In this paragraph we assess the influence of the wake length on the body potential field and the pressure coefficient c_p. To do so, we obtained an overall evaluation of the influence of the wake length on the body solution by computing the value of the lift coefficient C_L and plot it against the wake length (see Figure 4.5). Once again the analysis has been carried out on the NACA 0012 airfoil geometry at $\alpha = 6°$.

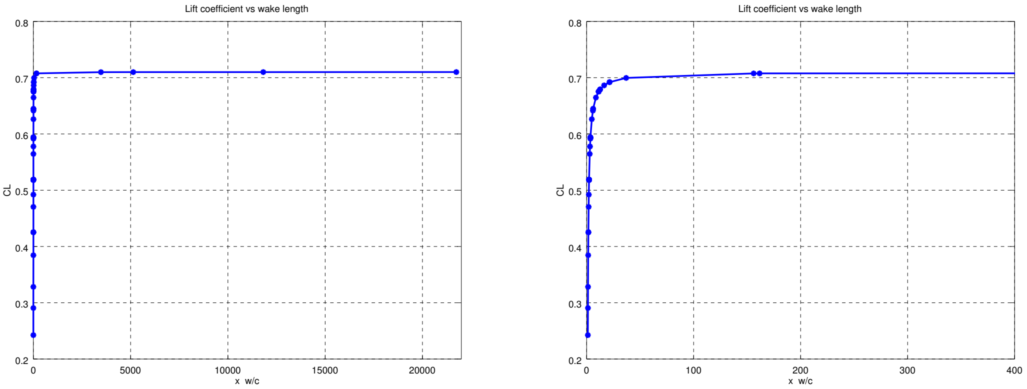


Figure 4.5: Lift coefficient versus wake length. The left plot depicts the computed C_L as a function of the considered wake length. The right plot represents a close up of the previous one, so as to better appreciate the influence of wake length on C_L in the lower ranges.

The plots suggest that after approximatively 70 chord lengths, C_L has reached a value differing 1% from the final one, obtained for a wake measuring 21000 c. For a C_L differing 0.1% from the final one, the plots suggest a wake length of 300 c.
We then evaluate the error on the lift coefficient as the difference between the solution obtained with wake length 21000 c and the solutions for different wake lengths.
In Figure 4.6 we can see that the error trend on the lift coefficient (blue dotted line) fits the linear convergence rate, thus the error is $o(\frac{1}{x_w/c})$. This convergence rate represents the expected behaviour, as the cells in the wake exert their influence on the body through the Green function gradient, that in 2D dacays as $\frac{1}{r}$ as the last point of the wake reaches a distance r from the airfoil.

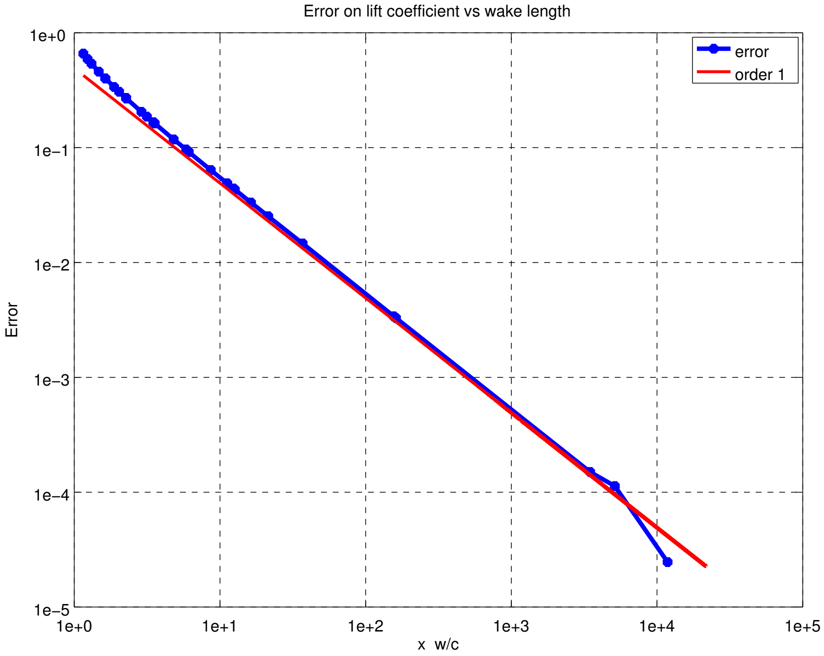


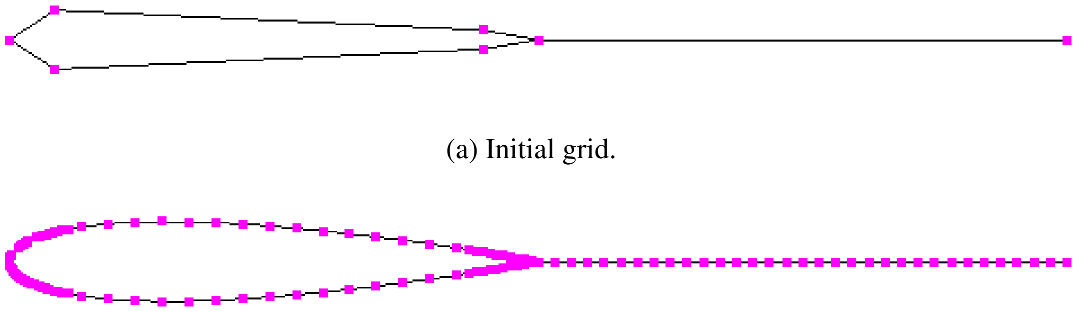
Figure 4.6: Lift coefficient convergence for increasing wake lengths. For each x_w in the blue dotted line, the error presented is evaluated as the difference between the C_L for $x_w = 21000\,c$ and the current one. The red continuous line, added for reference, indicates a linear convergence rate.

4.1.4 NACA 0015

To run the NACA 0015 test case, we first produced a CAD curve describing the profile of such symmetric airfoil. Then, once the initial grid composed of 7 cells is specified, the desired grid is obtained only prescribing the parameters for the automatic mesh refinement. Figure 4.7 portrays the result of the mesh refinement procedure with a grid that accounts for 206 nodes.

Figure 4.8 depicts a comparison of the computed pressure coefficient c_p and lift coefficient C_L with experimental results on NACA 0015 at different angles of attack. Plots a), b), c) respectively show the pressure coefficient distribution along the airfoil chordwise coordinate for angles of attack $\alpha = 0°, 6°, 12°$. In the case of $\alpha = 0°$, the comparison is done with experimental results from Abbott-VonDoenhoff-Stivers [45] (green line) and McAlister-Takahashi [44] (red line). In the cases of $\alpha = 6°$, $12°$ and in the analysis of the lift coefficient as a function of α the comparison is done only with McAlister-Takahashi.

Figure 4.8 shows that, for $\alpha = 0°$, the results of the developed model (blue line) fit with good accuracy the experimental data by Abbott [45] (green curve). The measurements presented in McAlister [44] NASA report (red curve) appear to overestimating both the present c_p curve and the Abbott's one by a somewhat constant offset of approxima-

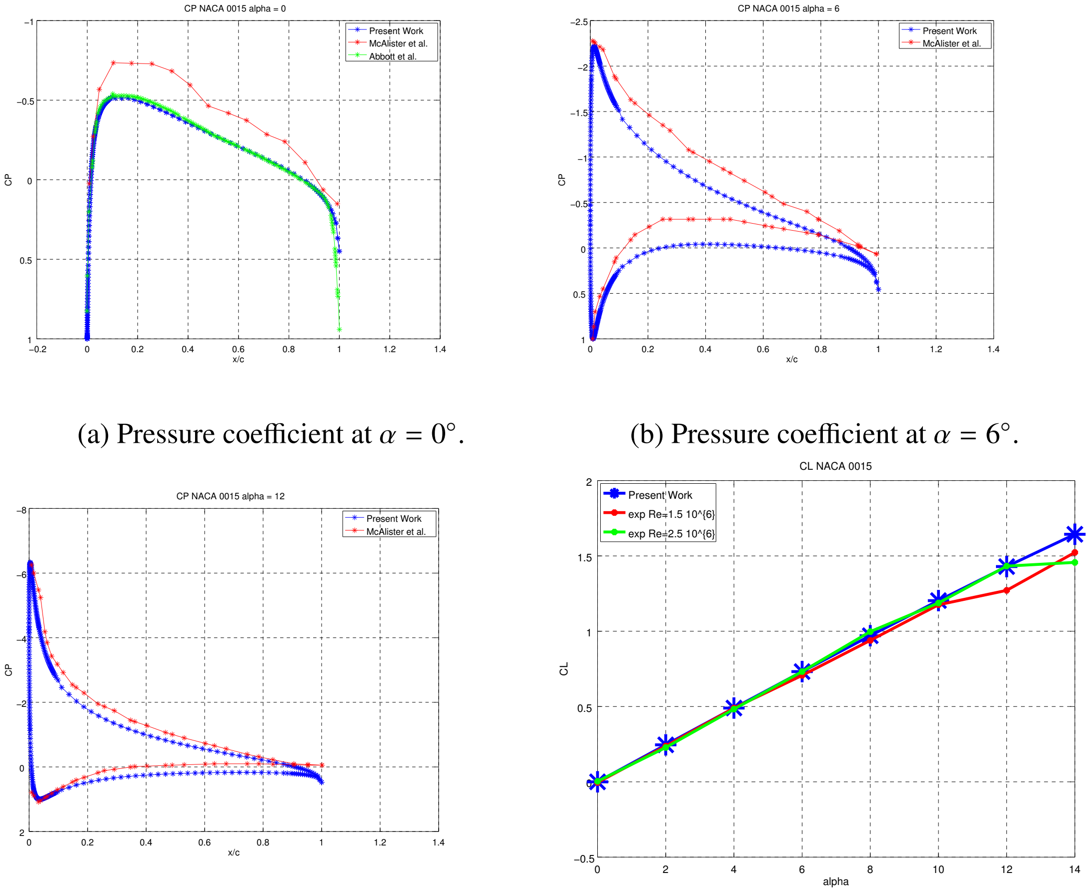


(a) Initial grid.

(b) Final grid.

Figure 4.7: On the top there is the initial two dimensional mesh of a NACA 0015 profile, that is the input file given to the algorithm. On the bottom it is presented the final grid after global refinements and curvature based strategy.

tively 0.2. The same trend is observable in the plots regarding $\alpha = 6°, 12°$ for which Abbott's results are not reported. Still the comparison with McAlister data suggests that our model consistently underestimates the pressure coefficient coming from those experiments by an approximatively constant 0.2 offset. In their report, the authors of such experimental campaign seem to suggest that the reduced wind tunnel dimension makes the comparison to numerical simulations of wings in free stream rather inaccurate. Indeed a more meaningful comparison would be obtained considering the actual wind tunnel dimensions in the simulations, so as to account for the blocking effect that might be responsible for the lower experimental c_p. Despite such underestimation of the c_p data, the lift coefficient data by McAlister seems to be correctly reproduced at most angles of attack, as can be appreciated in Figure 4.8d. We see in fact that for α below 10° the predicted curve matches the experimental data, while in the region between 10° and 14° the model of potential flow is not able to reproduce the incipient flow detachment leading to the $C_L - \alpha$ curve slope reduction. Such problem is particularly noticeable at lower *Re*.
We must remark that the possibility to refine the mesh where the CAD surface has a higher curvature results in an accurate reproduction of the stagnation point in the front of the body where c_p has unit value, since in that point the fluid has null velocity. Automatically refined grid reproduces in the proper way the leading edge region and its normal distribution.



(a) Pressure coefficient at $\alpha = 0°$.

(b) Pressure coefficient at $\alpha = 6°$.

(c) Pressure coefficient at $\alpha = 12°$.

(d) Lift coefficient at different angles of attack.

Figure 4.8: Comparison between numerical predictions (blue line) and experimental data referred to a NACA 0015 airfoil presented in [44] (red curve) and [45] (green curve). Plot a) depicts the local pressure coefficient c_p as a function of the chordwise coordinate for a null angle of attack α. Plots b) and c) present similar c_p curves for $\alpha = 6°$ and $\alpha = 12°$ respectively. Finally, plot d) shows the lift coefficient C_L as a function of the angle of attack α.

