Scuola IMT Alti Studi Lucca

A geometry aware arbitrary order collocation boundary element method solver for the potential flow past three dimensional lifting surfaces

Questa è la versione preprint della seguente opera:

Original
A geometry aware arbitrary order collocation boundary element method solver for the potential flow past three dimensional lifting surfaces / Cattarossi, L., Sacco Filippo Guido, D., Giuliani, N., Parolini, N., Mola, A.. - In: COMPUTERS & MATHEMATICS WITH APPLICATIONS. - ISSN 0898-1221. - 206:(2026), pp. 257-279. [10.1016/j.camwa.2026.01.021]

Availability:
This version is available at: 20.500.11771/45438

Publisher:
Elsevier

Published
DOI:10.1016/j.camwa.2026.01.021

Terms of use:

This publication is made accessible in accordance with the terms for deposit in the institutional repository, as defined by the IMT School for Advanced Studies Lucca's Open Access Policy. (https://library.imtlucca.it/sites/default/files/regolamento-policy-open-access-imtlib_0.pdf).

Si prega di consultare le pagine informative dell'editore relative alle politiche di autoarchiviazione.

(Article begins on next page)

23 November 2026

arXiv:2410.10529v1 [math.NA] 14 Oct 2024

A geometry aware arbitrary order collocation Boundary Element Method solver for the potential flow past three dimensional lifting surfaces

Luca Cattarossi[1], Filippo Sacco[4], Nicola Giuliani[2], Nicola Parolini[3], Andrea Mola[1,*]

Abstract

This work presents a numerical model for the simulation of potential flow past three dimensional lifting surfaces. The solver is based on the collocation Boundary Element Method, combined with Galerkin variational formulation of the nonlinear Kutta condition imposed at the trailing edge. A similar Galerkin variational formulation is also used for the computation of the fluid velocity at the wake collocation points, required by the relaxation algorithm which aligns the wake with the local flow. The use of such a technique, typically associated with the Finite Element Method, allows in fact for the evaluation of the solution derivatives in a way that is independent of the local grid topology. As a result of this choice, combined with the direct interface with CAD surfaces, the solver is able to use arbitrary order Lagrangian elements on automatically refined grids. Numerical results on a rectangular wing with NACA 0012 airfoil sections are presented to compare the accuracy improvements obtained by grid spatial refinement or by discretization degree increase. Finally, numerical results on rectangular and swept wings with NACA 0012 airfoil section confirm that the model is able to reproduce

*Corresponding author

[1]MUSAM Lab, Scuola IMT Alti Studi Lucca, Piazza S. Ponziano, 6 - 55100 Lucca, Italy, andrea.mola@imtlucca.it

[2]previously Scuola Internazionale Sueriore di Studi Avanzati (SISSA), now Applied Materials Inc., Santa Clara, CA 95054-3299 USA, nicola_giuliani@amat.com

[3]MOX, Dipartimento di Matematica, Politecnico di Milano, Piazza Leonardo da Vinci, 32 – 20133 Milano, Italy, nicola.parolini@polimi.it

[4]Collegio San Carlo, Corso Magenta, 71, 20123 Milano, filippogd.sacco@gmail.com

experimental data with good accuracy.

Keywords: Mathematical Modeling, Potential Flow Theory, Boundary Element Method, Lifting Surfaces, Wake Vortex Sheet

---

1. Introduction and literature review

Accurate force prediction on three dimensional lifting bodies is paramount for the design of wings, tailplanes, rudders, aircraft and naval propellers, hydrofoils and keels. The study of the forces and loads generated by such appendages through their interaction with the surrounding fluid is in fact not only necessary for reliable equilibrium and stability estimates for cruising aircraft or vessels, but also for the the proper sizing of their structure. For most of the 20th century, potential flow theory has been the backbone of fluid dynamic computations in aerospace and naval engineering. This is a result of the fact that — when used on the correct problems — potential flow solvers are capable of very accurate pressure force estimates while requiring very limited computational resources. Despite the fact that in the last decades the constant increase in computational power made affordable more accurate models, such as Navier–Stokes equations, potential flow theory is still widespread among aircraft and vessel designers. Simpler models are in fact extremely valuable in the earlier stages of design, when both faster feedback is needed and not enough data is available to feed more accurate solvers.

There are of course several different mathematical and numerical models based on potential flow theory that were developed to compute forces generated by lifting bodies. For the most part, these models differ in the way the boundary conditions of the potential flow governing equation — the Laplace equation — are handled. The lifting line [1] and lifting surface methods [2], for instance, exploit slenderness of the lifting body in one or two directions to linearize the Laplace problem boundary conditions. Due to their extremely favorable balance between accuracy and computational cost these methods still enjoy significant success in the engineering community. However, the linearized boundary conditions makes them unable to account for significantly thick surfaces, or for the presence of arbitrarily shaped streamlined bodies surrounding the lifting bodies. The latter cases — which include bulbs or fuselages — can be instead treated by models based on the exact formulation of the — Neumann — non penetration boundary condition for

the velocity potential, and are in most cases discretized via the Boundary Element Method (BEM). Since the early works of Hess et al. [3], BEM discretization of the Laplace equation for the flow velocity potential with exact non penetration boundary conditions represents a very interesting compromise, as it is able to both model complex streamlined geometries and obtain fast predictions.

This work describes the development of a numerical tool based on the BEM library π-BEM [4] for the computation of the flow past three dimensional lifting bodies. As opposed to most BEM discretizations typically used for this application [5], which make use of virtual singularity distribution formulations, π-BEM library implements an arbitrary order isoparametric discretization based on Lagrangian finite elements. To allow for the automatic generation of the grids required by high order formulations, the library is also geometry aware, as described in [6]. In the present context, this means that the simulations are directly interfaced with CAD data structures — in the form of IGES files — so that the solver is able to generate on demand new grid points on the user-prescribed surfaces. The BEM solver has been shown in [4] to have the expected convergence properties on academic test cases characterized by — regular enough — analytic solutions. The potential flow on the lifting bodies application tackled in this work poses instead several challenges, associated with the complex geometry features and lower solution regularity when compared to the academic test cases one. In addition, further complications had to be addressed to adapt the methods to the lifting body problem. Based on quasi-potential flow model [7, 8], the simulations have to account for the presence of the wake vortex sheet detaching from the wing trailing edge. In this regard, among the many different forms of the Kutta condition [9] used to determine the velocity potential jump on the wake vortex sheet, we elected to employ the nonlinear form of the condition, imposing continuous pressure across the wing trailing edge. At the numerical level, this condition involves the potential gradient evaluation at the trailing edge collocation points. In all BEM formulations based on standard Lagrangian finite elements, such potential gradients are discontinuous on all the edges of the computational grid cells. For such a reason, BEM formulations typically used in this application are based on piecewise constant discretizations, in which the solution gradient is computed making use of finite differences [5]. Clearly, such methodology results in a formulation heavily dependent on the specific topology and orientation of the computational grid used. Moreover, of course, it limits the degree of the finite elements that can be employed. A

possible interesting alternative presented in [10] is represented by the use of isogeometric formulation, in which smooth NURBS shape functions result in continuous gradients at the collocation points. However, isogeometric formulation typically requires water tight CAD surfaces in most cases composed by NURBS patches all having similar size, which is are quite difficult to obtain in industrial applications. So, to make the present isoparametric formulation independent on the kind of mesh used, and suitable for arbitrary degree Lagrangian finite elements, the potential gradients in the Kutta condition are here computed via Galerkin L_2 projection [11]. To the best of the authors' knowledge, such a procedure — borrowed from FEM — has never been coupled to the BEM solution of quasi-potential flow simulations. In addition, the geometry aware implementation allows for fully automated refinement of a coarse initial mesh on CAD surfaces that are not necessarily water tight. We refer the interested reader to [6, 12, 13] for more detail on such a methodology. Again, this represents an aspect of novelty, as no arbitrary order and CAD interfaced solver has been developed for this aeronautical and naval engineering application.

A further extension to adapt π-BEM functions to the simulation of the flow past lifting surfaces is related to the wake geometry relaxation. In the quasi-potential flow model [7, 8], the wake vortex sheet surface must be a flow surface everywhere parallel to the local velocity field. The wake relaxation algorithm used in this work to align the wake geometry to the local fluid velocity requires the computation of the the potential gradient on all the wake collocation points. Since on the wake vortex sheet the only variable available is the jump of the potential across such surface of discontinuity, the Galerkin L_2 projection method used to obtain the potential gradient on the body cannot be used in this context. As in other works [5, 14] we then resorted to the integration of the hypersingular Boundary Integral Equation (BIE) on the wake collocation points. Quadrature rules by Guiggiani et al. [15] were used to evaluate the hypersingular integrals resulting from the gradient BIE. However, such quadrature rules are based on the assumption that the potential gradient is Hölder continuous at each collocation point considered. This of course poses a problem for finite elements which feature collocation points located on cell edges. In particular, this problem affects all Lagrangian finite elements of order higher than zero. Also in the present case, it is possible to abandon Lagrangian shape functions and resort to NURBS shape functions as in [14] to build a single patch for the wake in which the potential gradient has the desired regularity. In the present work, we elected instead to write a

weak form of the hypersingular BIE and solve it based on a Galerkin method. As will be shown, at the numerical level this will only require the evaluation of the hypersingular BIE on the standard Gauss quadrature nodes. These are located inside the grid cells, where the potential derivatives are Hölder continuous. Once again, the Galerkin approach results in a formulation that is independent of the grid or finite element used, allowing for user prescribed discretization order selection at the start of each simulation.

Given the CAD data structures integration and use of Galerkin variational formulation for both the Kutta condition and wake velocity computation, the BEM solver developed is able to make use of Lagrangian shape functions of arbitrary order on structured or unstructured meshes with possible non conformal cells. As such, the present work shows that all the necessary ingredients have been put in place towards the implementation of *hp* refinement for the simulation of quasi-potential flow past lifting bodies. The derivation of adequate error estimators and the test of such a refinement strategy will be object of future work.

The content of this paper is organized as follows. Section 2 describes the physical problem and discusses the derivation of the mathematical model equations. Section 3 presents the main features of the BEM solver used, reporting details of the Galerkin formulation for the Kutta condition and wake velocity computation. Section 5 presents and discusses numerical results and comparison with experimental data. Finally, Section 6 is devoted to some concluding remarks and future perspectives.

2. Mathematical model equations

The equations of motion that describe the velocity and pressure fields $\boldsymbol{v}$ and p in a fluid region surrounding a moving body are the incompressible Navier–Stokes equations. In the present context, such equations are written in the moving domain $\Omega(t) \subset \mathbb{R}^3$ which represents the simply connected region of air and water surrounding the lifting body — or wing. Making use of the right handed orthogonal coordinate system $\widehat{xyz}$, where $\hat{\boldsymbol{i}}, \hat{\boldsymbol{j}}, \hat{\boldsymbol{k}}$ are the unit vectors aligned with axes x, y and z, respectively, the equations read:

$$\rho\left(\frac{\partial \boldsymbol{v}}{\partial t}+\boldsymbol{v}\cdot\boldsymbol{\nabla}\boldsymbol{v}\right)=\boldsymbol{\nabla}\cdot\boldsymbol{\sigma}+\boldsymbol{b} \qquad \text{in } \Omega(t) \tag{1a}$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{v}=0 \qquad \text{in } \Omega(t) \tag{1b}$$

$$\boldsymbol{v}=\boldsymbol{v}_g \qquad \text{on } \Gamma(t) \tag{1c}$$

in which ρ is the — constant — density of the fluid and $\boldsymbol{b}$ are external volume forces which in the present framework typically are related to gravity and possible inertial forces due to possible acceleration of the reference frame. The term $\boldsymbol{\sigma}=-p\boldsymbol{I}+\mu(\boldsymbol{\nabla}\boldsymbol{v}+\boldsymbol{\nabla}\boldsymbol{v}^T)$ is the stress tensor for an incompressible Newtonian fluid, $\Gamma(t):=\partial\Omega(t)$ is the boundary of the region of interest, and $\boldsymbol{n}$ is the outer normal to the boundary $\Gamma(t)$. On the boundaries of the domain, the prescribed velocity $\boldsymbol{v}_g$ is either equal to the body velocity, or to a given velocity field (for surfaces far away from body).

Equation (1a) is usually referred to as the momentum balance equation, while (1b) is referred to as the incompressibility constraint, or continuity equation.

2.1. Potential flow model

In the flow field past a slender body — such as a wing, as depicted in Figure 1 — aligned with the free stream velocity, vorticity is confined to the boundary layer region and to a thin wake following the body. In such conditions, the assumption of irrotational flow and non viscous fluid are fairly accurate in the vast majority of the computational domain.