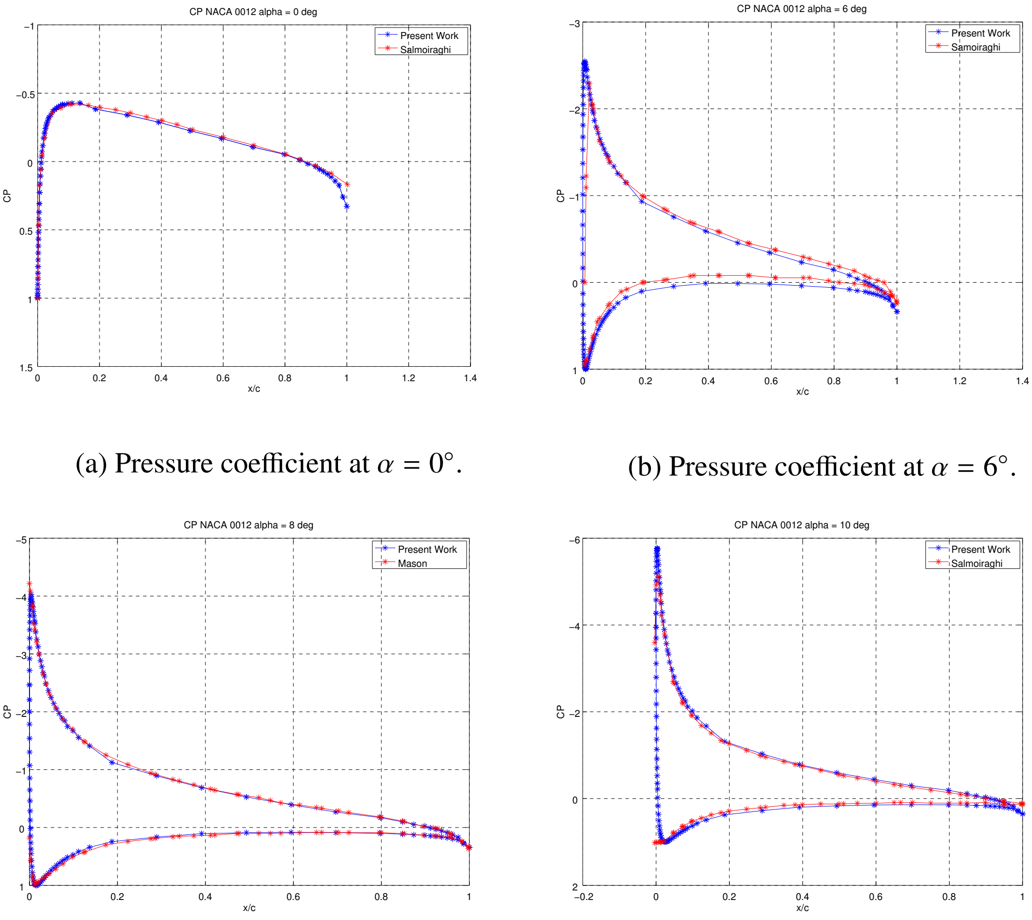
4.1.5 NACA 0012

Figure 4.9 presents a comparison of the pressure coefficient computed in this work, between numerical and experimental results on NACA 0012. Such symmetric airfoil has a lower thickness with respect to the NACA 0015 which typically implies higher leading edge curvatures, and results in higher c_p values. The analysis carried out consisted in computing the trend of the pressure coefficient distribution on the airfoil for angles of attack $\alpha = 0°, 6°, 8°, 10°, 12°, 15°$. The predicted c_p curves are compared against numerical results available in the literature [7, 46, 47, 48].

The plots suggest that the results of the present model (blue lines) are very similar to the numerical and experimental data reported for comparison (red lines). Also here, as was the case for NACA 0015, the automatic grid refinement in the high curvature regions results in an accurate reproduction of the stagnation point where c_p has unit value.

Figure 4.10 presents the lift coefficient curve as a function of the angle of attack α. The plot reports the comparison between the present model prediction (blue line with asterisks) and experimental data from Abbott et al. [45] (red line).

The trend of the lift coefficient is well fitted. As already pointed out for the NACA 0015 analysis, we see that if the angle of attack is bigger than 15° the model of potential flow is, as expectable, not able to reproduce the stall occurring due to the flow separation. In fact the lift coefficient calculated in our work has a linear trend also for high angles of attack.

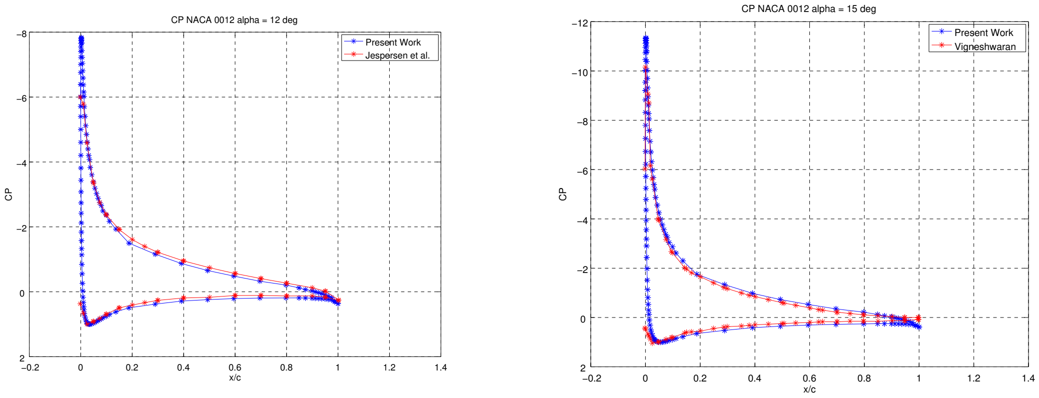


(a) Pressure coefficient at $\alpha = 0°$.

(b) Pressure coefficient at $\alpha = 6°$.

(c) Pressure coefficient at $\alpha = 8°$.

(d) Pressure coefficient at $\alpha = 10°$.



(e) Pressure coefficient at $\alpha = 12°$.

(f) Pressure coefficient at $\alpha = 15°$.

Figure 4.9: Comparison between numerical predictions (blue line) and numerical data referred to a NACA 0012 airfoil presented in [7, 46, 47, 48] (red curve). Plot a) depicts the local pressure coefficient c_p as a function of the chordwise coordinate for a null angle of attack α. Plots b), c), d), e) and f) present similar c_p curves for $\alpha = 6°$, $\alpha = 8°$, $\alpha = 10°$, $\alpha = 12°$ and $\alpha = 15°$ respectively.

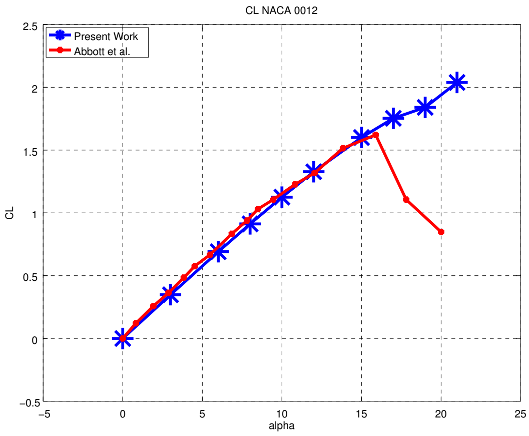
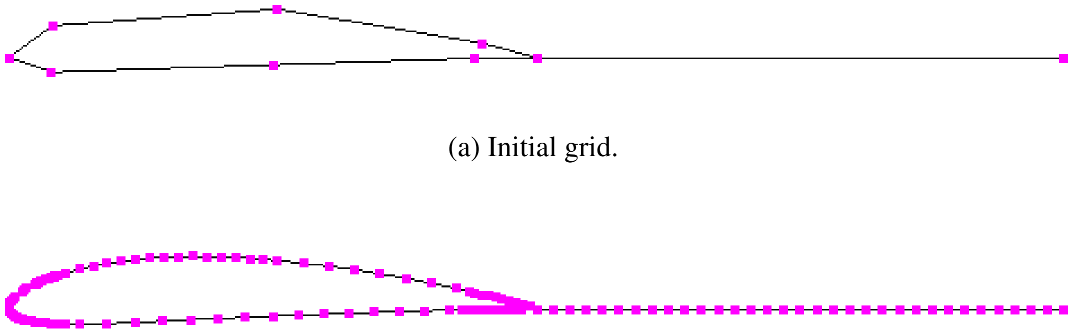


Figure 4.10: Lift coefficient of NACA 0012 as a function of the airfoil angle of attack. The present work numerical prediction (blue curve) is here compared with measurements (red curve) reported in [45].

4.1.6 NACA 4412

To run the NACA 4412 test case, the mesh has been automatically generated as illustrated in the previous cases. The initial grid composed of 7 cells is refined in order to obtain a resultant grid that accounts for 220 nodes as shown in Figure 4.11.
Figure 4.12 depicts a comparison of the computed pressure coefficient c_p with numerical results on NACA 4412 at different angles of attack available in the literature [7, 47]. Plots a), b), c), d), e) respectively show the pressure coefficient distribution along the



(a) Initial grid.

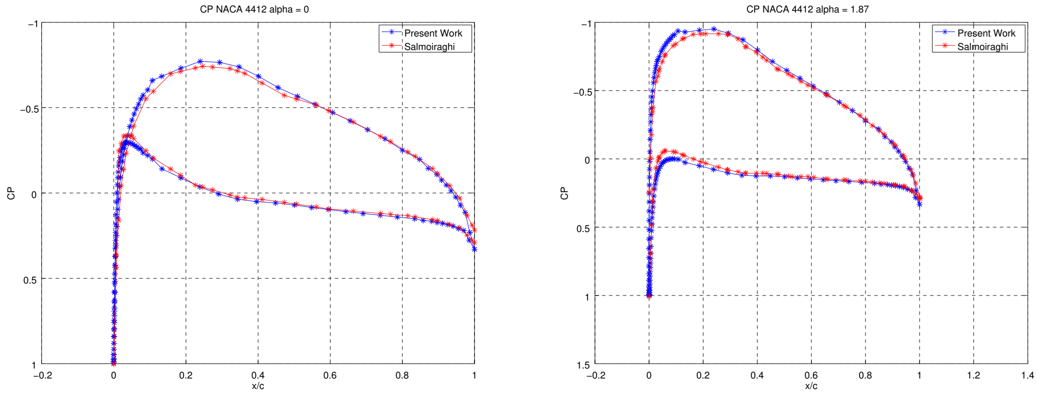
(b) Final grid.

Figure 4.11: On the top there is the initial two dimensional mesh of a NACA 4412 profile, that is the input file given to the algorithm. On the bottom it is presented the final grid after global refinements and curvature based strategy.

airfoil chordwise coordinate for angles of attack $\alpha = 0°, 1.87°, -2°, -5°, -10°$.
We see that even for this non symmetric airfoil this work's results (blue lines) are fairly similar to the numerical results (red line). Also in this case, as well as in the previous ones, the numerical predictions correctly reproduce the leading edge stagnation point, where the pressure coefficient is one. Again, this is made possible by refining high curvature regions.

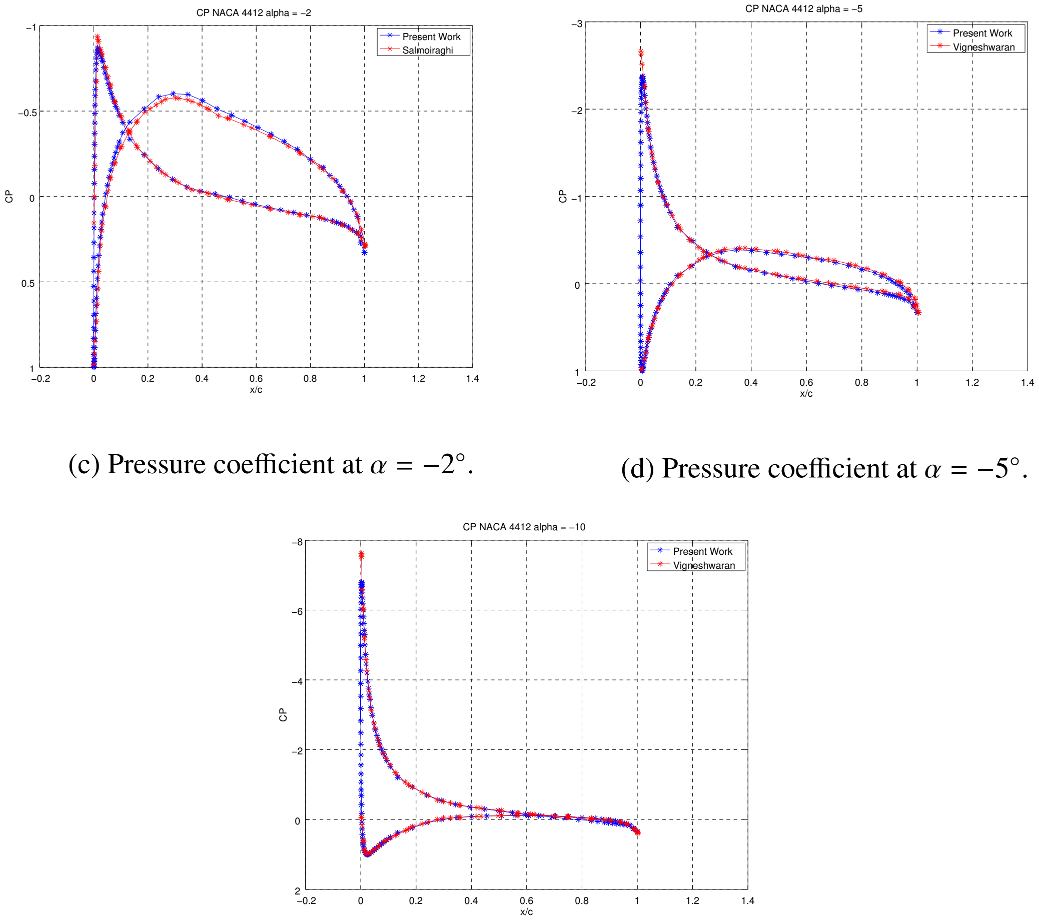
Finally, the predicted lift coefficient for the NACA 4412 as a function of the angle of attack is compared both with numerical [47, 48] (red and green lines) and with experimental data [45] (black line) (Figure 4.13).

The trend of the lift coefficient is well fitted in the central part. As already mentioned for the previous comparisons, we see that if the angle of attack is bigger than 10° or



(a) Pressure coefficient at $\alpha = 0°$.

(b) Pressure coefficient at $\alpha = 1.87°$.