If simply connected domains are considered, under the assumptions of irrotational flow and inviscid fluid, at each point $\boldsymbol{x}=x\hat{\boldsymbol{i}}+y\hat{\boldsymbol{j}}+z\hat{\boldsymbol{k}}$ the velocity field $\boldsymbol{v}(\boldsymbol{x},t)$ admits a representation through a scalar potential function $\Phi(\boldsymbol{x},t)$, namely

$$\boldsymbol{v}(\boldsymbol{x},t)=\boldsymbol{\nabla}\Phi(\boldsymbol{x},t) \qquad \text{in } \ \Omega(t). \tag{2}$$

In such a case, the equations of motion simplify to the unsteady Bernoulli equation and to the Laplace equation for the flow potential:

$$\frac{\partial\Phi}{\partial t}+\frac{1}{2}\left|\boldsymbol{\nabla}\Phi\right|^2+\frac{p}{\rho}+\boldsymbol{\beta}\cdot\boldsymbol{x}=C(t) \qquad \text{in } \Omega(t) \tag{3a}$$

$$\Delta\Phi=0 \qquad \text{in } \Omega(t) \tag{3b}$$

where $C(t)$ is an arbitrary function of time. In Equation (3a) we have assumed that all body forces can be expressed as $\boldsymbol{b} = -\boldsymbol{\nabla}(\boldsymbol{\beta} \cdot \boldsymbol{x})$, i.e., they are all of potential type. This is true for gravitational body forces and for inertial body forces due to uniform accelerations along fixed directions of the frame of reference.

We remark that the unknowns of System (3) Φ and p are uncoupled. Once the solution of Laplace Problem (3b) is obtained, the pressure can be in fact recovered by means of a postprocessing step through Bernoulli's Equation (3a).

2.2. Perturbation potential

As mentioned, we solve Problem (3) in the region $\Omega(t)$ surrounding the wing. In this work, we will assume that the wing is moving in a fluid at rest at velocity $\boldsymbol{V}_\infty = V_\infty \hat{\boldsymbol{i}}$. Exploiting Galilean relativity, we then solve the governing equations in a relative frame of reference moving with the wing average speed. In such an inertial frame, the x axis of the right-handed coordinate system used will be aligned with the asymptotic fluid velocity, the z is aligned with the vertical — gravity acceleration — direction, and the y axis identifies a lateral direction normal to both x and z. Figure 1 depicts the domain $\Omega(t)$, along with the asymptotic velocity vector and coordinate system. We remark that possible accelerations of the wing in the inertial frame chosen can be accounted for by specifying, at each time instant, the updated position and velocity of the wing surface points. Such a strategy would allow for the simulation of possible pitching or plunging motions of the wing under the action of fluid dynamic forces, or rotations around a shaft experienced by propeller blades. However, in all the numerical tests and simulations presented in this work the wing is not experiencing time dependent motions in the relative reference frame described. We also point out that, due to the absence of inertia in the potential flow model, whenever the boundary conditions are independent of time, a steady solution is expected. For such a reason, and to lighten the notation, we will omit the explicit time dependence of both Ω and its boundaries and the solution in the following sections.

It is convenient to split the potential Φ as the sum between a mean flow potential (due to the free stream velocity) and the so called *perturbation potential* ϕ due to the presence of the wing, namely

$$\Phi(\boldsymbol{x}) = \boldsymbol{V}_\infty \cdot \boldsymbol{x} + \phi(\boldsymbol{x}) \tag{4a}$$
$$\boldsymbol{v}(\boldsymbol{x}) = \boldsymbol{\nabla}\Phi(\boldsymbol{x}) = \boldsymbol{V}_\infty + \boldsymbol{\nabla}\phi(\boldsymbol{x}). \tag{4b}$$

The perturbation potential still satisfies Poisson equation

$$\Delta\phi = 0 \qquad \text{in } \Omega, \tag{5}$$

and in practice it is convenient to solve for ϕ, and obtain the total potential Φ from equation (4).

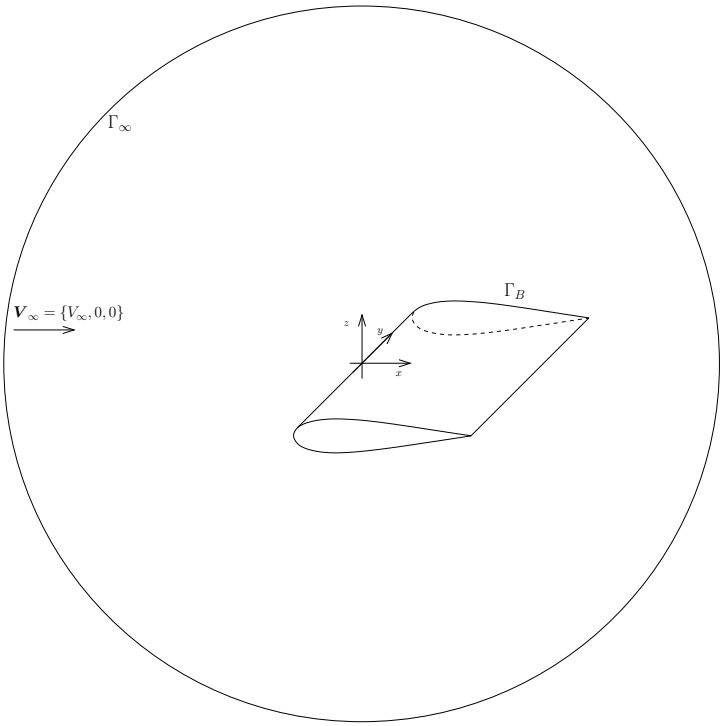

Figure 1: The computational domain

Figure 1 presents a sketch of the explicit splitting of the various parts of the boundary Γ. On the wing surface Γ_B, we apply a non penetration condition which takes the form

$$\phi_n := \boldsymbol{\nabla}\phi \cdot \boldsymbol{n} = -\boldsymbol{n} \cdot \boldsymbol{V}_\infty \qquad \text{on } \Gamma_B, \tag{6}$$

when expressed in terms of the perturbation potential and $\boldsymbol{V}_\infty$.

As the name suggests, the perturbation potential represents the flow potential component associated with the presence of the wing in the otherwise undisturbed stream. Clearly, the influence of a body having finite dimensions on the flow field is expected to fade at infinite distance from the body itself. Thus, a possible condition for the perturbation potential on the far field boundary Γ_∞ is the homogeneous Dirichlet condition

$$\phi = 0 \qquad \text{on } \Gamma_\infty. \tag{7}$$

A more accurate estimate of the expected asymptotic behavior of the perturbation potential will be obtained based on the boundary integral formulation of the problem described in Section 3.1.

2.3. Wake modeling and quasi-potential flow

The model described by Equation (5) and Boundary Conditions (6) and (7) results unfortunately unable to predict that forces are exchanged between the ideal fluid and a body moving at constant speed in it. This fact is known as D'Alembert's paradox [16]. However, it is possible to modify the potential flow model so as to extend its limits and make it able to compute accurate approximations of the fluid dynamic forces exchanged by the fluid and a streamlined body. This leads to the so called *quasi-potential* flow theory. We refer the interested reader to [7, 8] for a more detailed analysis such flow model, while in this section we will only report its main features.

In a quasi-potential model, the flow is assumed irrotational in all the flow domain except from a free vortex sheet of null thickness detaching from the trailing edge. In this way, the effects of the finite thickness vortical shear layer detaching from the wing or airfoil trailing edge are recovered, in the potential flow theory framework, by means of a wake of null thickness. The surface Γ_W representing such a wake sheet (see Figure 2) is by all means an additional boundary of the domain Ω. As such, on each point $\boldsymbol{x} \in \Gamma_W$ the limits of the potential as the surface is approached from the top side $\phi^+(\boldsymbol{x})$ or from the bottom side $\phi^-(\boldsymbol{x})$ are in principle different.

Enforcing the conservation of mass as in [17] on a infinitely thin control volume sitting across Γ_W yields

$$\frac{\partial \phi}{\partial n}^+ = \frac{\partial \phi}{\partial n}^- \qquad \text{on } \Gamma_W. \tag{8}$$

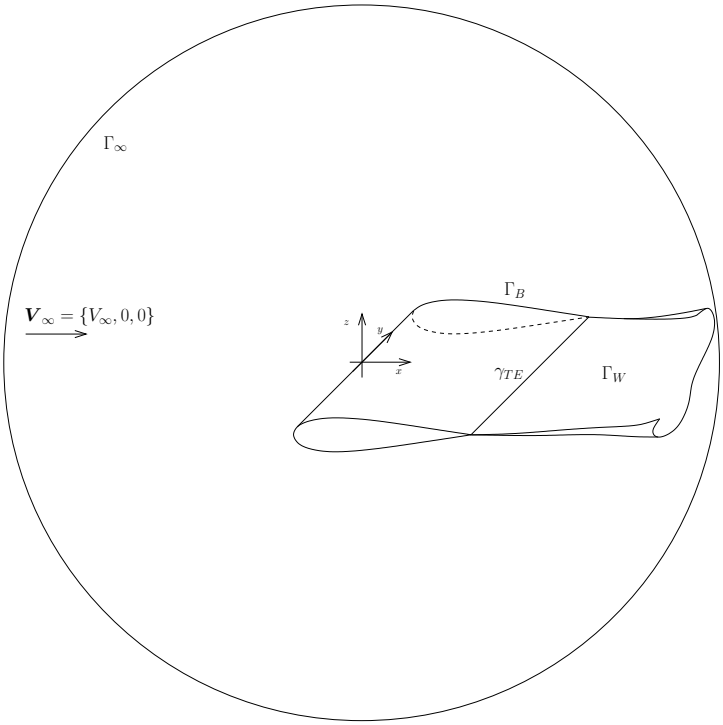

Figure 2: The computational domain including the presence of the wake sheet Γ_W detaching from the trailing edge γ_{TE}

On the same infinitely thin control volume across Γ_W, under the assumption that pressure is continuous across the wake sheet, the enforcement of the momentum conservation results in

$$\frac{D(\delta\phi)}{Dt} = \frac{\partial(\delta\phi)}{\partial t} + \boldsymbol{V}_w \cdot \boldsymbol{\nabla}(\delta\phi) = 0 \qquad \text{on} \;\; \Gamma_W, \tag{9}$$

in which

$$\delta\phi = \phi^+ - \phi^- \tag{10}$$

is the potential jump across the wake surface and $\boldsymbol{V}_w = \frac{(\boldsymbol{\nabla}\phi^+ + \boldsymbol{\nabla}\phi^-)}{2} + \boldsymbol{V}_\infty$ is the mean fluid velocity on Γ_w. Equation (9) indicates that the potential jump remains constant following a material point on the wake which moves with velocity $\boldsymbol{V}_w$. As a consequence, once the values of $\delta\phi$ on the trailing edge points are known, they can be used to readily obtain the potential jump field on the entire wake. Of course, this requires that on every point of the wake

boundary, the latter is aligned with the local wake velocity. Unfortunately, since the flow velocity is an unknown of our mathematical problem, the shape of the wake boundary Γ_w cannot be known *a priori*. So, the values of $\delta\phi$ on the trailing edge points and the effective shape of the wake are additional unknowns of the potential flow problem of the quasi-potential model governing equations. Section 2.4 will discuss the supplementary condition added to the problem to allow for the computation of the additional unknown $\delta\phi$. Section 3.5 will provide details of the fixed point wake relaxation algorithm used to compute the correct shape and position of the wake during the simulations.

2.4. Kutta condition

As mentioned, the wake vortex sheet is introduced to account for the important effects of the thin shear layer detaching from the trailing edge of an airfoil or wing. In the idealization of the quasi potential model, all the vorticity contained in such a shear layer is in fact carried within a null thickness wake sheet. As a consequence, the velocity potential becomes discontinuous across the wake.

A discontinuity in the potential is an acceptable feature in the simulations, but, of course, it should not translate into a discontinuity in the pressure field. Equation (9) is derived based on the assumption that the pressure is continuous across the wake. Thus, aligning the wake with the local velocity field ensures that pressure is continuous on the two sides of the wake. We then only need to impose pressure continuity on the points of the trailing edge, namely

$$\delta p = p^+ - p^- = 0 \qquad \text{on} \quad \gamma_{TE}, \tag{11}$$

where p^+ and p^- are the pressure values computed on the leeward and windward side of the trailing edge, respectively. The latter condition is the one added in this work to obtain the correct value of the additional trailing edge unknown $\delta\phi$. There are of course many forms available in the literature of the so called *Kutta condition* needed to find the correct $\delta\phi$ value (see among others [9]).

As pointed out in Section 2.1, the pressure value is obtained making use of Bernoulli's Equation (3a). Given the steady flow assumption used for all the simulations considered in this work, a perturbation potential based expression for pressure reads

$$p(\boldsymbol{x},t) = \frac{1}{2}\rho\left(|\boldsymbol{V}_\infty|^2 - |\boldsymbol{\nabla}\phi + \boldsymbol{V}_\infty|^2\right) - \rho g z \qquad \text{in } \Omega, \tag{12}$$

where in the derivation of Equation (12) we have assumed that the only inertial force considered is due to a gravity acceleration of module g aligned with the z axis. Kutta condition will then read

$$|\boldsymbol{\nabla}\phi^+ + \boldsymbol{V}_\infty|^2 = |\boldsymbol{\nabla}\phi^- + \boldsymbol{V}_\infty|^2 \qquad \text{on } \gamma_{TE}. \tag{13}$$

3. Numerical approximation based on the Boundary Element Method

In this work, we make use of the Boundary Element Method (BEM) for the spatial discretization of the governing Laplace equation. In the context of potential flow simulation, this is quite a common choice. We must however remark that the Laplace equation for the velocity potential can be also discretized by means of the Finite Element Method (FEM). At a first glance, it would appear that the most important advantage of BEM compared to FEM is the reduced number of unknowns associated with the codimension one grid. Unfortunately, in the practice such an advantage is offset by the presence of a dense resolution matrix in the discretized algebraic system. Yet, there are other advantages of BEM that made us favor it over FEM. In particular, in the context of the present physical problem, the codimension one grids required by BEM are much easier to generate, deform and manage without significant quality drop.

For the BEM discretization we use the same formalism presented in [4, 18, 19]

3.1. Boundary integral formulation

To rewrite (5) as a Boundary Integral Equation (BIE) we make use of a *fundamental solution* (or Green's function) of Laplace equation. More specifically, in this work we employ the free space Green's function

$$G(\boldsymbol{y}-\boldsymbol{x}) = \frac{1}{4\pi|\boldsymbol{y}-\boldsymbol{x}|},$$

which is the distributional solution of the following problem

$$\begin{aligned} -\Delta G(\boldsymbol{y}-\boldsymbol{x}) &= \delta(\boldsymbol{x}) \quad \text{in } \mathbb{R}^3 \\ \lim_{|\boldsymbol{y}|\to\infty} G(\boldsymbol{y}-\boldsymbol{x}) &= 0, \end{aligned}$$

where $\boldsymbol{y} \in \mathbb{R}^3$ is a generic point, and $\boldsymbol{x} \in \mathbb{R}^3$ is the center of the Dirac delta distribution $\delta(\boldsymbol{x})$. If we select $\boldsymbol{x}$ to be inside Ω, using the defining property of the Dirac delta and the second Green identity, we obtain

$$\phi(\boldsymbol{x}) = \int_\Gamma G(\boldsymbol{y}-\boldsymbol{x}) \frac{\partial \phi}{\partial n}(\boldsymbol{y}) \, \mathrm{d}s_y - \int_\Gamma \phi(\boldsymbol{y}) \boldsymbol{\nabla} G(\boldsymbol{y}-\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{y}) \, \mathrm{d}s_y, \quad \forall \boldsymbol{x} \in \Omega, \tag{14}$$

where in this case $\boldsymbol{y} \in \Gamma$ is a generic integration point on the domain boundary — as indicated by the subscript in differential $\mathrm{d}s_y$ — and $\boldsymbol{n}(\boldsymbol{y})$ is the outward unit normal vector to boundary Γ. The Green's function gradient is $\boldsymbol{\nabla} G = \frac{\boldsymbol{y}-\boldsymbol{y}}{-4\pi|\boldsymbol{y}-\boldsymbol{x}|^3}$.

From (14) we notice that if the solution and its normal derivative on the boundary Γ are known then the potential ϕ can be computed in any point of the domain. Considering the trace of (14) we can write the boundary integral form of the original problem as

$$\begin{aligned} c(\boldsymbol{x})\phi(\boldsymbol{x}) &= \int_\Gamma G(\boldsymbol{y}-\boldsymbol{x}) \frac{\partial \phi}{\partial n}(\boldsymbol{y}) \, \mathrm{d}s_y & (15) \\ &\quad - \int_\Gamma{}^{PV} \phi(\boldsymbol{y}) \boldsymbol{\nabla} G(\boldsymbol{y}-\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{y}) \, \mathrm{d}s_y \quad \forall \boldsymbol{x} \in \Gamma = \partial\Omega & (16) \end{aligned}$$

where $c(\boldsymbol{x})$ is obtained from the Cauchy Principal Value (CPV) evaluation of the integral involving the derivative of the Green's function — as indicated by the $\int^{PV}$ symbol. By a geometric perspective, the value of c represents the fraction of solid angle of the domain Ω seen from the boundary point $\boldsymbol{x}$. As we are interested in computing the solution on the surface of the lifting body, the point $\boldsymbol{x}$ is here assumed to always lie on Γ_B. We then need to introduce the boundary conditions of the Laplace problem described in Sections 2.2 and 2.3 to obtain the values of the integrals appearing in Equation (16) and formulate the discretized lifting body problem. To this end, we split the domain boundary into its non overlapping regions illustrated in Figure 2, namely $\Gamma = \Gamma_B \cup \Gamma_\infty \cup \Gamma_{W+} \cup \Gamma_{W-}$ and analyze separately the contributions of each region to the integrals in Equation (16).

Wake surface integrals. Note that Γ_{W+} and Γ_{W-}, are the top and bottom boundaries, respectively, of the potential flow region interfacing it with the vortical wake region. As discussed in Section 2.3, in the framework of the quasi-potential model the vortical region past the trailing edge is assumed

to have zero thickness, and it is represented as a vortex sheet surface. Thus Γ_{W+} and Γ_{W-} will be characterized by the same position and shape — which coincides with Γ_W — but by opposite orientation and normal vectors. Given these considerations, the contributions of the wake boundary Γ_W to the integrals appearing in Equation (16) result in

$$\int_{\Gamma_{W+}} G(\boldsymbol{y}-\boldsymbol{x})\frac{\partial \phi^+}{\partial n}(\boldsymbol{y})\,\mathrm{d}s_y + \int_{\Gamma_{W-}} G(\boldsymbol{y}-\boldsymbol{x})\frac{\partial \phi^-}{\partial n}(\boldsymbol{y})\,\mathrm{d}s_y = 0 \quad (17)$$

$$\int_{\Gamma_{W+}}^{PV} \phi^+(\boldsymbol{y})\boldsymbol{\nabla} G(\boldsymbol{y}-\boldsymbol{x})\cdot\boldsymbol{n}(\boldsymbol{y})\,\mathrm{d}s_y + \int_{\Gamma_{W-}}^{PV} \phi^-(\boldsymbol{y})\boldsymbol{\nabla} G(\boldsymbol{y}-\boldsymbol{x})\cdot\boldsymbol{n}(\boldsymbol{y})\,\mathrm{d}s_y = \int_{\Gamma_W}^{PV} \delta\phi(\boldsymbol{y})\boldsymbol{\nabla} G(\boldsymbol{y}-\boldsymbol{x})\cdot\boldsymbol{n}(\boldsymbol{y})\,\mathrm{d}s_y, \quad (18)$$

where we have used Equations (8) and (10), and the opposite sign of the normal unit vectors on the two sides of the wake surface.

Far field surface integrals. As for boundary Γ_∞, we can assume, without loss of generality, that it can be represented as the surface of a sphere with radius $R \to \infty$ (see Figure 2). Making use of spherical coordinates (r, φ, θ) a generic point $\boldsymbol{x} \in \Gamma_\infty$ can be expressed as $\boldsymbol{x} = R\sin\varphi\cos\theta\hat{\boldsymbol{i}} + R\sin\varphi\sin\theta\hat{\boldsymbol{j}} + R\cos\varphi\hat{\boldsymbol{k}}$ and the BIE integrals on Γ_∞ read

$$\int_{\Gamma_\infty} G(\boldsymbol{y}-\boldsymbol{x})\frac{\partial \phi}{\partial n}(\boldsymbol{y})\,\mathrm{d}s_y = \lim_{R\to\infty}\int_0^\pi \sin\varphi\,\mathrm{d}\varphi\int_0^{2\pi}\frac{1}{4\pi R}\frac{\partial \phi}{\partial n}(R,\varphi,\theta)R^2\,\mathrm{d}\theta \quad (19)$$

$$\int_{\Gamma_\infty} \phi(\boldsymbol{y})\boldsymbol{\nabla} G(\boldsymbol{y}-\boldsymbol{x})\cdot\boldsymbol{n}(\boldsymbol{y})\,\mathrm{d}s_y = \lim_{R\to\infty}\int_0^\pi \sin\varphi\,\mathrm{d}\varphi\int_0^{2\pi}\frac{R}{4\pi R^3}\phi(R,\varphi,\theta)R^2\,\mathrm{d}\theta,$$

where we have used the fact that as $R \to \infty$ $\boldsymbol{x}-\boldsymbol{y} \simeq \boldsymbol{x}$ and $(\boldsymbol{x}-\boldsymbol{y})\cdot\boldsymbol{n}_{\Gamma_\infty} \simeq R$. Equations (19) suggest that assuming $\phi = o(1)$ and $\frac{\partial\phi}{\partial n} = o(\frac{1}{R})$ as $R \to \infty$ results in null values for both integrals on Γ_∞. Such assumptions are by no means restrictive. Requiring that the perturbation potential is a function that fades at an infinite distance from the body generating the perturbation is in fact consistent with its very definition. In light of this, the perturbation potential normal derivative must present an asymptotic behavior proportional to the derivative of the perturbation potential, which makes the assumption $\frac{\partial\phi}{\partial n} = o(\frac{1}{R})$ also reasonable. For these reasons, in this work the contribution of Γ_∞ to the boundary integrals in Equation (16) is considered null. The discretization of such boundary will then not be necessary.

Body surface integrals. On the body boundary Γ_B, making use of Equation (6), we can write

$$\int_{\Gamma_B} G(\boldsymbol{y}-\boldsymbol{x})\frac{\partial \phi}{\partial n}(\boldsymbol{y})\,\mathrm{d}s_y = -\int_{\Gamma_B} G(\boldsymbol{y}-\boldsymbol{x})(\boldsymbol{n}(\boldsymbol{y})\cdot\boldsymbol{V}_\infty)\,\mathrm{d}s_y \tag{20}$$

The final boundary integral formulation of the problem, written for a generic point $\boldsymbol{x}\in\Gamma_B$, reads

$$\begin{aligned} c(\boldsymbol{x})\phi(\boldsymbol{x}) + \int_{\Gamma}^{PV} \phi(\boldsymbol{y})\boldsymbol{\nabla}G(\boldsymbol{y}-\boldsymbol{x})\cdot\boldsymbol{n}(\boldsymbol{y})\,\mathrm{d}s_y + \\ \int_{\Gamma_W}^{PV} \delta\phi(\boldsymbol{y})\boldsymbol{\nabla}G(\boldsymbol{y}-\boldsymbol{x})\cdot\boldsymbol{n}(\boldsymbol{y})\,\mathrm{d}s_y = -\int_{\Gamma_B} G(\boldsymbol{y}-\boldsymbol{x})(\boldsymbol{n}(\boldsymbol{y})\cdot\boldsymbol{V}_\infty)\,\mathrm{d}s_y \end{aligned} \tag{21}$$

3.2. Discretisation

The numerical discretization of (21) leads to a real-valued Boundary Element Method (BEM). The resolution of a BEM requires the discretization of the unknowns using functional spaces defined on a Lipschitz boundary. We address this problem introducing suitable discretizations of the unknown ϕ on the body surface and of the unknown $\delta\phi$ on the wake surface. Such discretizations are based on standard Lagrangian finite element spaces defined on Γ. We then use the same functional space to describe the geometry. This setting is often referred to as Isoparametric BEM.