(c) Pressure coefficient at $\alpha = -2°$.

(d) Pressure coefficient at $\alpha = -5°$.

(e) Pressure coefficient at $\alpha = -10°$.

Figure 4.12: Comparison between numerical predictions (blue line) and numerical data referred to a NACA 4412 airfoil presented in [7, 47] (red curve). Plot a) depicts the local pressure coefficient c_p as a function of the chordwise coordinate for a null angle of attack α. Plots b), c), d), e) and f) present similar c_p curves for $\alpha = 1.87°$, $\alpha = -2°$, $\alpha = -5°$ and $\alpha = -10°$ respectively.

lower than −10°, the model of potential flow is not accurate. In fact the lift coefficient calculated in this work has a linear trend also for higher or lower angles of attack. On the contrary in experimental data and in numerical data from non potential works, the pressure has a drop if the angle of attack is bigger than 10° or smaller than −10°. As pointed out, this expected behaviour is the consequence of the inability to observe a flow separation through a potential flow model.

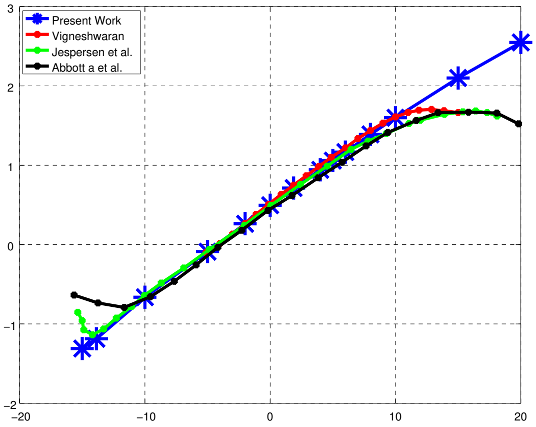


Figure 4.13: Lift coefficient of NACA 4412 as a function of the airfoil angle of attack. The present work numerical prediction (blue curve) is here compared with numerical data reported in [47] (red curve) and [48] (green curve) and measurements reported in [45] (black curve).

4.2 3D results

This section will present some analysis carried out to establish a suitable confidence on the behaviour of the three dimensional version of the fluid dynamic solver developed. Such analysis, which involves assessments of the solution sensitivity to wake position and grid refinement, will be the base for the comparison against experimental results presented in the next section. Then numerical results will be tested against numerical and experimental data available in the literature for a variety of benchmark problems. We will then present some further results obtained with the 3D simulations so as to highlight the model ability to readily tackle problems with arbitrary geometries.

4.2.1 3D wake analysis

In this section we analyse the influence of the wake on the predicted flow past a rectangular NACA 0015 wing with aspect ratio $AR = 6.58$ at an angle of attack of 12°. The influence of the wake is analysed considering increasing wake length and looking at the central section of the wing. A further set of simulations aimed at understanding how the wake geometry (and in particular its roll up) at different streamwise distances from the body is affected by the number of wake iterations performed in the simulation and by the grid refinement.

Figure 4.14 reports the convergence of the L_2- and L_∞-norm of the errors on the pressure coefficient at the central section $z/s = 0$ as the length of the wake considered in the simulations is increased. The error presented in the plot is computed as the difference between the c_p obtained for each solution, and that computed with the best available wake length ($\sim 460\,c$).

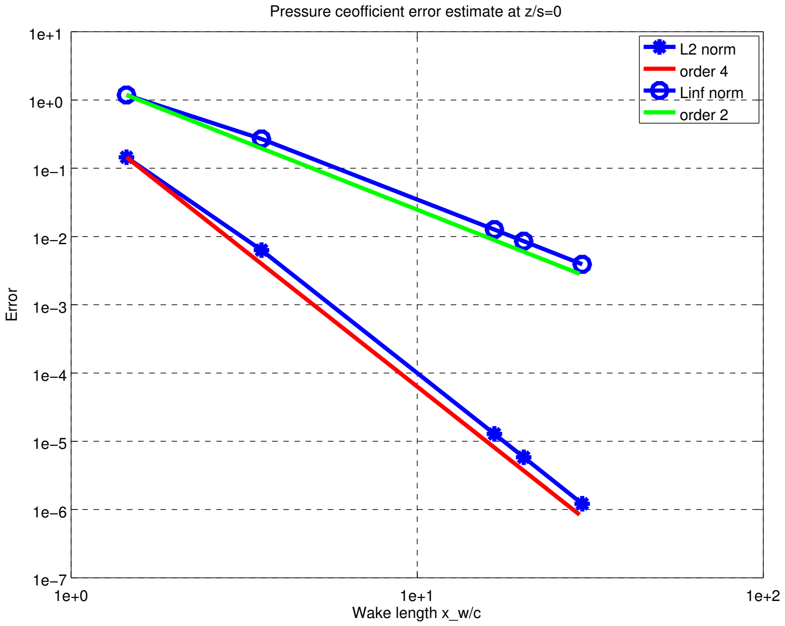


Figure 4.14: Pressure coefficient error as the wake length increases at the section at $z/s = 0$ for the wake past a NACA 0015 rectangular wing with AR=6.58 at an angle of attack $\alpha = 12°$. The blue line with circles indicates the L_∞-norm of the difference between the c_p obtained for each solution and the solution with wake length of 460. For reference, order 2 (green line) convergence rate curves is also included. The asterisks and the corresponding blue line represent instead the L_2-error as a function of the wake length. For reference, order 4 (red line) convergence rate curves is also included.

We can see that the rate of convergence of the L_2-error (blue asterisks) fits the trend of the red line that represents the convergence order of 4, so the L_2-error is $o((\frac{1}{x_w/c})^4)$. The L_∞-error's convergence (circled blue line) follows the green line that represents the quadratic convergence rate, so the L_∞-error is $o((\frac{1}{x_w/c})^2)$. Such convergence rates appear significantly higher than those observed in 2D simulations. This is not surprising. We recall in fact that the cells in the wake exert their influence on the body surface through the Green function gradient appearing in the BIE. In the 2D case such function decays as $\frac{1}{r}$ as the last point of the wake reaches a distance r from the body. In the 3D case instead, the Green function gradient decays as $\frac{1}{r^2}$.

The result of such investigation suggests that to obtain a difference in L_2-norm of 10^{-6} with respect to the best available solution the wake has to be long at least 30 times the body length. In such case the L_∞-error will be of 10^{-2}.

A similar analysis is done on the convergence of the lift coefficient, as shown in Figure 4.15. Here the error presented is evaluated as the difference between the C_L obtained for each solution and that obtained for the solution with wake length of 460 c.

The rate of convergence of the lift coefficient (blue dotted line) fits the red line that

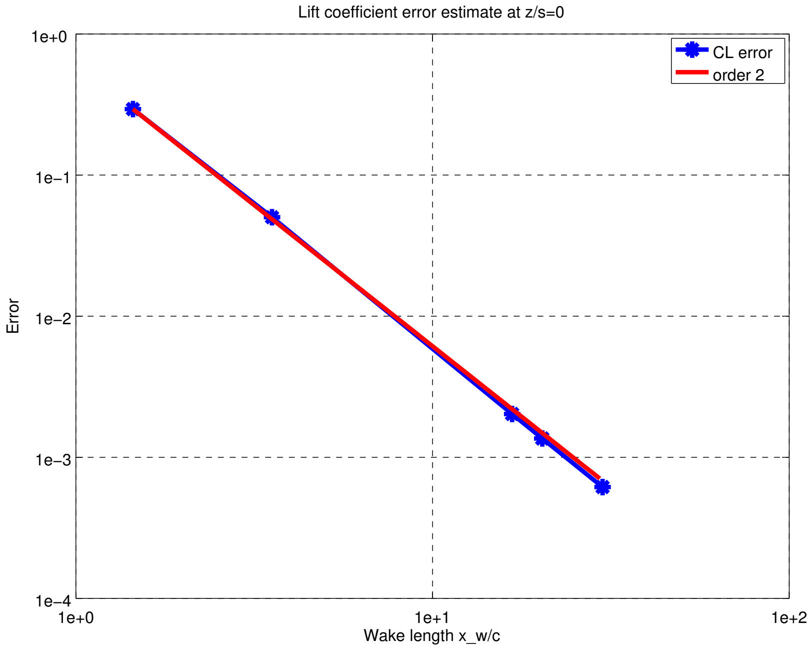


Figure 4.15: Lift coefficient error as the wake length increases at $z/s = 0$ for the wake past a NACA 0015 rectangular wing with AR=6.58 at an angle of attack $\alpha = 12°$. The asterisks and the corresponding blue line represent the error as a function of the wake length. For reference, order 2 (red line) convergence rate curves is also included.

represents the convergence order of 2, so the lift coefficient converges quadratically as the wake length is increased. If we want an error of 10^{-3}, the wake has to be long at least 30 times the body length.

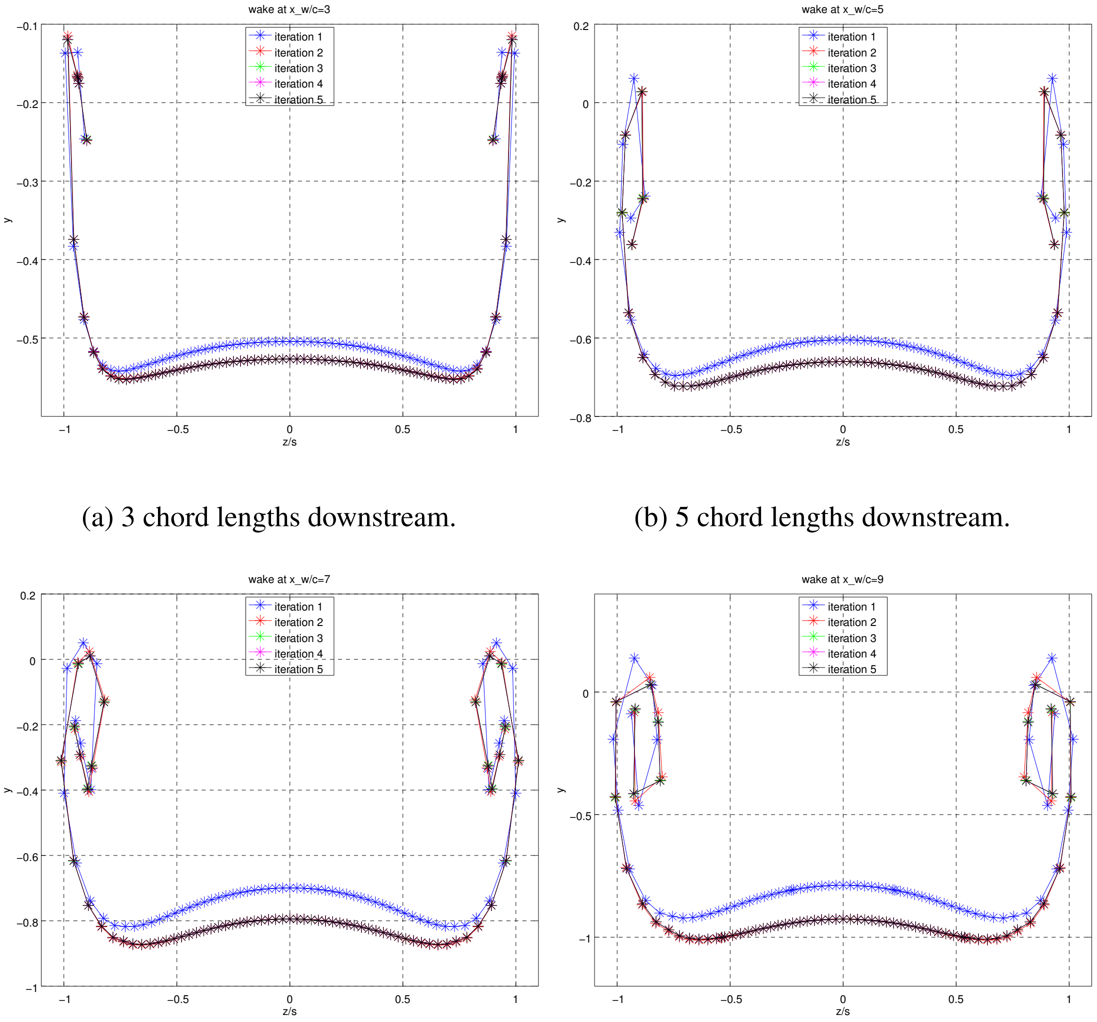
A further analysis presented in Figure 4.16 is carried out to assess the number of iterations needed to reach convergence of the wake geometry. The simulation used for such investigation accounts for 1885 wake degrees of freedom and 1547 body degrees of freedom. The plots in the Figure depict vertical cuts of the wake at different downstream locations past the wing.

The plots show a rather clear convergence of the wake to an asymptotic solution. Aside from the curve associated with the first iteration, the other tend to cluster around a final solution. This is particularly evident in the center wing section, but is quite appreciable also in correspondence with the wake roll up regions due to the tip vortex presence. So, even for 3D simulations, the wake relaxation algorithm proposed seems to be effective in convergence to a stable solution.

The analysis also suggests that the difference between the first and final iteration is higher downstream. To make sure that at the end of the wake reaches convergence we need more iterations, since the influence of the body takes more iterations to propagate.

A mesh sensitivity predicted analysis reported in Figure 4.17 has been carried out to investigate on the influence of the number of degrees of freedom at the trailing edge on the wake geometry. Using a refinement strategy that allows us to increase the number of cells only at the trailing edge, we have analysed the wake roll up dependence on the trailing edge's degrees of freedom. In particular the number of trailing edge's degrees of freedom are 33, 65, 129, corresponding to a total number of wake degrees of freedom which are respectively 1584, 3120 and 6192, while the body degrees of freedom are 1155,1421 and 1943. We remark that the body degrees of freedom increase is the result of the constraint in order to have a conformal trailing edge mesh. So, as the nodes are increased, the mesh is finer at the trailing edge, while the leading edge resolution is not modified.

The plots show that the central part of the wake is well captured even with a coarser mesh since at all sections there is not a significant wake geometry difference between

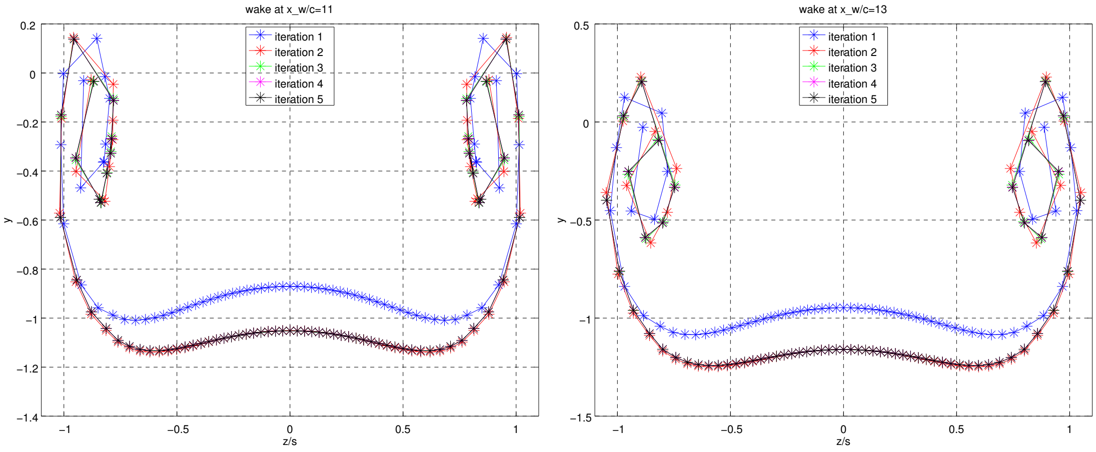


(a) 3 chord lengths downstream.

(b) 5 chord lengths downstream.

(c) 7 chord lengths downstream.

(d) 9 chord lengths downstream.

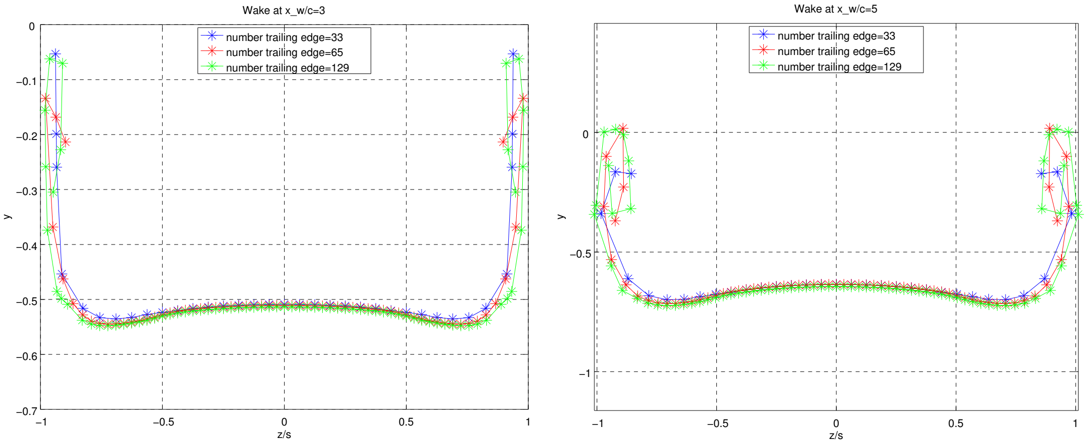


(e) 11 chord lengths downstream.

(f) 13 chord lengths downstream.

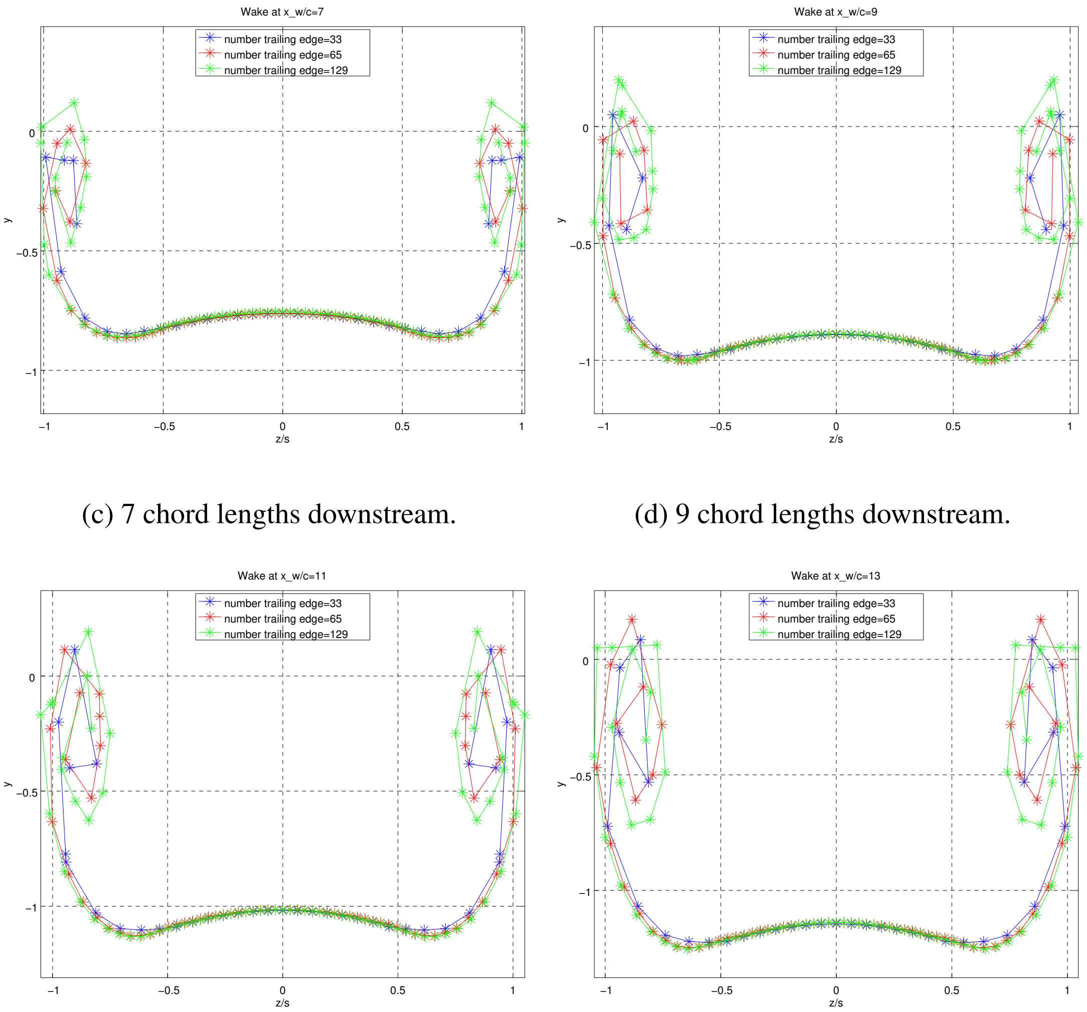
Figure 4.16: Convergence analysis for the wake past a NACA 0015 rectangular wing with AR=6.58 at an angle of attack $\alpha = 12°$. The plots show vertical cuts of the wake surface on planes perpendicular to the $\mathbf{V}_\infty$ direction and located at increasing distances from the wing trailing edge at different iterations of the algorithm proposed.

the three simulations. On the contrary at the external part of the wake, finer meshes can better capture the wake roll up. In such region, a spacial convergence is not yet achieved with the finest mesh tested. Given such results, further investigations are being considered, in which the trailing edge refinements are only concentrated in the tip region, so that a spacial convergence in such region can be fully achieved.



(a) 3 chord lengths downstream.

(b) 5 chord lengths downstream.



(c) 7 chord lengths downstream.

(d) 9 chord lengths downstream.

(e) 11 chord lengths downstream.

(f) 13 chord lengths downstream.

Figure 4.17: Analysis on the trailing edge number of degrees of freedom for the wake past a NACA 0015 rectangular wing with AR=6.58 at an angle of attack $\alpha = 12°$. The blue curve refers to the simulation with 33 degrees of freedom at the trailing edge, the red one 65 and the green one 129.

4.2.2 Mesh adaptivity qualitative analysis

We have seen in section 3.4 that the mathematical and numerical set up allows for the use of adaptive refinement strategy in our algorithm. Figure 4.18 shows a qualitative comparison between the results obtained with adaptive and global refinement for a NACA 0015 rectangular wing with aspect ratio 6.58 at an angle of attack 12°. We must here remark that in absence if an analytical solution, a proper quantitative error evaluation with respect to the best grid available can only be done projecting the adaptive solutions on such grid. We have not yet employed such feature, which will be considered in the further developments. The plot depicts the c_p distribution on the central wing section ($z/s = 0$). The solutions obtained with successive steps adaptive refinement (yellow, red, green, cyan, magenta and blue lines) are compared to the solution obtained with the best available grid. Such grid is obtained by adding two global refinement cycles to one cycle of curvature refinement (black line).

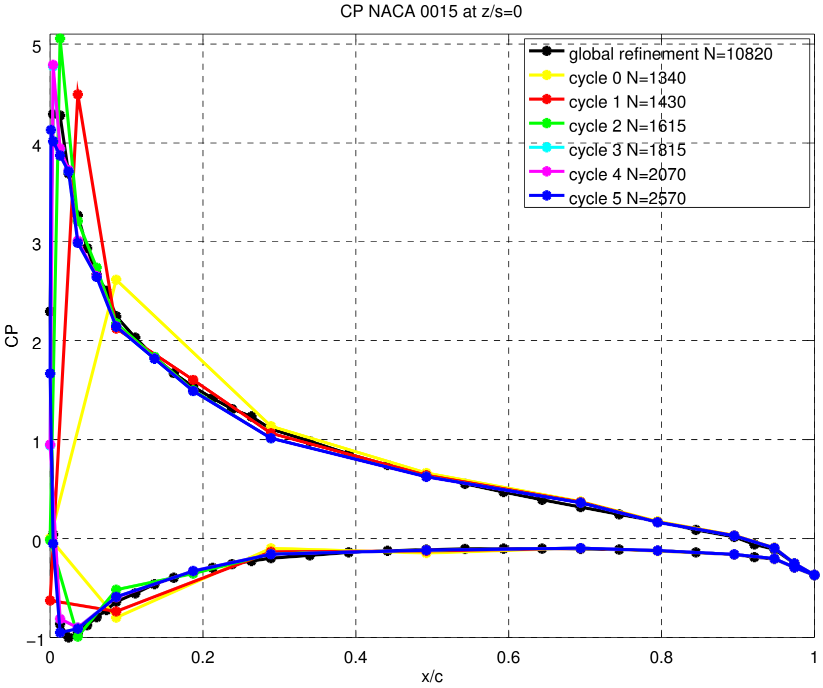
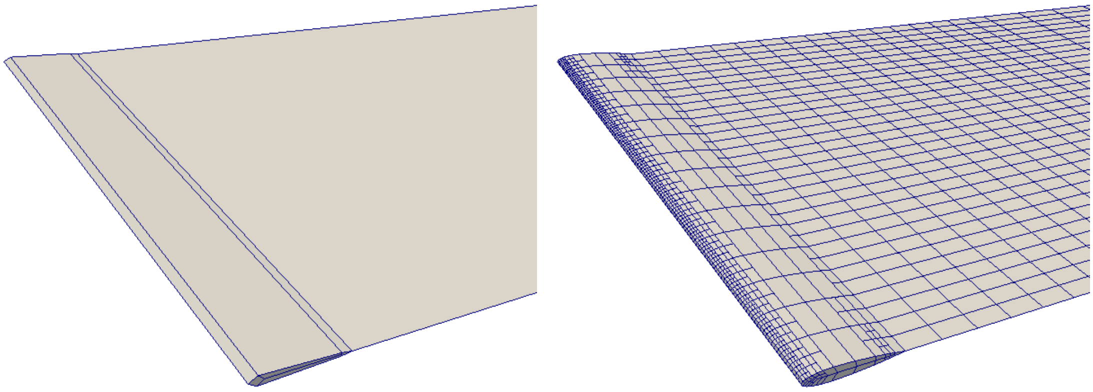


Figure 4.18: Pressure coefficient at the central section $z/s = 0$ for a NACA 0015 rectangular wing with aspect ratio 6.58 at an angle of attack 12°. The results are showed with different colours: in black for a mesh with global and curvature refinement with number of degrees of freedom on the body N=10820, in yellow for the starting mesh with N=1340, in red after the first cycle of adaptive refinement N=1430, in green after the second cycle N=1615, in cyan after the third cycle N=1815, in magenta after the fourth cycle N=2070, in blue after the fifth cycle N=2570.