We define the computational mesh as a quadrilateral decomposition Γ_h of the boundary Γ. We require that two cells K, K' of the mesh only intersect on common edges or vertices, and that there exists a mapping from a reference cell $\hat{K}$ to K. The Jacobian must be uniformly bounded away from zero for all cells K. To ease mesh generation, the simulation tool developed allows for the definition of a very coarse initial grid, which is then automatically refined on the user prescribed geometry of the wing up to the desired level of refinement. Following [20, 21] an interface to CAD data structures — which are the most common tool to define arbitrary geometrical descriptions [6, 22, 23] — is used to specify the desired geometry of the wing. This feature has been previously employed in hydrodynamics simulations through BEM, [11, 24, 25], and [4] presents an example of a Laplacian problem with known analytic solution solved for convergence testing purposes on the geometry of an aeronautics-like NACA wing shape. As will be shown in Section 4, such CAD-interfacing feature has also been used in this work for the proper

placing of the degrees of freedom on the CAD surface whenever a high order finite element is selected for the mapping and the solution. We point out that the latter feature is ultimately what results in the possibility for the user to effectively select arbitrary order finite elements at the start of each simulation. In fact, the generation of quality unstructured high-order meshes remains a significant obstacle in the adoption of high-order finite elements. In the framework of the methodology here presented however, only a standard low order grid is required at the start of the simulation, as possible additional high order mapping degrees of freedom are automatically generated on the CAD surfaces.

If ϕ and $\frac{\partial \phi}{\partial n}$ lie in the spaces V, defined as

$$V := \left\{ \phi \in H^{\frac{1}{2}}(\Gamma) \right\},$$

where $\Gamma = \partial\Omega$, then the integrals in Equation (21) are bounded. $H^{\frac{1}{2}}(\Gamma)$ is the space of traces on Γ of functions in $H^1(\Omega)$. We construct the discretized space V_h as conforming finite dimensional subspace of V

$$V_h := \left\{ \phi_h \in C^0(\Gamma_h) : \phi_{h|K} \in \mathcal{Q}^r(K),\, K \in \Gamma_h \right\} \equiv \text{span}\{\psi_i\}_{i=1}^{N_V} \tag{22}$$

where $\mathcal{Q}^r(K)$ is the space of polynomials of order r in each coordinate direction. In Equation (22) ψ_i denotes the shape function associated with the i-th degree of freedom of the discretized space. For additional details on the finite elements used in the BEM formulation we refer the interested reader to [26]. Following [4] we use iso-parametric discretizations based on standard Q_N Lagrangian finite elements, and by collocating the support points of the geometry patches directly on the CAD surfaces. This work reports results obtained with bi-linear elements ($r = 1$), as well as of higher order bi-quadratic ($r = 2$) or bi-cubic ($r = 3$) elements.

Since the unknowns ϕ and $\delta\phi$ are defined on complementary portions of the discretized domain, we can define a single function in the discretized space $\phi_h \in V_h$ that approximates ϕ on the body surface and $\delta\phi$ on the wake surface. The generic element of the discretized space reads

$$\phi_h(\boldsymbol{x}) = \sum_{j=1}^{N_V} \hat{\phi}_j \psi_j(\boldsymbol{x}), \tag{23}$$

where the vector $\hat{\boldsymbol{\phi}} \in \mathbb{R}^{N_V}$ contains the value of the discretized potential ϕ_h on each collocation point located on the wing surface Γ_B, and of the potential jump $\delta\phi_h$ on each collocation point located on the wake. N_V then represents the overall number of degrees of freedom of the discretized space V_h. To allow for possible discontinuities of the unknown functions across the trailing edge γ_{TE} we use a methodology inspired to the double nodes technique presented in [27]. More precisely, the use of a continuous Lagrangian Finite Element space is in principle not suited for the representation of the solution at the trailing edge. On one hand in fact, the value of the potential obtained approaching the trailing edge from the windward side of the airfoil is different from that obtained approaching the same point from the leeward side. On the other hand, the value obtained approaching the same point from the wake side must be the difference between the two airfoil side values, as stated by Equation (10). To accommodate for this, the computational mesh will be built in such a way that in correspondence with the sharp trailing edge, the degrees of freedom of the discretization space will be triple (see Figure 3). This allows for the functions of space V_h to have different trailing edge values on the cells of the airfoil leeward side, windward side, and wake. Note that the increased number of unknowns implies that additional equations must be introduced for the closure of the mathematical problem. As it will be shown in Section 3.3, such equations will be obtained making use of Equation (10) and by imposing a discretized version of the Kutta condition (13).

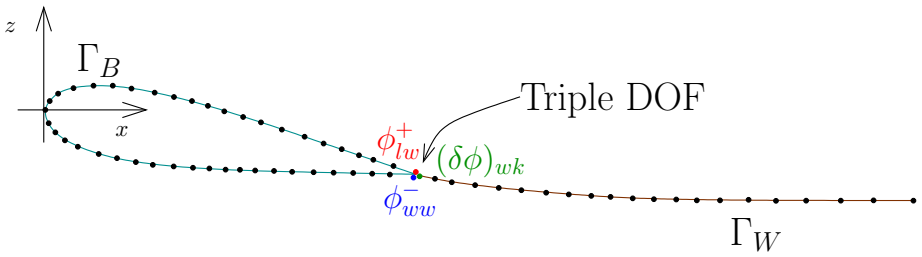

Figure 3: A two dimensional sketch of the computational domain including details of the collocation nodes, coinciding with on the degrees of freedom (DOFs) of the discretization space. The image also indicates that in correspondence with the trailing edge γ_{TE}, a triple DOF is present to allow for the space V_h functions to have different trailing edge values on the cells of the airfoil leeward side, windward side, and wake.

The collocation method is a common resolution technique for a BEM since it does not require any additional integration of (21). We refer the

interested reader to [4] for a detailed analysis of the accuracy of this setting. Collocating Equation (21) results in the following linear system

$$(C + N)\hat{\boldsymbol{\phi}} - \boldsymbol{b} = 0, \tag{24}$$

where, for all the row indices i corresponding to collocation points $\boldsymbol{x}_i \in \Gamma_B$:

- C is a diagonal matrix with diagonal entries $C_{ii} = c(\boldsymbol{x}_i)$
- $N_{ij} = \sum_{k=1}^{K} \sum_{q}^{N_q} \dfrac{\partial G}{\partial n}(\boldsymbol{x}_i - \boldsymbol{x}_q)\psi_q^j J^k(\boldsymbol{x}_q)$, where J^k is the determinant of the first fundamental form for each panel K, and $\boldsymbol{x}_q$ are the N_q quadrature nodes used to compute the integral at the numerical level;
- $b_i = \sum_{k=1}^{K} \sum_{q}^{N_q} G(\boldsymbol{x}_i - \boldsymbol{x}_q)(-\boldsymbol{V}_\infty \cdot \boldsymbol{n}(\boldsymbol{x}_q))J^k(\boldsymbol{x}_q)$;

When the collocation point lies inside the current integration cell, we make use of bidimensional quadrature formulas based on Duffy transformation as in [28] to treat singular kernel integrals; in any other case, we use standard Gauss quadrature rules.

It is important to point out that the linear system equation written on each row corresponding to a wake collocation point is not the discretized BIE, but is instead the discretized version of Equation (9). This requires the use of a structured grid in such a region, and that the grid is conformal along the trailing edge γ_{TE}. An example of structured wake grid is depicted in Figure 5. With this kind of grids it is possible to arrange the degrees of freedom of the wake in lines that sit on the pathlines detaching from the trailing edge. The wake relaxation algorithm used to align such wake degrees of freedom lines with the pathlines is described in Section 3.5. Thus, for all the row indices i corresponding to collocation points $\boldsymbol{x}_i \in \Gamma_W$:

- computing integrals for the BEM matrices is not needed on the wake, as on Γ_W Equation (9) is exploited;
- the system matrix $(C+N)$ is used to impose that the value of $\delta\phi$ at each degree of freedom on the wake is equal to that of the previous degree of freedom on the same pathline. Thus, for each line the only non null entries will be a 1 on the principal diagonal, and a -1 in correspondence with the index of the previous degree of freedom along the pathline;
- the right hand side entries b_i are null in this region.

3.3. Numerical enforcement of the nonlinear Kutta condition

System (24) has to be completed with the additional conditions on the trailing edge degrees of freedom to close the problem and obtain the desired solution. As discussed in the previous section and illustrated in Figure 3, each collocation point on γ_{TE} will correspond to three degrees of freedom in the resolution system. The equations written on the row corresponding to each of these degrees of freedom in the system will be:

- **Wing leeward side:** on the row corresponding to this degree of freedom, we write the Boundary Integral Equation (21), centered at the collocation point on the trailing edge.
- **Wing windward side:** on the row corresponding to this degree of freedom, we write the discretized Kutta condition (13), namely

$$|\boldsymbol{\nabla}\phi^{+}_{lw} + \boldsymbol{V}_{\infty}|^2 = |\boldsymbol{\nabla}\phi^{-}_{ww} + \boldsymbol{V}_{\infty}|^2, \tag{25}$$

  where the subscripts lw and ww indicate the local degree of freedom index on the airfoil leeward side and airfoil windward side, respectively.
- **Wake degree of freedom:** in this degree of freedom, the unknown quantity is the potential jump $\delta\phi$ on the wake point attached to the trailing edge. As such, the equation written is

$$(\delta\phi)_{wk} = \phi_{lw} - \phi_{ww}, \tag{26}$$

  where the subscript wk indicates the local degree of freedom index on the wake.

Introducing Equation (26) in the resolution system does not pose significant problems, as it translates in adding a line in correspondence with the wk-th degree of freedom, in which only a unit diagonal term and two negative unit off diagonal terms at columns lw and ww are present. On the other hand, the discretization of Equation (25) requires special attention, as the gradients of the potential function are not a single valued function if evaluated in correspondence with collocation points located on cells edges. We then resort to a L_2 projection strategy to obtain suitable approximations $\chi_x, \chi_y, \chi_z \in V_h$ of the components of the surface gradient $\nabla_s\phi_h$ combined

with the potential normal derivative $\left(\frac{\partial \phi}{\partial n}\right)_h = -\boldsymbol{V}_\infty \cdot \boldsymbol{n}$ and the asymptotic velocity $\boldsymbol{V}_\infty$. The weak form of the velocity x component computation reads

$$\int_\Gamma v_h \chi_x \, \mathrm{d}s = \int_\Gamma v_h \left(\nabla_s \phi_h + \left(-\boldsymbol{V}_\infty \cdot \boldsymbol{n}\right) \boldsymbol{n} + \boldsymbol{V}_\infty\right) \cdot \hat{\boldsymbol{i}} \, \mathrm{d}s \quad \forall v_h \in V_h. \tag{27}$$

From the discretization of Equation (27) we get the N dimensional sparse system

$$M \hat{\boldsymbol{\chi}}_x = \boldsymbol{b}^{\chi_x}, \tag{28}$$

where

- $M: \ m_{ij} = \int_\Gamma \psi_i \psi_j \, \mathrm{d}s$ is a sparse mass matrix;
- $\boldsymbol{b}^{\chi_x}: \ b_i^{\chi_x} = \int_\Gamma \psi_i \left(\nabla_s \phi_h - \boldsymbol{V}_\infty \cdot \boldsymbol{n} + \boldsymbol{V}_\infty\right) \cdot \hat{\boldsymbol{i}} \, \mathrm{d}s$ is the right hand side vector;
- $\hat{\boldsymbol{\chi}}_x$ is the vector containing the values of the fluid velocity on each collocation point.

Note that the value of the local discretized velocity at each integration point is computed as

$$\nabla_s \phi_h - \boldsymbol{V}_\infty \cdot \boldsymbol{n} + \boldsymbol{V}_\infty = \sum_{j=1}^{N_V} \left(\left(\nabla_s \phi\right)_j - \boldsymbol{V}_\infty \cdot \boldsymbol{n} + \boldsymbol{V}_\infty\right) \psi_j. \tag{29}$$

Since all the standard Gauss quadrature points are located inside the grid cells, the velocity evaluated with Equation (29) is single valued. Thus, $\chi_x(\boldsymbol{x}) = \sum_{j=1}^{N_V} \hat{\chi_x}_j \psi_j(\boldsymbol{x})$ is the best possible approximation, in the L_2 norm sense, of the x velocity component in space V_h.

A system similar to (27) — with identical matrix M — is solved for the y and z components of the velocity. Thus, the discretized version of Equation (25) reads

$$\hat{\chi_x}_{lw}^{\,2} + \hat{\chi_y}_{lw}^{\,2} + \hat{\chi_y}_{lw}^{\,2} = \hat{\chi_x}_{ww}^{\,2} + \hat{\chi_y}_{ww}^{\,2} + \hat{\chi_y}_{ww}^{\,2}, \tag{30}$$

3.4. Newton iterations solution

The final resolution nonlinear system is assembled combining Equations (24), (26) and (30) as discussed in Section 3.3, where the nonlinearity is given by the quadratic term in Equation (30). The final resolution system is solved by means of the Newton method. The nonlinear solver implemented is typically able to converge to the solution in few iterations.