We can see that as the adaptive refinements are carried out, only the number of cells in the leading edge region is incremented. The solution obtained after five adaptive refinement cycles (blue line) is very close to the one obtained with the global refinement strategy (black line). Moreover, in the central part of the profile section the same solution is obtained with less degrees of freedom. So we have obtained a problem solution with comparable accuracy, at a lesser computational cost with respect to the reference solution. In fact we used only 2570 degrees of freedom on the body (32 in the section illustrated in the plot) in comparison to the 10820 ones (66 in the section) for the global refinement strategy. Yet, in the adaptive refinement framework, obtaining the last result shown required to run the previous 5 simulations. So to obtain a fair comparison we need to compare the computational cost of the global refinement solution with the overall cost of the 6 simulations involved in the adaptive refinement cycles. We recall that the computational cost of assembling and solving the BEM linear system is $O(n^2)$ flops. Considering a 10^4 grid in the case of the global strategy such cost is proportional to flops 10^8. With the adaptive strategy overall computational cost must be obtained as the sum of the computational costs of six problems having 1340, 1430, 1615, 1815, 2070 and 2570 unknowns respectively. Assuming that the number of Newton iterations and wake cycles remains the same as in the global refinement case, it turns out that the cost of all cycles is about 10^7 flops, resulting in a clear reduction even accounting for all the refinement process. Using this strategy the gain is not only that of lowering the computational cost, but it also lies in the possibility of obtaining a solution with high resolution in some regions (such as at the leading edge) where the physical solution presents strong gradients which require finer resolution to be captured adequately. With the heavy memory requirements of a direct BEM solver, in fact, saving nodes in regions where they are not needed is very important as it allows to increase the resolution in other regions in which more degrees of freedom are needed.

4.2.3 NACA 0015

The first results we present are referred to the flow simulation past a rectangular wing with a NACA 0015 foil section with square tip and aspect ratio $AR = 6.58$ (Figure 4.19).



(a) Initial grid.

(b) Final grid.

Figure 4.19: The left image represents the initial mesh on the NACA 0015 wing. Such grid is the input file along with the CAD model of the wing. The right illustration depicts the grid used for the simulations. Such grid is obtained after aspect ratio reduction, curvature based and global refinement strategy cycles.

The solver input is a grid file containing the vertices and the connectivity of the initial cells (as illustrated in Figure 4.19a) along with CAD surfaces and edges on which the new vertices will be generated upon each mesh refinement. The software also reads a parameter file prescribing the tolerances and the number of cycles of refinements based on aspect ratio reduction, on the curvature and on global refinement. The resulting grid with maximum aspect ratio of 2.5, one curvature cycle and one global refinement cycle is displayed in Figure 4.19b. As expected a finer mesh resolution is obtained at the leading edge, where the surface curvature is higher. We remark that in presence of sharp edges (such as the trailing edge or the wing tips) the refinement based on the curvature uses the curvature of the curve describing the edge, not the surface curvature that is infinite in presence of a sharp edge.

The mesh illustrated in Figure 4.19b has been used to carry out a simulation of the potential flow past a NACA 0015 rectangular wing. Figure 4.20 depicts contours of pressure coefficient on the surface of the wing.

The Figure confirms that the pressure coefficient value is close to 1 at the stagnation point whereas it becomes negative on the top as the fluid increases its flow velocity to turn around the obstacle represented by the wing. Moreover, the airfoil section maximum pressure coefficient decreases near the tips. This phenomenon is in line

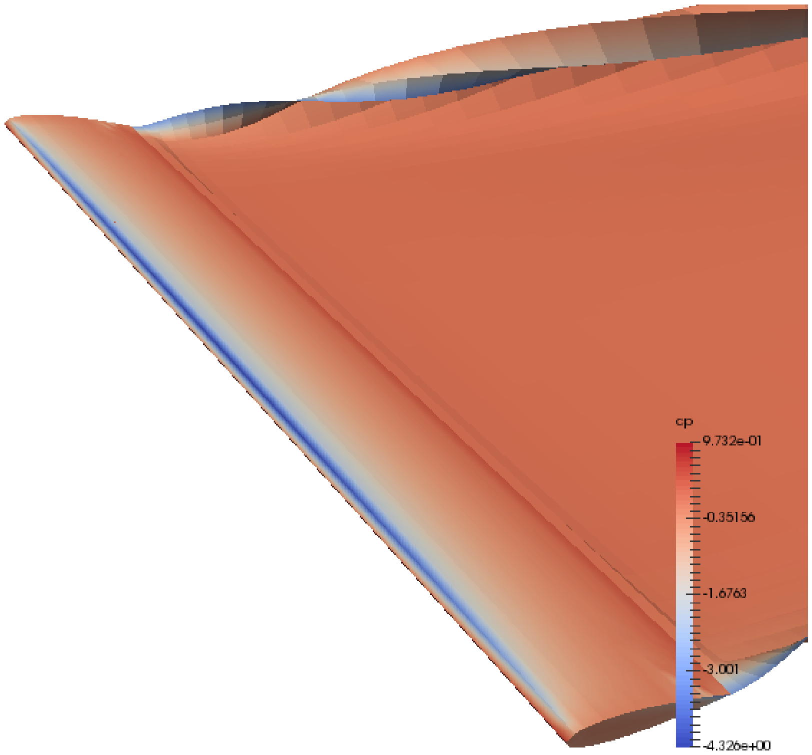
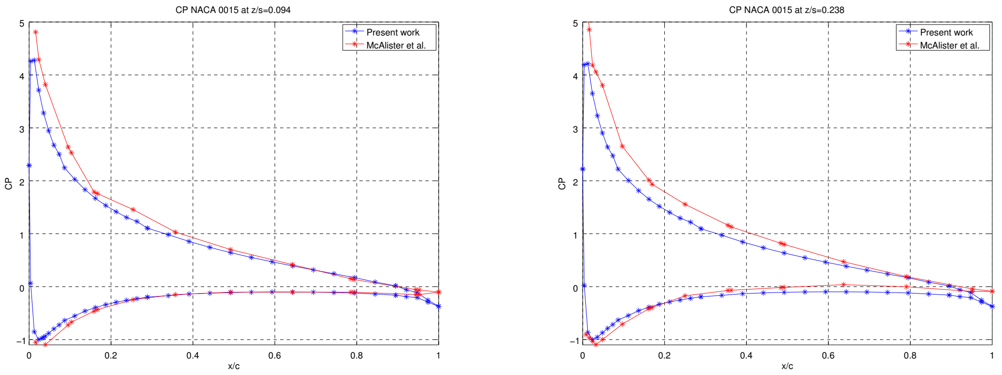


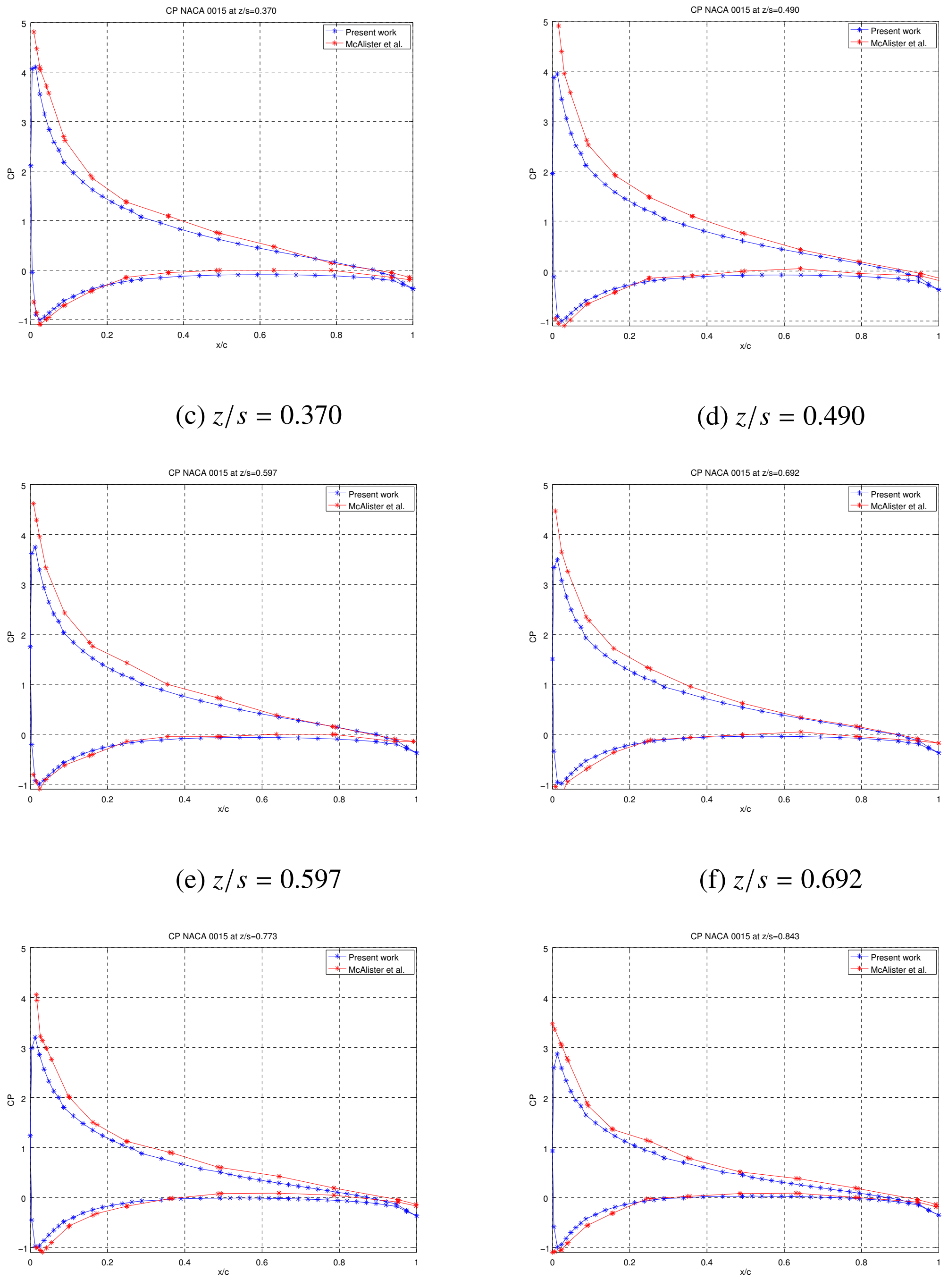
Figure 4.20: Contours of pressure coefficient on the surface of a NACA 0015 rectangular wing with AR=6.58 at an angle of attack $\alpha = 12°$.

with experimental data, since at the wing tips the flow can turn around the body from the sides with much less acceleration. This effect is well captured by our model of potential flow, although the results quality of the predictions at the tip sections drop due to the presence of viscous effect due to the tip vortex detachment from the sharp edges of the wing.



(a) $z/s = 0.094$

(b) $z/s = 0.238$



(c) $z/s = 0.370$

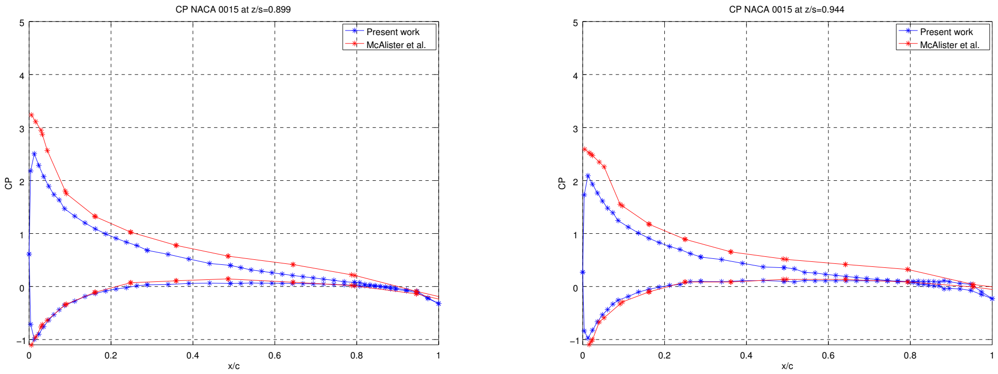
(d) $z/s = 0.490$

(e) $z/s = 0.597$

(f) $z/s = 0.692$

(g) $z/s = 0.773$

(h) $z/s = 0.843$



(i) $z/s = 0.899$

(j) $z/s = 0.944$

Figure 4.21: Comparison between numerical predictions (blue line) and experimental data referred to a NACA 0015 rectangular wing with AR=6.58 at an angle of attack $\alpha = 12°$ in [44] (red curve) at different z/s distances from the center of the wing.

In Figure 4.21 we slice the wing at different z/s distances from the center of the wing and we show the trend of pressure coefficient as a function of x/c in comparison to experimental data in [44].

The behaviour of the c_p is captured quite well by the potential flow model. However, we can notice that the present model results (blue lines) and experimental ones (red lines) by McAlister et al. in [44] differ in a rather consistent way. Such rather uniform shift of the pressure coefficient curve has been also observed in the comparison of the two dimensional simulations for the same profile with the McAlister experimental data. As already said for the 2D case, the authors of such experimental campaign seem to suggest that the reduced wind tunnel dimension makes the comparison to numerical simulations of wings in free stream rather inaccurate. Further investigations are being considered in this regard.

Figure 4.22 displays the distribution of the velocity potential on the wing and its finite jump on the wake. We can see that the condition $[\phi]_{\partial\Gamma_w} = 0$ is satisfied and that the most important contribution of the potential jump comes from the central part of the wake where its maximum value is located. On the side edges of the wake sheet we can also clearly see the typical roll ups due to the presence of tip vortices.

The distribution of the potential jump on the wake can be also found in Figure 4.23 which depicts the section circulation as a function of the normalised spanwise coordinate z/s on the wing. We recall that in potential flows the circulation is equivalent to the potential jump at the trailing edge, so the plot represents actually the potential jump. This trend fading at the wing tips is what we expected from the theory.

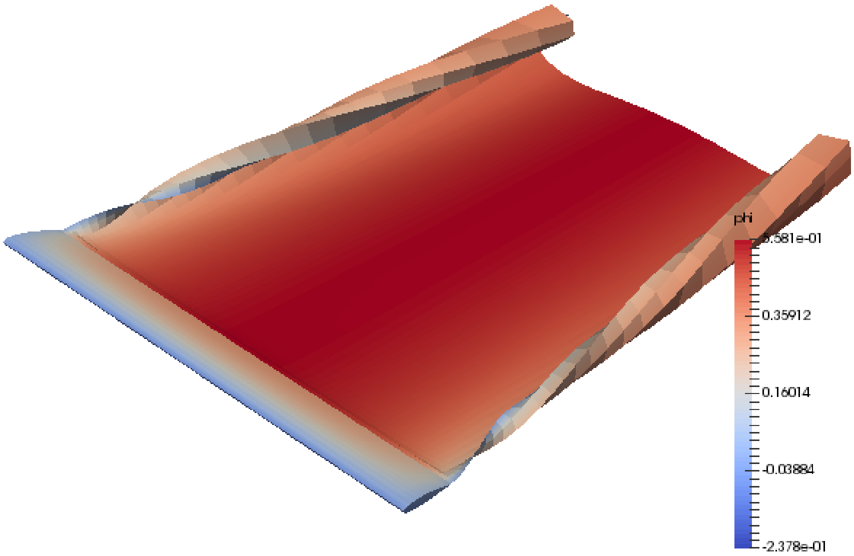


Figure 4.22: A three dimensional representation of the wake past a NACA 0015 rectangular wing with AR=6.58 at an angle of attack $\alpha = 12°$. The visualization includes the wing surface. Wake surface roll ups due to the presence of tip vortices are clearly visible on the side edges of the wake sheet. The wing is coloured according to contours of velocity potential ϕ, while the wake is coloured according to potential jump $[\phi]$.

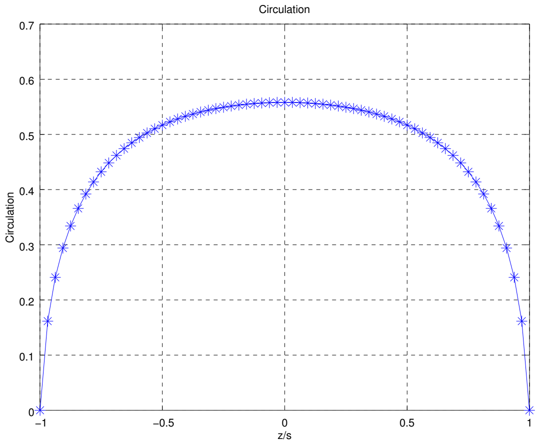


Figure 4.23: Spanwise distribution of the section circulation on the surface of the NACA 0015 rectangular wing with AR=6.58 at an angle of attach $\alpha = 12°$.

4.2.4 Tapered wing

In order to show the ability of the method in analysing different geometrical configurations, we consider now a tapered wing with a NACA 0015 profile. The chord length is varied linearly to obtain a taper ratio 1/3 and the straight line along which the airfoil section is swept is inclined of the sweep angle $\beta = 60°$ with respect to the symmetry plane. A scheme of the wing geometry is presented in Figure 4.24. The tapered wing has been prepared with such geometrical specifications to match the numerical experiments reported in [15] The numerical results are compared with similar results presented by Morino et al. in [15]. The computational grid has been automatically refined on the CAD surface of the wing.

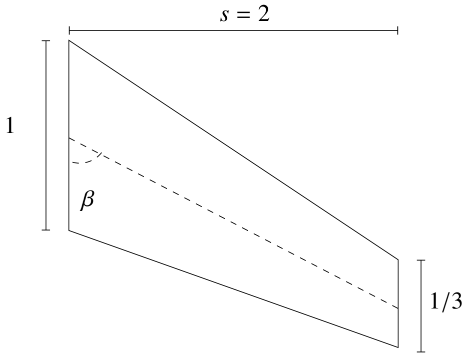


Figure 4.24: The geometry of the swept and tapered wing NACA 0015 considered by Morino et al. [15]. The wing has a 1:3 taper ratio and a positive sweep angle $\beta = 60°$.

The resulting computational mesh based on curvature adaptive refinements strategy is shown in Figure 4.25. We can see that the grid is finer at the leading edge where the curvature is high. In this specific case, a specific refinement cycle in the trailing edge region has been added. The grid also features coarser cells in the centre of the body, where the surface is flatter. On the wake, the wake roll ups at the side edges of the converged grid are clearly visible.

Figure 4.26 portrays contours of pressure coefficient over the tapered wing at an angle of attack of $\alpha = 5°$.

The maximum value of the pressure coefficient is assumed exactly in the central section. Such maximum value is clearly lower than the theoretical value 1. This is due to the fact that just as the case of the trailing edge, the BEM library used in this work, treats the sharp edges assigning two separate sets of degrees of freedom for each side of the edge. Since there is a sharp connection between the two parts of the wing, there are two different normals and two non null velocities on the two sides of the central section. Thus, there is no stagnation point. This is also the case for the leading edge on all other sections of the wing. In fact these types of swept wings have the peculiarity of reducing the pressure coefficient at the leading edge which is not in fact perpendicular to $\mathbf{V}_\infty$. The maximum sectional value of the pressure coefficient in fact is related to the sweep angle β.

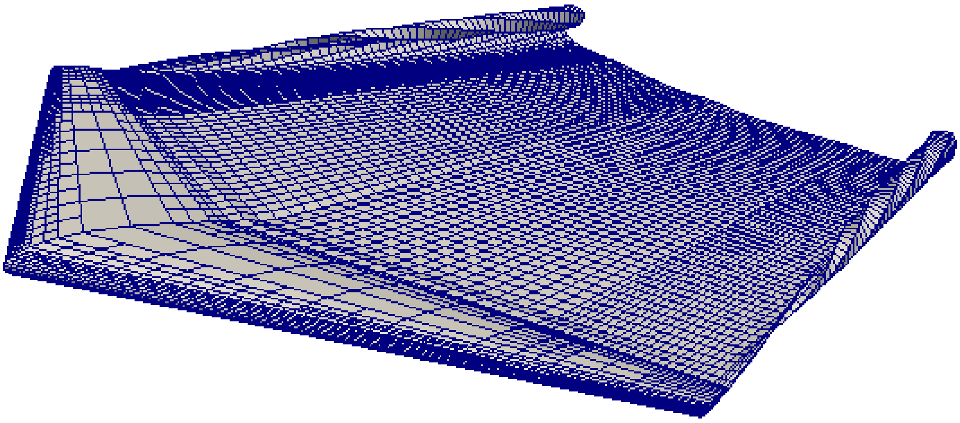


Figure 4.25: A visualization of the computational grid for the NACA 0015 tapered wing with the geometrical specifications reported in [15] at an angle of attack $\alpha = 5°$ test case. The picture depicts both the non conformal grid on the wing surface and the converged wake sheet.

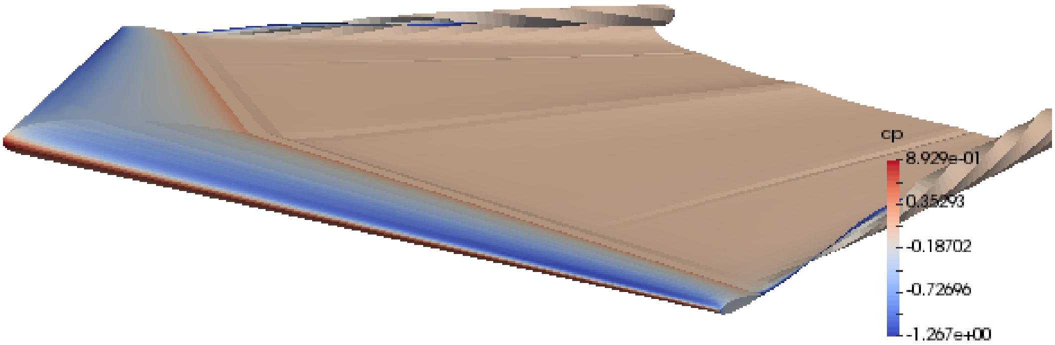


Figure 4.26: Contours of pressure coefficient on the surface of a tapered wing NACA 0015 at an angle of $\alpha = 5°$.

In this test case we can see tip effects in the lateral parts of the wing, but also, at a lower extent, in the central part since the connection between the two parts is, as said, not smooth. In such regions the pressure coefficient has a different distribution with respect to that observed in the central region of each half of the wing. This effect of the non sharp connection between the different surfaces, also has some effect on the wake shape, which bends upwards in the symmetry plane region.

Figure 4.27 presents a comparison between the present model results and those obtained by Morino et al. [15] in this wing tapered test case. The plot refers to the section located at $z/s = 0.549$, reporting the c_p for the flow past the tapered wing at an angle of attack $\alpha = 5°$.

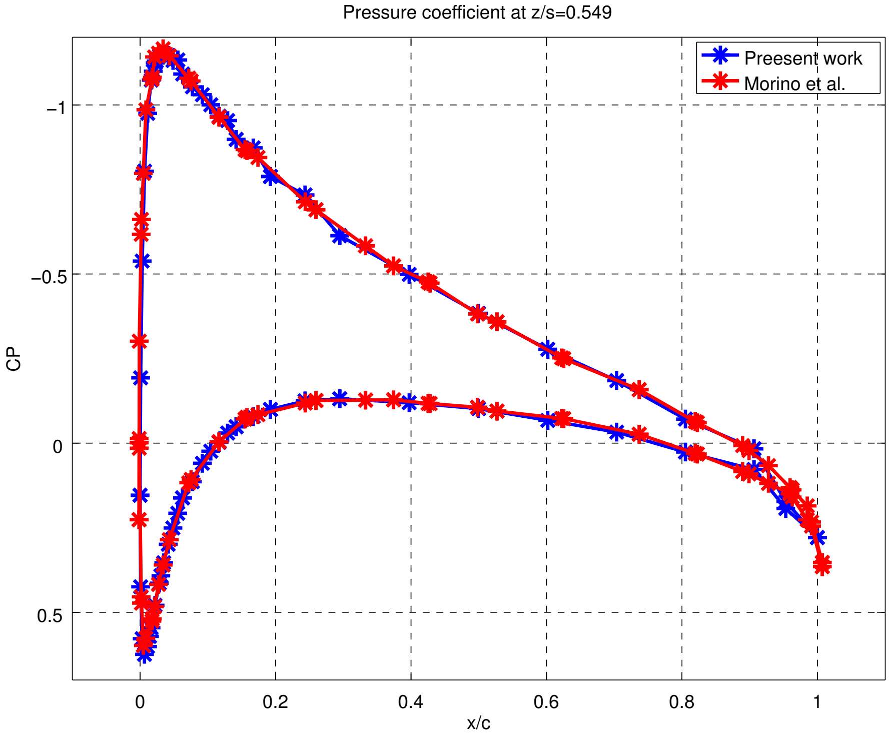


Figure 4.27: Pressure coefficient comparison for the flow past the NACA 0015 tapered wing at an angle of attack $\alpha = 5°$. The plot represents the c_p on the upper and lower portion of a longitudinal wing section locate at a $z/s = 0.548$ fraction of the wing span. The blue line represents the prediction obtained in this work, while the red line refers to data presented by Morino et al. in [15].

We can see that the results obtained in the present work (blue line) fit well the numerical data obtained by Morino et al. (red line). The only appreciable difference is a modest deviation on the trailing edge, which might be due to an insufficient refinement of the mesh in that region.
As already mentioned, the maximum value of the pressure coefficient in this section is not 1 but it is about 0.65, because of the shape of the tapered wing.

4.2.5 Double NACA

The capability of the proposed method to treat multiple wing geometries is tested by considering a double rectangular wing with NACA 0015 foil section and square tip

(Figure 4.28). This analysis is done sloping the first profile of an angle of 10° and with an angle of attack of 0°.

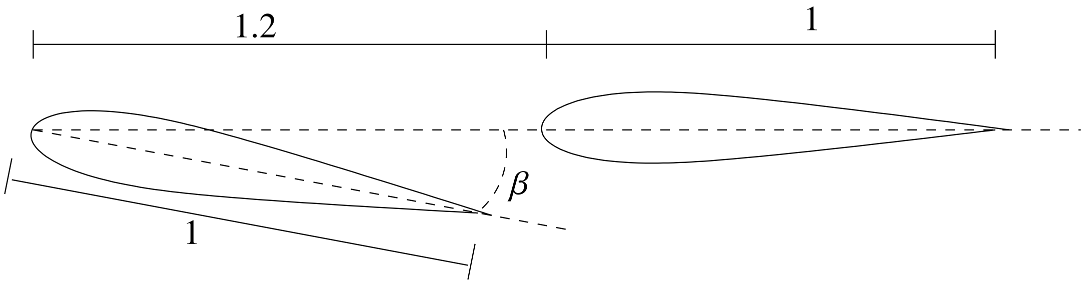


Figure 4.28: The geometry of the double NACA 0015 wing with AR=6.58. The two profile are identical, but the first one is sloped by $\beta = 10°$.