3.5. Velocity computation on wake and wake relaxation

After the resolution of the nonlinear system, the values of the velocity potential ϕ_h and of the wake potential jump $\delta\phi_h$ are available at the body and wake collocation points, respectively. As discussed in Section 2.3, the position of the wake surface is an unknown of the numerical problem. For such a reason, the values of ϕ_h and $\delta\phi_h$ computed through the resolution of the nonlinear system are used to obtain the fluid velocity values at the wake collocation points, and consequently deform the wake so as to align it with the local velocity field. Once the new wake position is computed, the nonlinear system is solved to obtain the new body surface potential and wake potential jump, leading to new wake collocation point velocities. Such a fixed point strategy is repeated until convergence, which typically requires 5 to 10 iterations.

An important aspect of the procedure just outlined is represented by the computation of the fluid velocity in correspondence with the wake collocation points. Once again, we point out that with our finite elements choice, the collocation points are often located on the edges of the quadrilaterals composing the grid, where the potential gradients, and consequently the velocity, is discontinuous. We could in principle think of avoiding the problem, also in this case, making use of a gradient recovery strategy as in Equation (27). However, the unknown in the wake boundary is not ϕ, but its jump $\delta\phi$ across the vortex sheet. For such a reason, the result of the application of a gradient recovery strategy on such region would result in a function with dubious physical interpretation. In this work, we then resort to the Hypersingular Boundary Integral Equation for the computation of the fluid velocity on the wake. The Hypiersingular BIE is obtained taking the gradient of Equation (16) with respect to $\boldsymbol{x}$. Such equation reads

$$\begin{aligned}\boldsymbol{a}(\boldsymbol{x})\phi(\boldsymbol{x}) + \bar{\bar{C}}(\boldsymbol{x})\boldsymbol{\nabla}\phi(\boldsymbol{x}) + \int_\Gamma^{FP} \phi(\boldsymbol{y})(\boldsymbol{\nabla}(\boldsymbol{\nabla}G(\boldsymbol{y}-\boldsymbol{x}))) \cdot \boldsymbol{n}(\boldsymbol{y})\,\mathrm{d}s_y \\ = \int_\Gamma^{FP} \boldsymbol{\nabla}G(\boldsymbol{y}-\boldsymbol{x})(-\boldsymbol{V}_\infty \cdot \boldsymbol{n}(\boldsymbol{y}))\,\mathrm{d}s_y \qquad (31)\end{aligned}$$

where the notation $\int^{FP}$ indicates the Hadamard's finite part evaluation of the integrals involving the Green's function gradients and Hessian, $\boldsymbol{a}$ is a free coefficient vector and $\bar{\bar{C}}$ is a free coefficient double tensor. Given the free space Green's function here used, the double gradient term in the first

integral takes the form

$$(\boldsymbol{\nabla}(\boldsymbol{\nabla}G(\boldsymbol{y}-\boldsymbol{x})))\cdot\boldsymbol{n}(\boldsymbol{y}) = -\frac{1}{4\pi|\boldsymbol{y}-\boldsymbol{x}|^3}\left(\frac{(\boldsymbol{y}-\boldsymbol{x})\cdot\boldsymbol{n}}{|\boldsymbol{y}-\boldsymbol{x}|^2}(\boldsymbol{y}-\boldsymbol{x})-\boldsymbol{n}\right). \quad (32)$$

Please note that, as pointed out in [15], such synthetic notation is not providing specific information on the — spherical — shape of exclusion region in the finite parts evaluation. Such shape has a direct influence on the specific values assumed by the free coefficients, as well as on the numerical quadrature rules developed Guiggiani et al. in [15] for the integrals evaluation, which we employ in the present work. We then refer the interested reader to [15] for a more detailed description of the equation and its derivation, which goes beyond the scope of the present paper.

After the solution of the potential problem, the collocation of Equation (31) at a point $\boldsymbol{x}\in\Gamma_W$ allows us for the direct evaluation of the potential full gradient $\boldsymbol{\nabla}\phi$, that can be in turn used to compute the local fluid velocity $\boldsymbol{V}_s = \boldsymbol{V}_\infty + \boldsymbol{\nabla}\phi_h$. Ideally, $\boldsymbol{\nabla}\phi_h$ should be evaluated at each BEM collocation point on the wake surface Γ_W, so that it can be readily used to align the wake geometry with the fluid velocity. However, in [15] Equation (31) is derived under the assumptions that function $\frac{\partial\phi}{\partial n}$ is Hölder continuous at $\boldsymbol{x}$, and that ϕ must be differentiable at the same point, with Hölder continuous first derivatives. With our Lagrangian finite elements choice, the latter condition is clearly violated at all the collocation points located on the cell edges and vertices. For such a reason, in this work we resort the weak form of the hypersingular BIE. The corresponding Galerkin numerical problem will be that of finding the approximation of $\boldsymbol{\nabla}\phi$, $\boldsymbol{u}_h\in W_h$, such that

$$\begin{aligned}
&\forall\boldsymbol{v}\in W_h \qquad (33)\\
&\int_\Gamma\Big(\boldsymbol{a}(\boldsymbol{x})\phi_h(\boldsymbol{x})+\bar{\bar{C}}(\boldsymbol{x})\boldsymbol{u}_h(\boldsymbol{x})\Big)\cdot\boldsymbol{v}(\boldsymbol{x})\,\mathrm{d}s_x =\\
&=\underbrace{\int_\Gamma\Big(\int_\Gamma{}^{FP}\Big(\boldsymbol{\nabla}G(\boldsymbol{y}-\boldsymbol{x})(-\boldsymbol{V}_\infty\cdot\boldsymbol{n}(\boldsymbol{y}))-\phi_h(\boldsymbol{y})(\boldsymbol{\nabla}(\boldsymbol{\nabla}G(\boldsymbol{y}-\boldsymbol{x})))\cdot\boldsymbol{n}(\boldsymbol{y})\Big)\mathrm{d}s_y\Big)}_{\boldsymbol{g}(\boldsymbol{x})}\cdot\boldsymbol{v}(\boldsymbol{x})\mathrm{d}s_x.
\end{aligned}$$

Here, the scalar solution space V_h is used to define the space W_h for the vector valued solution function $\boldsymbol{u}_h$ as

$$W_h := \{(u_h)_k\in V_h, k=1,2,3\}\equiv\mathrm{span}\{\xi_i\}_{i=1}^{3N_V}. \quad (34)$$

Note that in Equation (33) the function $\boldsymbol{g}(\boldsymbol{x})$ indicated in the right hand side represents the perturbation potential gradient computed on point $\boldsymbol{x}$ of the external integration loop. In this regard, the main advantage of the Galerkin formulation used for the gradient computation problem in Equation (33) is that, in the framework of the double integrals computation, the Hypersingular BIE singularities are always located on the — Gauss — quadrature nodes. Since such points are always internal to the cells of the computational grid, the Hölder continuous first derivatives requirement on ϕ is satisfied, resulting in a stable numerical algorithm. In addition, the surface of each cell is smooth in correspondence with the quadrature nodes. On such a geometry, the values for the free coefficients on the double sided wake surface are set to $\boldsymbol{a} = \boldsymbol{0}$ and $\bar{\bar{C}} = \bar{\bar{I}}$, where $\bar{\bar{I}}$ is the identity double tensor. Thus, the Galerkin formulation results in the following algebraic linear system

$$A\hat{\boldsymbol{U}} = \boldsymbol{b}^U, \tag{35}$$

where

- $\hat{\boldsymbol{U}} \in \mathbb{R}^{3N_V}$ is the solution vector containing all the components of the potential gradient solution on all the BEM collocation points;
- $A \in \mathbb{R}^{3N_V \times 3N_V} : \; A_{ij} = \int_\Gamma \xi_i(\boldsymbol{x})\xi_j(\boldsymbol{x})\,\mathrm{d}s$ is a sparse mass matrix;
- $\boldsymbol{b}^U \in \mathbb{R}^{3N_V} : \; b_i^U = \int_\Gamma \boldsymbol{g}(\boldsymbol{x}) \cdot \xi(\boldsymbol{x})\mathrm{d}s_x$ is the right hand side vector.

Finally, the velocity on the BEM collocation points is obtained adding the asymptotic velocity to the gradient values contained in $\hat{\boldsymbol{U}}$. The resulting velocity vectors are used to carry out the wake relaxation step. As previously mentioned, and as illustrated in Figure 5, the computational grid is structured in the wake region. Thus, it is possible to arrange the wake collocation points in lines oriented in the streamwise direction. Given a wake degree of freedom iw in the BEM problem, and the degree of freedom pw corresponding to the previous collocation point on a pathline, the position of collocation point iw will be computed as

$$\boldsymbol{x}_{iw} = \boldsymbol{x}_{pw} + \frac{\boldsymbol{v}_i}{|\boldsymbol{v}_i|} d \quad \forall \boldsymbol{x}_i \in \Gamma_W \tag{36}$$

in which d is a user prescribed wake cell length. The wake relaxation step in Equation 36 is carried out on all the wake collocation points, leading to

a modified wake geometry. The BEM nonlinear problem for the velocity potential is then solved making use of the new wake shape, and the new fluid velocities are used to again modify the wake geometry. Such a fixed point iteration strategy is repeated until convergence of the wake position and velocity potential, which in the test cases considered in Sections 4 and 5 requires less than 10 iterations.

4. Convergence tests

This section presents and discusses the numerical results obtained with the lifting surface potential flow solver proposed. The numerical tests have been designed to evaluate the accuracy and spatial convergence capability of both the nonlinear system solved to obtain the potential and the wake relaxation algorithm.

4.1. Test case setup and resolution

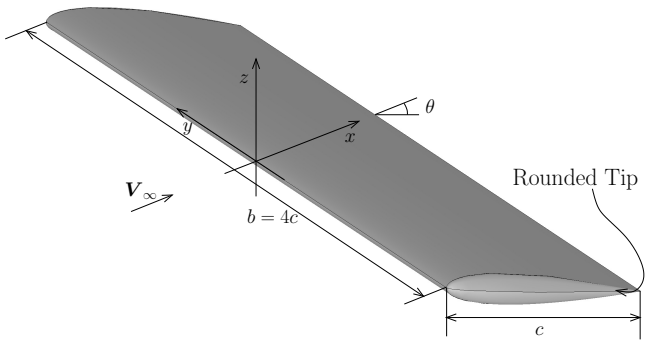

Figure 4: A CAD rendering of the wing used in the first numerical tests, with relevant length measures.

The initial test case considered is the potential flow past a rectangular and constant-section wing. The airfoil section considered is a NACA 0012 with a chord $c = 1$ m, and the wing is generated translating such a shape for a span $b = 4$ m along a direction perpendicular to it. In the simulations, we make use of the right handed reference frame described in Sec. 2.2 in which

axis x is aligned with the inflow velocity vector $\boldsymbol{V}_\infty = V_\infty \hat{\boldsymbol{i}}$, axis z is aligned with the gravity acceleration vector $\boldsymbol{g} = -g\hat{\boldsymbol{k}}$, with $g = -9.81 m/s^2$ and axis y normal to both x and z. In such framework, the NACA 0012 airfoil sections are parallel to the xz plane. The wing span direction is then aligned with the y axis, and the center of the reference frame is located at the leading edge of the section corresponding with the symmetry plane of the wing, as illustrated in Figure 4. As the image shows, the angle of attack between the the chord c and the x axis is $\alpha = 8.5°$. In this particular test the wing tip has been capped with a solid of revolution smoothly connected to the remaining part of the solid. As will be seen, this will help showcasing the ability of the solver developed, to automatically refine the computational mesh on curved CAD surfaces. In fact, an extremely coarse initial mesh composed by three rectangular cells (one covering the leeward side of the wing, one covering the windward side of the wing, and one covering the initially flat wake — having length $4c$) is automatically refined to obtain the desired computational grid for each simulation. As discussed in [4], the automated mesh generation process consists of an initial phase in which the aspect ratio of each cell is limited to a user defined maximum value, followed by user specified numbers of uniform refinement cycles and adaptive refinements based on the local underlying CAD surface curvature.