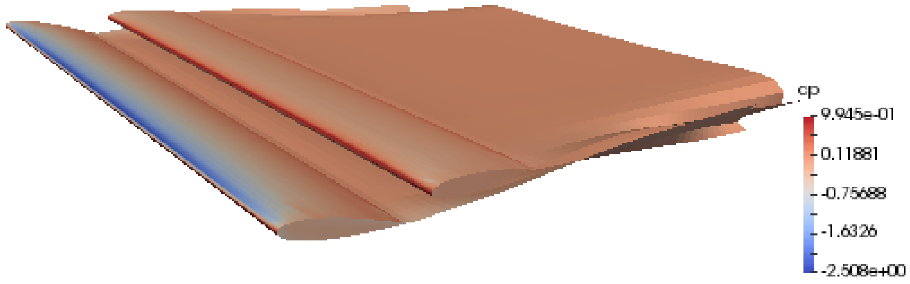
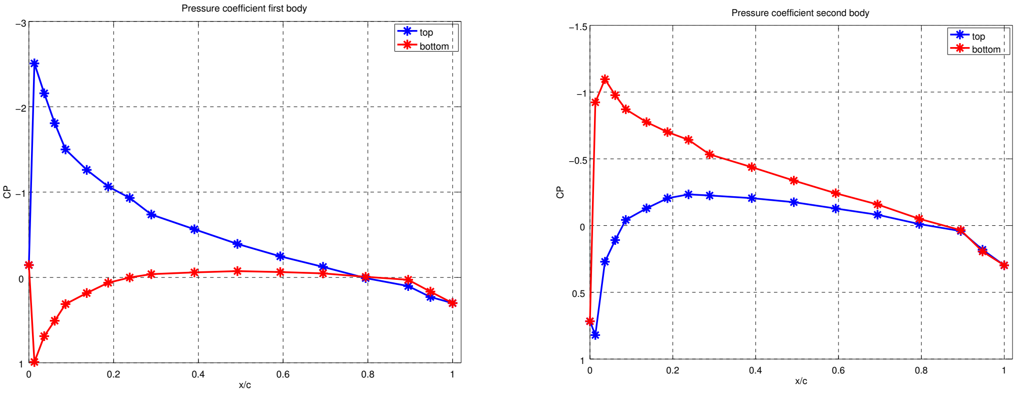


Figure 4.29: Contours of pressure coefficient on the surface of the double NACA 0015 at an angle of attack of $\alpha = 0°$.

In Figure 4.29 we present the contours of pressure coefficient on the surfaces of the two bodies at null angle of attack.
We can see more clearly in Figure 4.30 how the pressure coefficient profile evaluated in the central line is influenced by the close presence of the bodies.
We see that in the first body (Figure 4.30a) the trend on the leading edge is standard and the stagnation point is in the bottom of the NACA profile, while we have an influence on the trailing edge. The situation on the second body is more important: in absence of the upwind body, the pressure coefficient would be symmetric because the angle of attack is 0°, but now it is not. We see in Figure 4.30b that the stagnation point is at the top of the body. The presence of the first body creates a downwash effect, changing the effective angle of attack on the second wing. This fact influences also the roll up of the two wakes, in fact the first body's wake roll up is in the upper direction while



(a) Pressure coefficient first body.

(b) Pressure coefficient second body.

Figure 4.30: Pressure coefficient contours for the flow past the double NACA 0015 at an angle of attack $\alpha = 0°$. The plot represents the c_p on the upper (blue line) and lower (red line) portion of a longitudinal wing section located at a $z/s = 0$ fraction of the wing span. On the left there is the pressure coefficient of the first body, while on the right we have the second body's one.

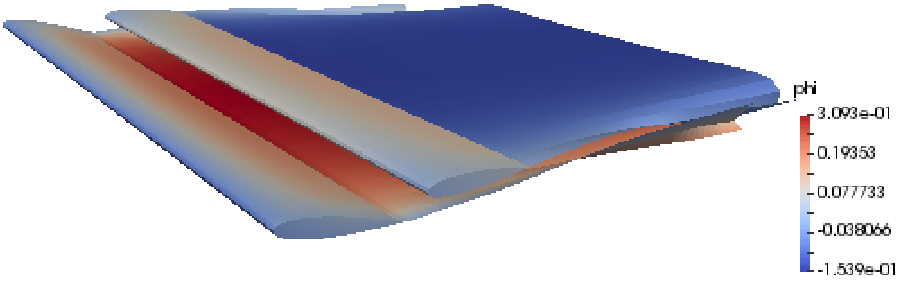


Figure 4.31: Contours of velocity potential ϕ on the bodies and potential jump $[\phi]$ on the wakes on the surface of the double NACA 0015 at an angle of attack of $\alpha = 0°$.

the wake of the second body rolls up in the other direction.
The potential jump on the wake, and therefore the circulation, is positive for the first body, while negative for the second one as shown in Figure 4.31. In both cases, we see that the condition of null potential jump at the edges of the wake is satisfied.

4.2.6 Nacelle

Finally we consider a different geometrical configuration where we analyse the potential flow around a nacelle.

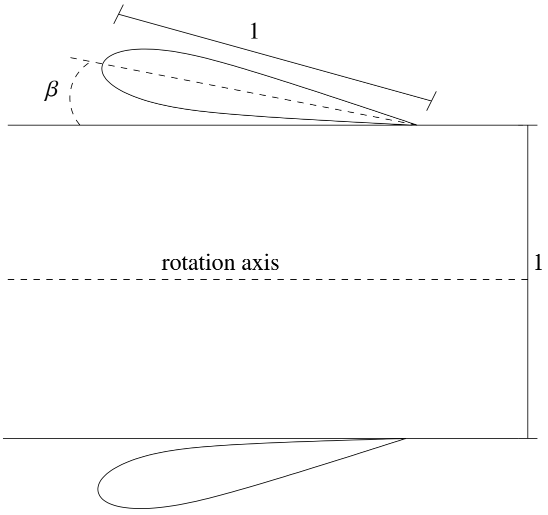


Figure 4.32: The geometry of the nacelle with NACA profile sloped by an angle of $\beta = 5°$ and distant from the rotation axis of 0.5.

This geometry is obtained by rotating a NACA profile around an axis (see Figure 4.32). This domain has an axial symmetry, so the pressure coefficient, when the angle of attack is null, has the same trend in each section of the body, whereas, when the angle of attack is not null, we have different results in the sections.

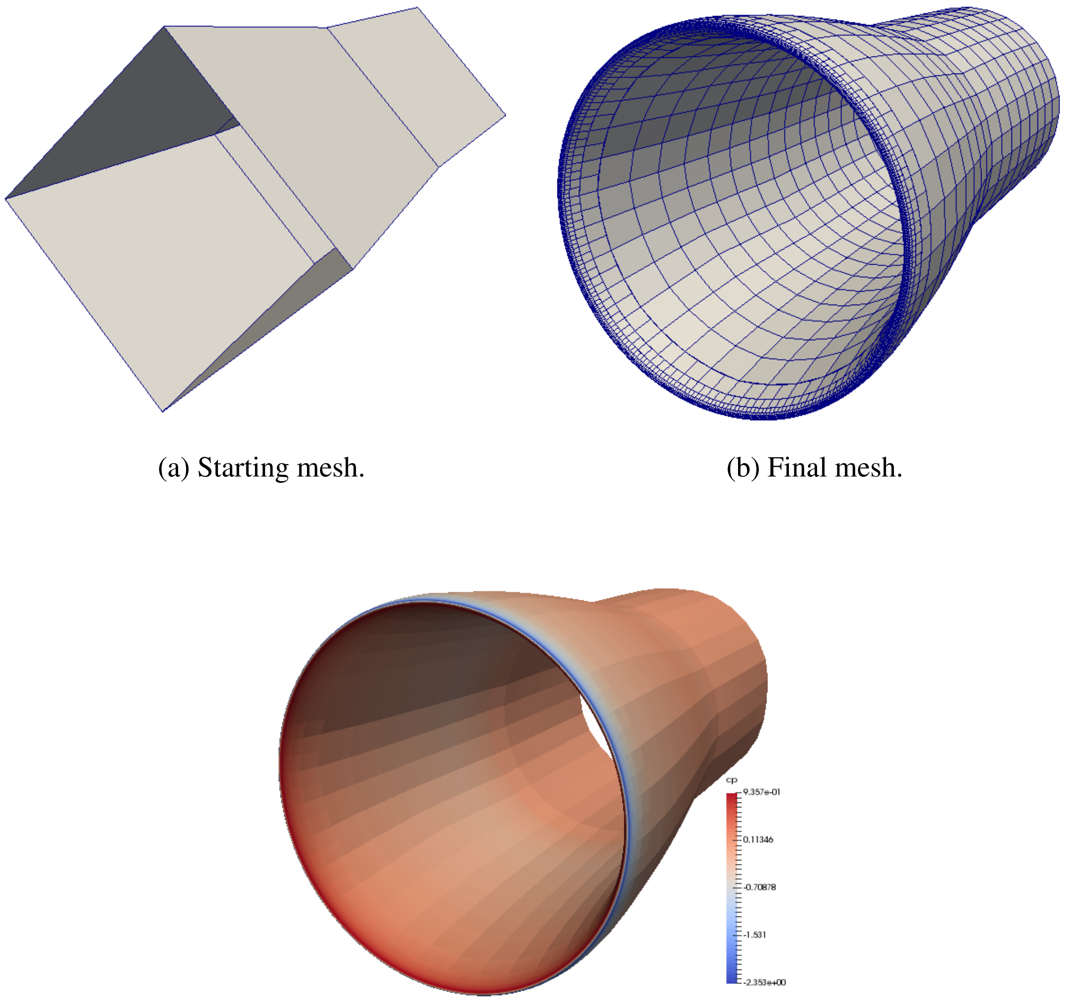
In Figures 4.33 and 4.34 we want to show how the mesh over two nacelles, obtained by the rotation of NACA 0006 and NACA 0015 profiles, is generated and some results on the pressure coefficient. Figures 4.33a and 4.34a show the input grids given to the algorithm, while Figures 4.33b and 4.34b present the resulting meshes.

The first profile is slimmer and in order to obtain pressure coefficient at the stagnation point close to 1 the mesh needs to be refined a lot in that region. Using a refinement based on the curvature, the mesh is sharp on the leading edge, but not enough to obtain pressure coefficient 1 (Figure 4.33c).

The second profile is thicker, so it is needed less refinement based on the curvature in order to capture the stagnation point (Figure 4.34c).

In both profiles we see how the flow accelerate either outside or inside the profile. This is due to the fact that in the first case the flow has to turn around an obstacle, in the second one the flow passes through a constriction and it is explainable by the Venturi effect.

In Figure 4.34d we consider the NACA 0015 nacelle with an angle of 5°. The result is that we continue to catch the stagnation point, the pressure coefficient is no more axial symmetric and the wake shape is influenced by this rotation. In particular we can see that the pressure coefficient is smaller in the top outer region in Figure 4.34d than in



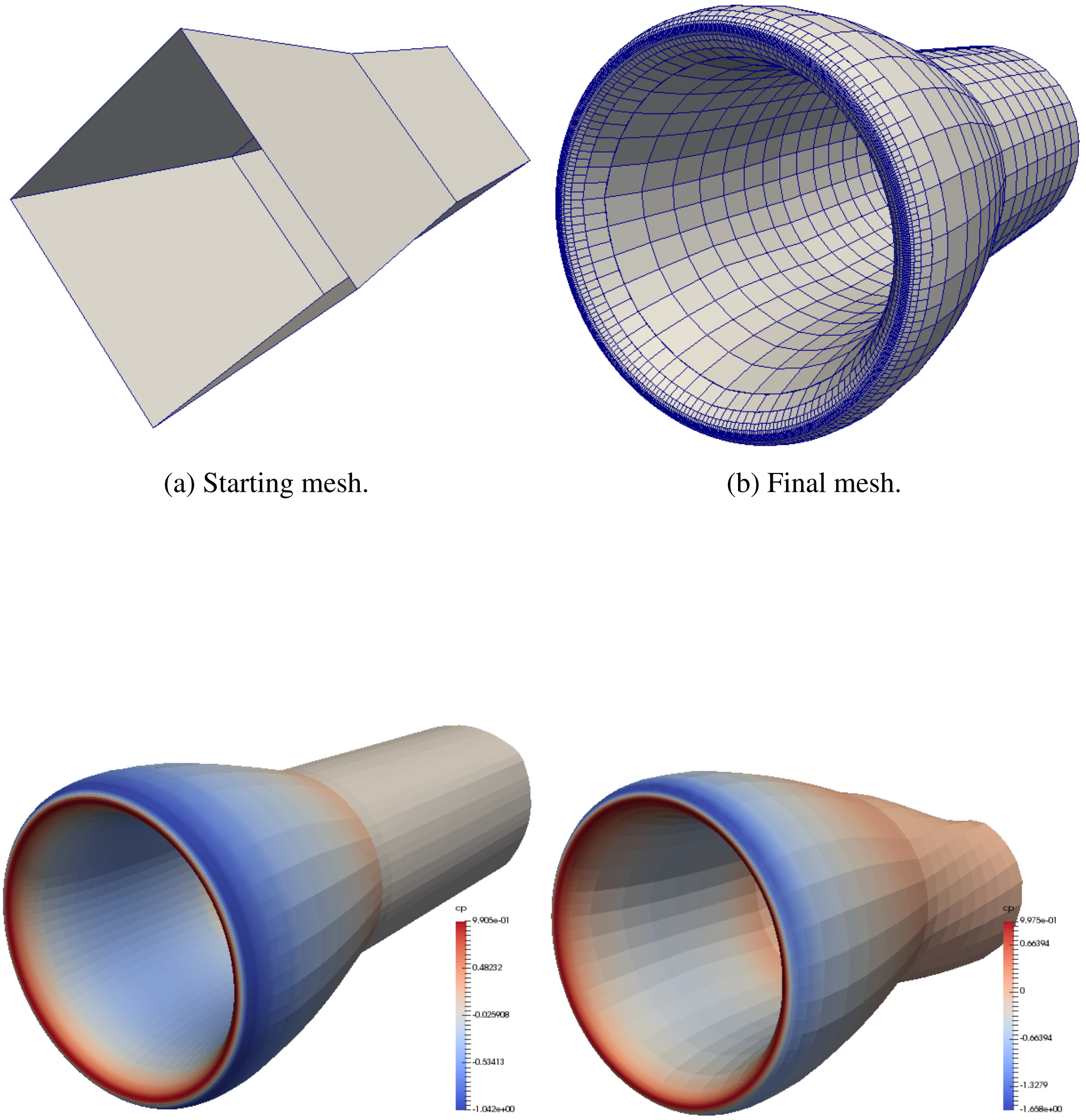
(a) Starting mesh.