Examples of the initial grids obtained with the automated mesh generation algorithm are presented in Figure 5. The grid shown on the left, in particular, accounts for a total of 384 cells. Such a grid results in 459, 1683 and 3675 degrees of freedom when bi-linear, bi-quadratic and bi-cubic finite elements are selected, respectively. The additional collocation points required by the high order finite elements are automatically placed on the curved CAD surfaces at the moment of the degrees of freedom distribution. Thus, the switch from low to high order finite elements does not require the generation of specific computational grids. For reference, the solution of the nonlinear problem with 459 degrees of freedom requires few seconds — including jacobian matrix assembling — on a single processor of the Intel Quad Core i7-7700HQ 2.80GHz, 32 GB RAM laptop used to produce all the results shown in this work. After this, the velocity computation combined with the wake relaxation step, requires up to 10 seconds. Thus, the execution time required by 10 wake relaxation cycles is approximately 2 minutes. Scaling up to the largest grids considered in this work, which feature up to 5500 degrees of freedom, the resolution of the nonlinear problem requires approximately

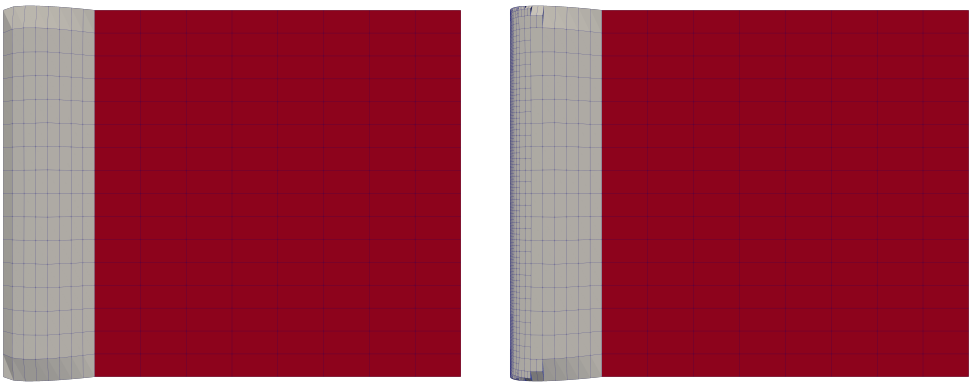

Figure 5: A top view of the automatically generated initial mesh on the wing and wake grid. On the left, the — structured — grid obtained setting maximum aspect ratio 2.5, and 3 uniform refinement cycles. On the right, the — unstructured — grid obtained setting maximum aspect ratio 2.5, 3 uniform refinement cycles, and 4 adaptive refinement cycles based on CAD curvature. Note that, regardless of the wing grid type, the wake mesh is structured. The interface between the two regions, occurring at the trailing edge, is conformal.

a minute, while the computation of the velocity on the wake nodes can take as much as 10 minutes. In such case, the required computation time for the full wake relaxation can reach more than an hour. Things can be made even worse when high order finite elements are considered, as they typically demand higher quadrature order to obtain accurate solutions. Clearly, the computational cost highlighted, can be significantly reduced through parallelization of both the nonlinear system resolution and the wake velocity computation algorithms. For such a reason, a parallel version of the solver is currently being implemented.

4.2. Nonlinear system resolution accuracy and convergence

The first result presented aims at confirming that the algorithm developed is correctly solving the nonlinear system described in Section 3.3. Thus, the wake geometry is kept flat in these first numerical tests presented. Clearly, no analytic solution is available for the realistic — and non analytic — wing geometry selected. For such a reason, the convergence of the resolution algorithm developed is assessed observing the numerical solutions behavior when increasingly refined grids or higher order discretizations are considered.

4.2.1. Potential jump at trailing edge

Figure 6 presents the values of the potential jump $\delta\phi$ obtained at the wing trailing edge, as a function of the span wise coordinate y. The automated grid generation algorithm is here set up to produce structured computational grids (as in 5, on the left), which are progressively refined doubling at each additional cycle the number of nodes in both the span wise and chord wise directions. As the main focus of the present numerical test is that of evaluating the solution at the trailing edge, such structured grids appear better suited to evaluate the spatial convergence of the solver in such region.

The top left plot in Figure 6 refers to the convergence results obtained making use of bi-linear finite elements. In this case, the $\delta\phi$ curves clearly suggest that the solution converges as the number of equispaced nodes on the trailing edge is increased. This is the case also when bi-quadratic and bi-cubic finite elements are considered, as shown in the top right and bottom left diagram, respectively. The limit solutions obtained with finite elements of different degrees are also compared in the bottom right plot of Figure 6. We point out that, because a structured mesh is used in the present test, the bi-linear and bi-quadratic solutions in the bottom right plot feature the same number — and location — of the system degrees of freedom. In fact, given any structured quadrilateral grid, the degrees of freedom obtained using bi-quadratic elements or using bi-linear finite elements on a grid with an additional uniform refinement step are the same. For such a reason, the bottom right plot of Figure 6 suggests that the best solution obtained with bi-linear and bi-quadratic finite elements, when using 128 of degrees of freedom, appear rather similar. Also the solutions obtained with bi-cubic elements appear similar — although with these elements it is not possible to have a comparison with exactly the same degrees of freedom number and distribution used for the lower degree discretizations. However, the plots in Figure 6 also suggest that the high order solutions seem, as expected, to converge faster to the limit solution, as the cubic and quadratic curves obtained on the coarser grids are much closer to the limit solution than their linear counterparts.

4.2.2. Pressure coefficient on wing sections

The second numerical test is aimed at the characterization of the solver accuracy in the estimation of the pressure field on the wing surface. This is relevant for the present application, in which the main goal of the simulations

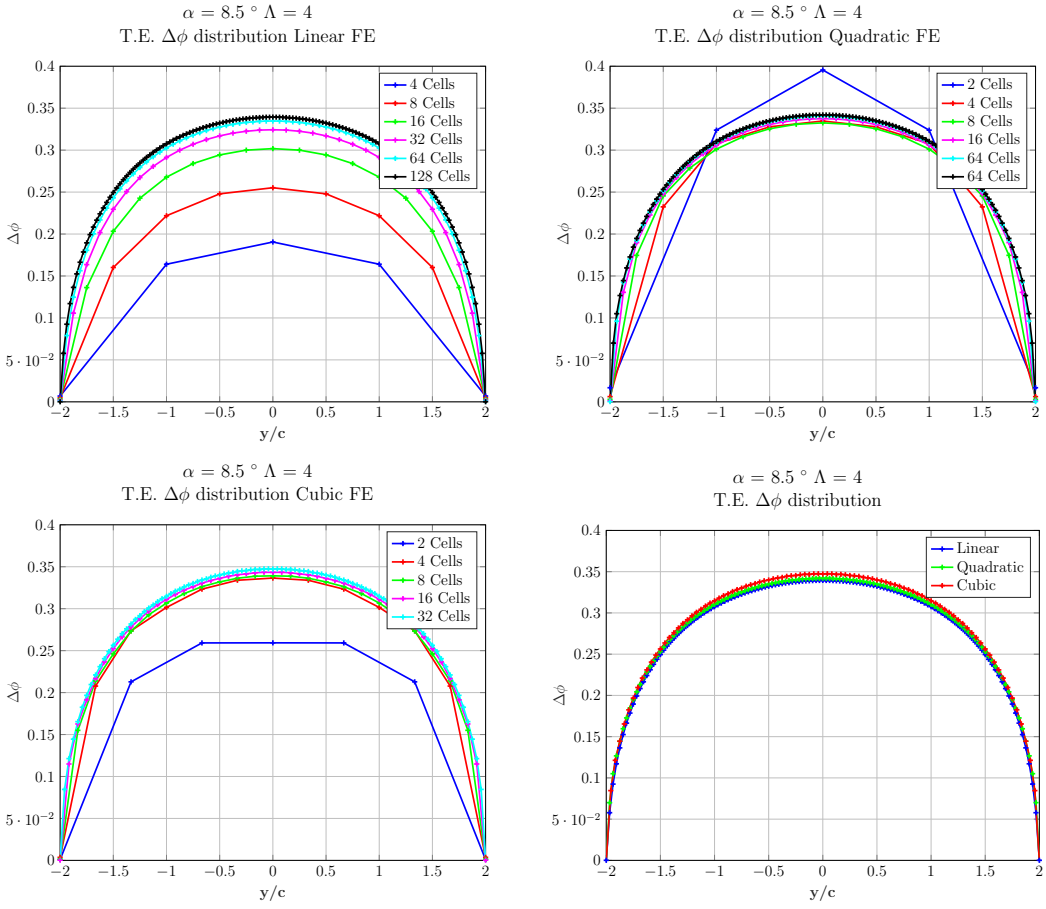

Figure 6: Analysis concerning the distribution of $\delta\phi$ on the wake surface. The graphs refer to a plane section normal to the x-direction. This section is located downstream of the wing trailing edge, at $x/c = 1.05$. In three of these four graphs, a mesh convergence analysis was conducted, while in the fourth (the one on the bottom right), the various degrees of approximation of the solution were compared in terms of the quality of the results.

is the computation of the fluid dynamic forces on the wing. And, as well known, such forces in the context of potential flow theory, are in fact obtained as the integral of the pressure stresses on the wing surface. The pressure field is here evaluated by means of Equation (12), and Figure 7 presents plots of the non dimensional pressure coefficient

$$c_p = \frac{p - p_\infty}{\frac{1}{2}\rho V_\infty^2} \tag{37}$$

at vertical sections of the wing located at different span wise coordinate. The present convergence test starts from a moderately accurate base solution obtained with bi-linear elements on a computational grid generated setting maximum aspect ratio 2.5, 3 uniform refinement cycles, and 2 adaptive refinement cycles based on CAD curvature. To evaluate the effectiveness of possible *hp* refinement strategies, the system degrees of freedom are then increased either by refining the computational grid, or by increasing the finite element degree. The three plots on each row of Figure 7 depict values of $-c_p$ as a function of non dimensional chord wise coordinate x/c, on vertical sections located at non dimensional span wise coordinate values $y/2c = .211$, $y/2c = .611$ and $y/2c = .811$, from left to right. In the first row plots, the base solution (blue lines) and is compared with the one obtained with bi-linear elements on a grid featuring an additional uniform refinement step (green lines). The three diagrams suggest that the bi-linear solution on the finer grid appears significantly less dissipative than the solution on the coarser grid. With the finer grid, the peak c_p value on the wing windward side corresponding to the flow stagnation point is approaching the theoretical limit $c_p = 1$. In addition, also the suction side peak value is significantly higher on the refined grid simulation. As a result of the increased peak values located close to the leading edge, the entire solution amplitude up to the trailing edge appears slightly higher, with a consequent increase of the overall area included between the leeward and windward side c_p lines. A similar behavior is observed also when the base solution degrees of freedom are increased by means of an increment of the discretization degree. The three plots in the second row in of Figure 7 refer to the comparison between the base bi-linear solution c_p values, and the pressure coefficient values obtained making use of bi-quadratic finite elements on the same grid. In the third row plots instead, the base solution c_p field is compared to the one computed making use of the same grid and bi-cubic finite elements. In both cases, the results show that a refinement strategy based on the increase of the discretization degree is effective in reducing numerical diffusion. The peak values in the leading edge region, including the one associated with the stagnation point, are in fact sharper, and as a result the overall area between the leeward and windward curves is increased. The plots in the final row of of Figure 7 finally compare the c_p solutions obtained with the different refinement strategies tested. As

a general trend, the diagrams indicate that the solution obtained with the three refinement strategies appear quite similar. Once again, this seems to suggest that the solver solutions converge as the number of degrees of freedom in the system is increased. A more detailed look at the plots indicates that the bi-linear solution on the finer grid and the bi-quadratic solution on the coarser grid are particularly close to each other across all the regions and sections. Instead, the bi-cubic solution seems to have less numerical dissipation, as its peaks present higher values with respect to the bi-linear and bi-quadratic ones. This could be associated to the fact that, in the case of the present numerical test, the cubic solution has a slightly higher amount of degrees of freedom with respect to the bi-linear and bi-quadratic solutions, which share the same spacing among the collocation points.

4.3. Wake relaxation

After the convergence and accuracy features of the nonlinear problem resolution algorithm have been described and discussed, the present section focuses on highlighting and discussing the performance of the wake relaxation algorithm described in Section 3.5. Also in this case, we make use of the geometrical layout described in Figure 4, and compare the results obtained with different discretization degrees. The grid used for the first bi-linear finite elements test here considered is set up prescribing maximum cell aspect ratio 2.5, 2 uniform refinement cycles, and 1 adaptive refinement cycle based on CAD curvature. To obtain an even comparison in terms of collocation point distribution with respect to the bi-linear test case, the grid generated for the bi-quadratic case features identical refinement parameters, but one less uniform refinement cycle. As for the bi-cubic test case a similar spacing of the collocation points is obtained imposing maximum cell aspect ratio 1.85, 1 uniform refinement cycle, and 1 adaptive refinement cycle based on CAD curvature. In all the simulations discussed in the present sections, the wake relaxation algorithm has been set up with a prescribed wake cell length (see Equation (36)) $d = 0.15$ m, and has been arrested after 9 iterations.

The final wake geometries obtained with the grids just described are depicted in the top row of Figure 8. The left, right and middle images refer to the solutions obtained with bi-linear, bi-quadratic and bi-cubic finite elements. We point out that, in order to show all the collocation points of the solutions obtained with higher degree finite elements, in all the following

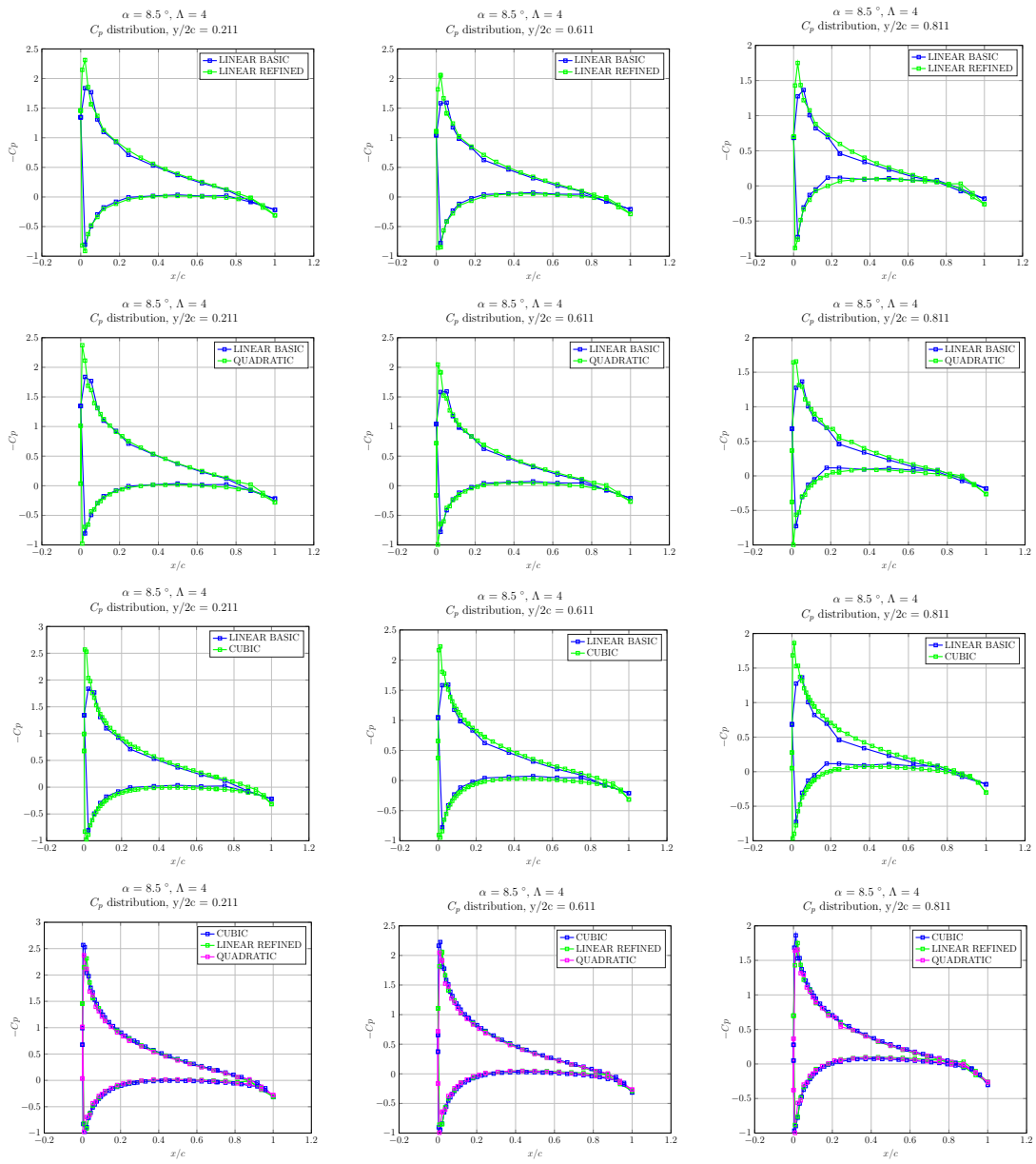

Figure 7: Development of the pressure coefficient in three different wing sections. All of the sections were obtained by using planes parallel to the chord of the profile and perpendicular to the y-axis. These are located at $y/c = 0.211$, 0.611 and 0.811. The comparison was made by analysing four different case studies: starting with a base case, the aim is to observe how the solution changes in response to grid refinement or the use of higher degree finite elements, such as quadratic and cubic.

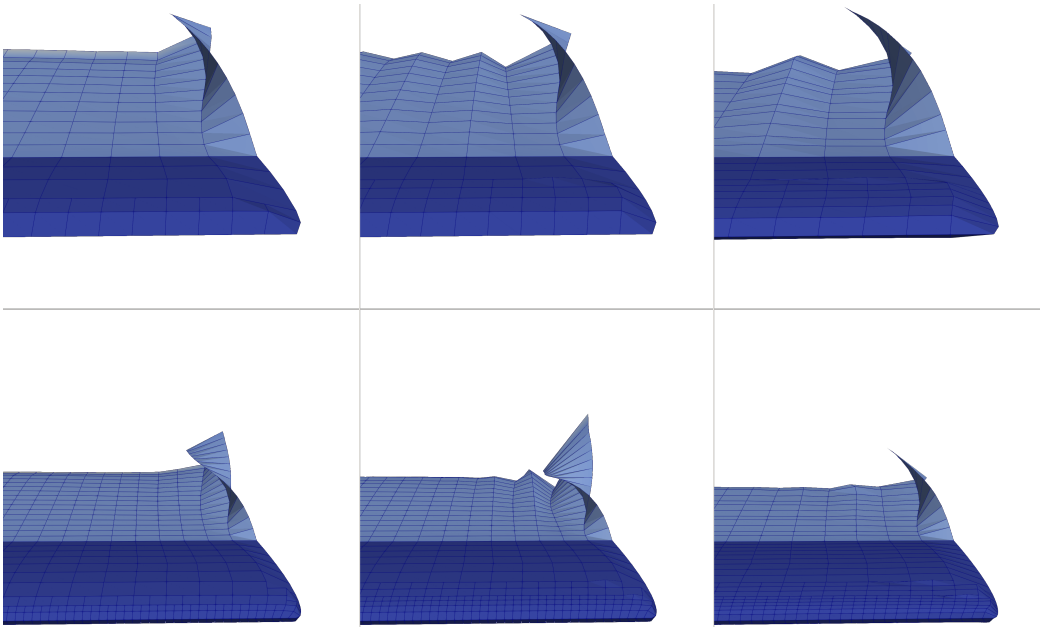

Figure 8: A front view of the grid in the — rounded — tip region of the wing and of the fully developed wake. The three images on the top refer to the base solution obtained with linear (left), quadratic (middle) and cubic (right) finite elements, respectively. The three images on the bottom show the linear (left), quadratic (middle) and cubic (right) solutions obtain with one additional uniform refinement level.

wake geometry visualizations the higher order cells have been split in multiple bi-linear cells. In this way, each bi-quadratic cell is represented in the visualizations as a set of 4 bi-linear cells, whereas each bi-cubic cell is visualized by means of 9 bi-linear cells. The wake plots confirm that the wake velocity computation and wake relaxation algorithm used lead to a qualitatively correct representation of the wake geometry. The images in fact clearly show the presence of a vortex developing past the wing tip, which curls upward the lateral edge of the wake vortex sheet. Instead, in the region located downstream with respect to the wing middle portion, the wake appears flat and aligned with the trailing edge slope. We point out that in the framework of the wake relaxation algorithm developed the wake orientation at the trailing edge is not imposed, but is instead a result of the computations. Thus, the latter observation is quite satisfactory. At a closer look, the wake geometries obtained with bi-linear and bi-quadratic finite elements appear very simi-

lar, especially in the wake roll up region. The bi-cubic solution also appears similar to that obtained with lower degree finite elements, despite being computed making use of a different distribution and number of collocation points. In the central portion of the wake, the bi-quadratic and bi-cubic solutions differ from their bi-linear counterpart, as they both present a series of oscillations. The plots in the bottom row of 8, which have been obtained adding a uniform refinement step to the grids in the top row, confirm the presence of these oscillations. On the finer grid here considered, the oscillations in the bi-quadratic solution appear more visible, as they extend to the wake roll up region, which appears wider than the one computed with bi-linear elements. These oscillations are the consequence of not fully accurate computations of the velocity at the wake collocation points. Higher order shape functions change sign multiple times over each cell, which causes cancellation problems when computing the hypersingular kernel integrals required by the wake velocity Galerkin BEM problem. Clearly, when the oscillations are so severe that they are able to cause self contact of the wake surface, they induce instabilities that lead to simulations break up. Indeed, the problem can be significantly reduced both through an increase of the number of quadrature nodes in both the inner and outer loop of the wake velocity Galerkin BEM problem, and with a choice of shorter cells in the wake. However, we must warn that increasing these parameters leads to more than linear increment in the computational cost of the wake relaxation algorithm. For such a reason, the execution of the velocity computation and wake relaxation algorithm in the present example were extremely more time consuming when higher order finite elements were employed. In addition, to avoid instabilities, the bi-cubic simulation on the finer grid required a shorter wake cell length, which in turn resulted in shorter overall wake length.

Given the aforementioned considerations, in the remainder of the article we will only present solutions based on bi-linear finite elements, in which localized grid refinements are used to compensate for the slower convergence rate with respect to higher order elements, highlighted in Figure 6. A combination of h-adaptivity and linear elements appear in fact to strike a balance between accuracy, stability and efficiency of the simulations. Thus, the next test case is designed to showcase the local refinement capabilities of the solver developed. To this end, three different grids are generated starting from a base grid with maximum cell aspect ratio 1.85, 3 uniform refinement cycles and 1 adaptive refinement cycle based on CAD curvature. So, the first grid considered is obtained adding one cycle refining cells located in corre-

spondence with the rounded tip of the wing. Similarly, the second and third computational grids are generated adding to the base grid two and three local refinement cycles in the tip region, respectively. Figure 9 shows the results obtained using bi-linear elements on the three grids generated. As can be appreciated in the plots, the rounded tip geometry is reproduced with greater accuracy as the grid is locally refined. This has clearly a consequence on the computed tip vortex, which appears confined in a smaller region when finer tip grids are considered (right plot), as opposed to the wider curl computed with coarser grids. The wider wake tip vortex appearing in the coarser mesh simulation (left plot) is also affecting the nearby portion of the wake vortex sheet, which appears less flat than the one computed with the finer meshes. Clearly, this can have significant effects on the accuracy of the computed downwash caused by the wing presence.

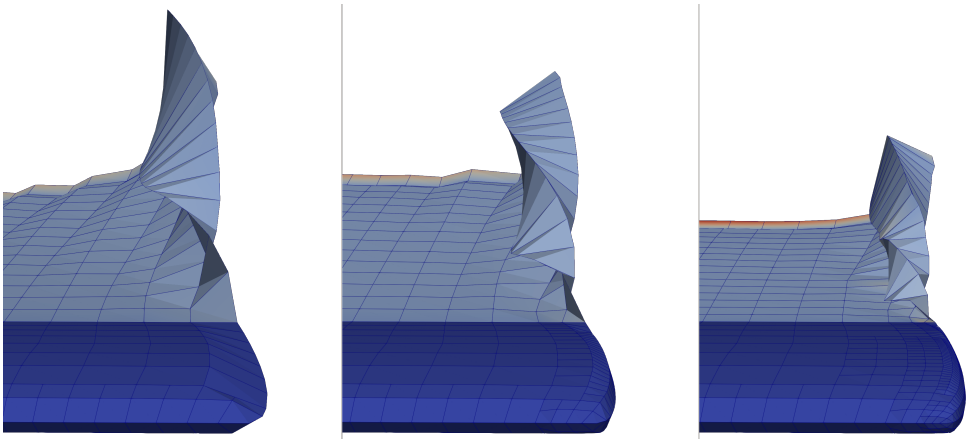

Figure 9: A front view of the grid in the — rounded — tip region of the wing and of the fully developed wake. The three images refer to the solution obtained with the base grid (on the left), with one additional refinement on the tip (in the middle), and with two additional refinements on the tip (on the right).