(b) Final mesh.

(c) Pressure coefficient contours

Figure 4.33: Results over nacelle NACA 0006. The first figure presentsthe initial mesh, that is the input file we give to the algorithm. The second one depicts the final mesh obtained using a refinement strategy based on the curvature and projecting the new vertices of the cells on the CAD surfaces and edges. In the third one it is shown the pressure coeffcient contours for the flow past the nacelle NACA 0006 at an angle of attack $\alpha = 0^\circ$

Figure 4.34c according to what we expected. The pressure coefficient increases as we go at the bottom of the nacelle. Since we do not have edges effect, there is no wake roll-up in all nacelle simulations.



(a) Starting mesh.

(b) Final mesh.

(c) Pressure coefficient nacelle NACA 0015 $\alpha = 0°$.

(d) Pressure coefficient nacelle NACA 0015 $\alpha = 5°$.

Figure 4.34: Results over nacelle NACA 0015. The first figure presentsthe initial mesh, that is the input file we give to the algorithm. The second one depicts the final mesh obtained using a refinement strategy based on the curvature and projecting the new vertices of the cells on the CAD surfaces and edges. In the third and in the fourth figure it is shown the pressure coeffcient contours for the flow past the nacelle NACA 0015 at an angle of attack $\alpha = 0°$ and $\alpha = 5°$

Conclusions and future developments

In this work we have seen how to deal with lifting problems in 2D and 3D simulation using a potential flow model derived from an existing Boundary Element library that was able to treat only non lifting problems. In particular we have implemented a new way to impose the Kutta condition in terms of pressure, solving a nonlinear system of equation. We also see how to treat the wake, since its position is an unknown of the problem. We see how it is not easy to deal with hypersingular terms, so it is useful to change the formula for the computation of the velocity. A key characteristic of the proposed method is related to the possibility of generating a grid over a CAD surface by automatic mesh adaptivity and thus accelerating the global simulation process.

Part of this work has been devoted to the validation of the model developed. Several experimental and numerical benchmark test cases available in the literature have been considered and reproduced. The results confirm that the model presented in this work well reproduces the potential flow of a fluid past airfoils and wings.

Several further improvements of our algorithm are possible, in order to extend the range of possible engineering problems that can be solved.

The first extension may be the solution of time-dependent problems of unsteady flows or moving domain. In this case the boundary integral equation will be the same, but the constraint on the potential jump on the wake (1.35) and the Bernoulli equation (1.5) will be different, because they have to consider the term $\frac{\partial \phi}{\partial t}$.

The second possible improvement may be the solution of potential flow around thin bodies. In this case the governing equation won't take into account the potential ϕ and its normal derivative $\frac{\partial \phi}{\partial n}$, but their jump $[\phi]$ and $\left[\frac{\partial \phi}{\partial n}\right]$. The trailing edge condition will be slightly different, because we will have to write it in term of potential jump. Thus would allow to consider other families of engineering problems such as sail aerodynamic analysis.

The third aim is to couple our program with another one using the same library that solves problems of ship-wave interaction [8]. The contribution that can come from this work is to simulate the flow around the drift and the rudder blade or even to the foils of modern sailing boats.

Our intent in this work was focused on the computational performance of the code, however there exist some techniques that can speed up this algorithm. Such techniques are the parallelization of the code, the fast multipole method and high order elements [1].

Bibliography

[1] Giuliani N., Mola A., and Heltai L. π-`BEM`: A flexible parallel implementation for adaptive, geometry aware, and high order boundary element methods. *Advances in Engineering Software 121:39-58*, 2018.

[2] Bangerth W., Davydov D., Heister T., Heltai L., Kanschat G., Kronbichler M., Maier M., Turcksin B., and Wells D. The `deal.II` library, version 8.4. *Journal of Numerical Mathematics, 24(3):135-141*, 2016.

[3] Arndt D., Bangerth W., Davydov D., Heister T., Heltai L., Kronbichler M., Maier M., Pelteret J. P., Turcksin B., and Wells D. The `deal.II` library, version 8.5. *Journal of Numerical Mathematics*, 2017.

[4] On-No Tai A. Simulation of flow around deforming bodies with a boundary element method. *Degree of Doctor of Philosophy of the University of Bath Department of Mechanical Engineering*, 2009.

[5] Gaggero S. and Brizzolara S. Exact modeling of trailing vorticity in panel method for marine. *2nd International Conference on Marine Research and Transportation*, 2007.

[6] Morino L. Subsonic potential aerodynamics for complex configurations: A general theory. *AIAA Journal, 12(2):191-197*, 1974.

[7] Salmoiraghi F. Reduced-order models for potential flows past parametrized naca airfoils based on an isogeometric boundary element method. *Master Thesis Politecnico di Milano*, 2013.

[8] Mola A., Heltai L., and DeSimone A. A stable and adaptive semi-lagrangian potential model for unsteady and nonlinear ship-wave interactions. *Engineering Analysis with Boundary Elements, 37(1):128-143*, 2013.

[9] Mola A., Heltai L., and DeSimone A. Wet and dry transom stern treatment for fully nonlinear potential flow simulations of naval hydrodynamics. *Journal of Ship Research, 61(1):1-14*, 2017.

[10] Mola A., Heltai L., and DeSimone A. Nonlinear free surface potential flow simulations for hulls with a transom stern operating in dry and wet conditions. *18th International Conference on Ships and Shipping Research*, 2015.

[11] Mola A., Heltai L., and DeSimone A. Model for olympic rowing boats. *PhD thesis, Politecnico di Milano*, 2009.

[12] Mola A., Heltai L., and DeSimone A. A fully nonlinear potential model for ship hydrodynamics directly interfaced with cad data structures. *24th International Ocean and Polar Engineering Conference*, 2014.

[13] Giuliani N., Mola A., Heltai L., and Formaggia L. Engineering analysis with boundary elements fem supg stabilisation of mixed isoparametric bems : Application to linearised free surface flows. *Engineering Analysis with Boundary Elements, 59:8-22*, 2015.

[14] Raven H. C. A solution method for the nonlinear ship wave resistance problem. *PhD thesis, Technische Universiteit Delft*, 1996.

[15] Morino L. and Bernardini G. Singularities in bies for the laplace equation; joukowski trailing-edge conjecture revisited. *Engineering Analysis with Boundary Elements 25 (2001) 805-818*, 2001.

[16] Manzoni A., Salmoiraghi F., and Heltai L. Reduced basis iso- geometric methods (rb-iga) for the real-time simulation of potential flows about parametrized naca airfoils. *Computer Methods in Applied Mechanics and Engineering, 284:1147-1180*, 2015.

[17] Gaggero S., Villa D., and Brizzolara S. Rans and panel method for unsteady flow propeller analysis. *Journal of Hydrodynamics, 22(5 SUPPL.1):547-552*, 2010.

[18] Brebbia C. A., Telles J. C. F., and Wrobel L. C. *Boundary Element Techniques.* Springer, 1984.

[19] Smigaj W., Betcke T., Arridge S., Phillips J., and Schweiger M. Solving boundary integral problems with bem++. *ACM Trans- actions on Mathematical Software, 41(2):1-40*, 2015.

[20] Ueta T. Finite element and boundary element applications in quantum mechanics. *Journal of Physics A: Mathematical and General, 36(33):8913*, 2003.

[21] Henriquez V. C. and Juhl P. M. Openbem - an open source boundary element method software in acoustics. *Internoise*, 2010.

[22] Hess L. and Smith A. Calculation of non-lifting potential flow about arbitrary three-dimensional bodies. *Technical report, DTIC Document*, 1962.

[23] Lee C. S. and Kerwin J. E. A b-spline higher-order panel method applied to two dimensional lifting problem. *J. Ship Res., 47(4):290-298*, 2003.

[24] Kim G. D., Lee C.S., and Kerwin J. E. A b-spline based higher order panel method for analysis of steady flow around marine propellers. *A Ocean engng, 34(14):2045-2060*, 2007.

[25] Scorpio S. M. Fully nonlinear ship-wave computations using a multipole accelerated, desingularized method. *Report (University of Michigan. Dept. of Naval Architecture and Marine Engineering). University of Michigan*, 1997.

[26] Mousavi S. E. and Sukumar N. Generalized duffy transformation for integrating vertex singularities. *Computational Mechanics, 45(2-3):127-140*, 2010.

[27] Telles J. C. F. A self-adaptive co-ordinate transformation for efficient numerical evaluation of general boundary element integrals. *International Journal for Numerical Methods in Engineering, 24:959-973*, 1987.

[28] Lachat J.C. and Watson J. O. Effective numerical treatment of boundary integral equations: A formulation for three-dimensional elastostatics. *Int J Numer Methods Eng 10: 991-1005*, 1976.

[29] Duffy M. G. Quadrature over a pyramid or cube of integrands with a singularity at a vertex. *SIAM J Numer Anal 9(6): 1260-1262*, 1982.

[30] Bassanini P., Casciola C. M., Lancia M. R., and Piva R. Edge singularities and kutta condition in 3d aerodynamics. *Meccanica vol 34, 199-299*, 1999.

[31] Grilli S. T. and Svendsen I. A. Corner problems and global accuracy in the boundary element solution of nonlinear wave flows. *Engineering Analysis with Boundary Elements, pages 178-195*, 1990.

[32] Fiala P. and Rucz P. Nihu: An open source c++ bem library. *Advances in Engineering Software, 75:101-112*, 2014.

[33] Heltai L., Arroyo M., and DeSimone A. Nonsingular isogeometric boundary element method for stokes flows in 3d. *Computer Methods in Applied Mechanics and Engineering, 268:514-539*, 2014.

[34] Giuliani N. An hybrid boundary element method for free surface flows. *PhD thesis, Politecnico di Milano*, 2013.

[35] Morino L. Boundary integral equations in aerodynamics. *Applied Mechanics Reviews vol 40, no 8, August 1993*, 1993.

[36] Batchelor G. *An Introduction to Fluid Dynamics*. Cambridge Mathematical Library, 1973.

[37] Steinbach O. *Numerical Approximation Methods for Elliptic Boundary Value Problems*. Springer, 2008.

[38] Hartmann F. *Introduction to boundary elements: theory and applications*. Springer, 1989.

[39] Morino L. and Riva P. *Boundary Integral Methods*. Springer, 1990.

[40] Kadner H. Numerical treatment of integral equations by collocation methods. *Num. Math., 10:241-260*, 1967.

[41] Heroux M., Bartlett R., Howle V., Hoekstra R., Hu J., Kolda T., Lehoucq R., Long K., Pawlowski R., Phipps E., Salinger A., Thornquist H., Tuminaro R., Willenbring J., and Williams A. An overview of trilinos. *Sandia National Laboratories SAND2003-2927*, 2003.

[42] Dassi F., Mola A., and Si H. Curvature-adapted remeshing of cad surfaces. *Procedia Engineering 82 (2014) 253-265*, 2014.

[43] Kelly D. W., Gago O. C., Zienkiewicz O. C., and Babuska A. A posteriori error analysis and adaptive processes in the finite element method. *Int J Numer Methods Engrg 1983; 19; 1593-1619*, 1983.

[44] McAlister K. W. and Takahashi R. K. Naca 0015 wing pressure and trailing vortex measurements. *NASA Technical Paper 3151, AVSCOM Technical Report 91 -A-003*, 1991.

[45] Abbott I. H. and Von Doenhoff A. E. *Theory of Wing Sections*. Dover Publications Inc., 1959.

[46] Mason W. H., Cummings R. M., Morton S. A., and McDaniel D. R. *Applied Computational Aerodynamics*. Camridge Aerospace Series, 1998.

[47] Vigneshwaran J. Lift and drag analysis on naca 0012 and naca 4412 airfoils. *ISSUU*, 2014.

[48] Jespersen D. C., Pulliam T. H., and Childs M. L. Overflow turbulence modeling resource validation results. *NAS Technical Report: NAS-2016-01*, 2016.