Further confirmation of the effect that tip refinements can have on the overall wake shape is given by Figure 10. The plot displays a vertical cut of the converged wake geometry on a plane located at constant longitudinal coordinate $x = 2.5$ m. The curves in the plot show that adding nodes to the final portion of the wake vortex sheet increases the tendency of the wake edge to curl up in a spiraling fashion. As this results in smaller spiral coils,

the influence of the tip vortex presence on the central part of the wake is weakened. Thus, the central portion of the wake geometry seems to converge to a more flat solution as the tips are refined.

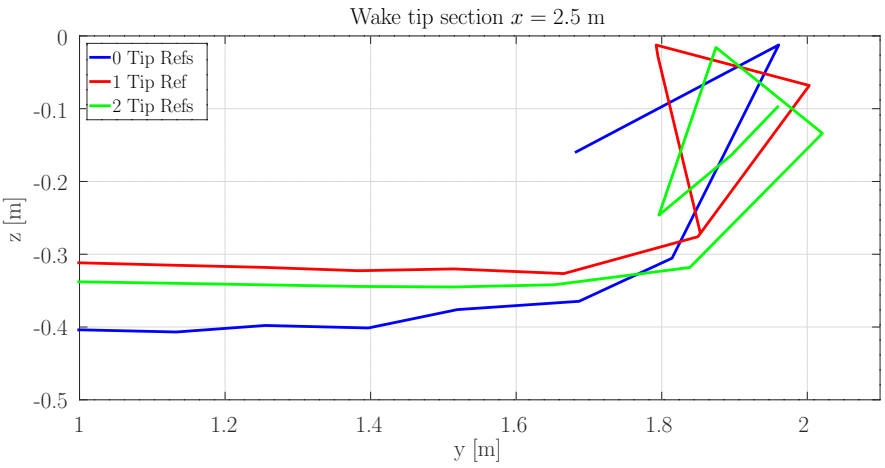

Figure 10: Wing tip refinement effect on a section of the wake located 1.5 chord lengths downstream with respect to the trailing edge. The blue, red and green curves in the plot refer to solutions obtained with the base solution, one and two tip refinements, respectively.

5. Comparison with experimental data

As a final confirmation of the accuracy of the mathematical model presented in this work, the present section is devoted to the comparison of the numerical results with experimental measurements available in the literature. The test cases considered are presented in [29] and both feature a semi-wing with constant NACA 0012 airfoil section and chord length $c = 1$ m. In the first test, the wing has a regular rectangular shape with semi-span $s = 2.95c$.

The semi-wing is mounted on a wall on one side, and has a free tip on the other. In the second test, the wing is rotated along the vertical axis z to obtain a swept semi-wing in which the leading edge is inclined by an angle $\beta = 20^\circ$ with respect to the y axis. In such a case, the semi-wing span reported is $s = 2.91c$. In the specific experiments here considered, both wings are set at an angle of attack $\alpha = 6.75^\circ$.

5.0.1. Rectangular wing

The experimental setup corresponding to the rectangular semi-wing described, is reproduced in the simulations with a geometric arrangement by all means identical to that of the numerical tests presented in section 4. To obtain an equivalent flow on a wing in free air — as opposed to the experimental one on a semi-wing — we considered an aspect ratio of $\Lambda = 5.9$, corresponding in our case to a wingspan $b = 5.9c = 5.9$ m.

The geometric CAD model, designed using FreeCad, features in this case a flat cap surface on each wing tip. An initial mesh made of approximately 10 cells is then refined onto the surface of the CAD model imposing maximum cell aspect ratio 4.5, 2 uniform refinement cycles and 1 adaptive refinement cycle based on CAD curvature. This results in a computational grid featuring approximately 3800 quadrilateral cells. Bi-linear finite elements are used for this test, which result in a resolution system of 4240 degrees of freedom.

Figure 11, presents a comparison between experimental and numerical values of $-c_p$ on four cross-sections perpendicular to the wingspan direction. The wing sections are located at wingspan coordinates $y/c = 0.200$, 0.350, 0.610, 0.900. In each section, the numerical values of windward and leeward pressure coefficients obtained imposing a flat wake (blue lines) and with relaxed wake (green lines) are plotted against their experimental counterparts (red dots).

By a qualitative standpoint, the comparisons appear satisfactory, as the model appears able to reproduce with accuracy the main features of the experimental curves. By a more quantitative point of view, the peaks location is well captured and their height, especially on the windward side stagnation point, appears accurately reproduced by the numerical results. Also the height of the suction side peaks seems reasonably accurate, although slightly underestimated. This is more evident for the peaks on the sections closer to the tip. Such a discrepancy is likely due to the potential flow model inability to reproduce viscous effects associated with the increasingly close tip vortex.

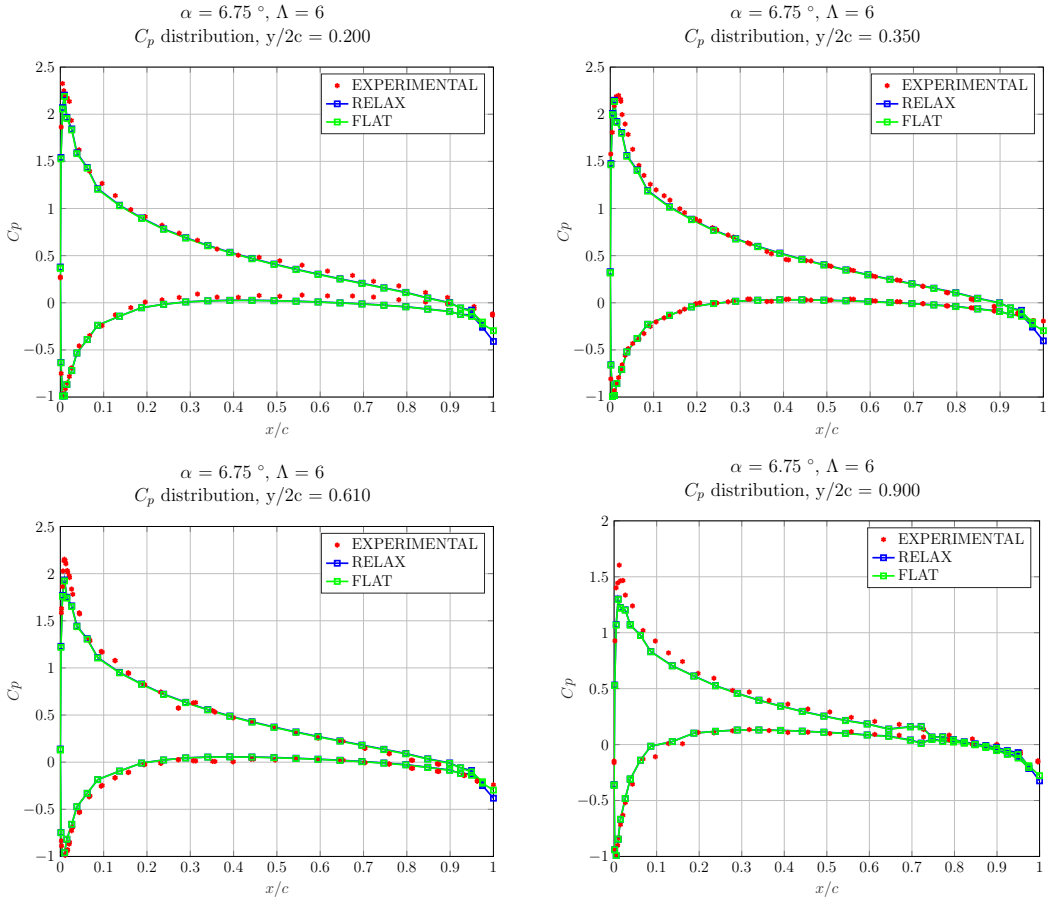

Figure 11: Comparative analysis between results obtained using Π-BEM and experimental data for a rectangular planform wing. The plots refer to four planar sections normal to the y-axis located along the semi-wingspan direction at $y/s = 0.050, 0.350, 0.610, 0.900$.

In addition, in every plot in Figure 11 the numerical solution appears to be consistently overestimating the experimental pressure at the trailing edge. This should be in part associated to the sharp trailing edge considered in the potential flow simulations which — as opposed to the finite thickness one used in the experimental campaign — leads to higher pressure recovery. Finally, we point out that the results obtained with flat wake are extremely close to the ones obtained including the wake relaxation part of the algorithm. The only visible differences can be in fact observed at the trailing edge, where the relaxed wake solution consistently exhibits higher pressure values than its flat

wake counterpart. On one hand, such a marginal difference might not seem enough to justify the additional computational cost associated with the wake velocity computation and wake relaxation part of the resolution algorithm. On the other hand, there surely are different applications in which the wake geometry has more influence on the computed body pressure, with respect to the isolated wing considered in the present numerical tests. This is definitely the case for propellers, in which the helical shape of the wake is typically closer to the propeller blades than it is in the example here considered.

5.1. Swept wing

The experimental setup corresponding to the semi-wing with sweep angle $\beta = 20^\circ$ previously described, is also in this case reproduced in the simulations by considering the entire extent of a V-shaped wing, as illustrated in Figure 12. In the experimental setup [29], the swept wing was obtained first by rotating the NACA 0012 rectangular wing previously considered, and then adding a fairing at the tip. This resulted in a wing having semi-wing span $s = 2.91c$. The geometric CAD model, designed using FreeCad, features again flat cap surfaces on the wing tips. The initial mesh made of approximately 10 cells is refined onto the surface of the CAD model imposing, also in this case, maximum cell aspect ratio 4.5, 2 uniform refinement cycles and 1 adaptive refinement cycle based on CAD curvature. This results in a computational grid featuring approximately 4000 quadrilateral cells. Bi-linear finite elements are used for this test too, which result in a resolution system of 4520 degrees of freedom. Figure 12 depicts the computational grid on the wing and on the fully relaxed wake. The body surface is colored according to contours of pressure coefficient c_p.

Figure 13 presents a comparison between experimental and numerical values of $-c_p$ on four cross-sections perpendicular to the wing axis direction. We point out in fact that because in the experimental setup the swept wing has been obtained merely rotating the NACA 0012 wing model, all the pressure taps in this test are not here aligned with the incident flow, but are instead inclined of an angle $\beta = 20^\circ$ with respect to such a direction. The wing sections are located at coordinates $y/s = 0.260$, 0.510, 0.710, 0.840, as measured at their intersection with the trailing edge. In each section, also in this case the numerical values of windward and leeward pressure coefficients obtained imposing a flat wake (blue lines) or computing the relaxed wake

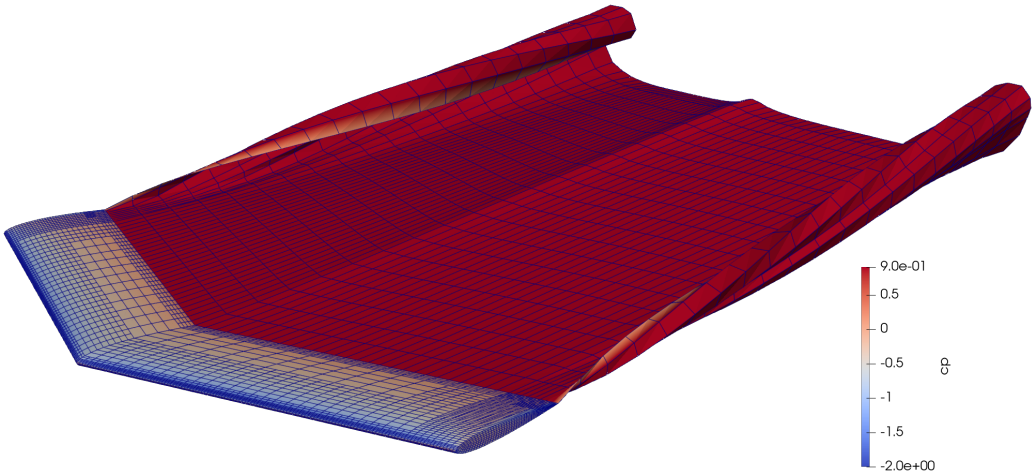

Figure 12: A view of the grid obtained on the geometry of the NACA 0012 wing with aspect ratio $\Lambda = 5.47$ and sweep angle $\beta = 20°$, and on the fully developed wake following the body. The surfaces are colored according to contours of pressure coefficient c_p.

(green lines) are plotted against their experimental counterparts (red dots).

The results appear quite consistent with the ones obtained on the rectangular wing geometry. The comparisons with the experimental results appear in fact once again satisfactory, by a qualitative perspective. The experimental curves are in fact reproduced in all their main traits by the numerical results. By a more quantitative point of view, also in this case the peaks location appear accurately reproduced by the numerical results. As for the peaks height, on one hand the windward side pressure peak is captured with good accuracy also in this swept wing case, in which a stagnation point is missing and the c_p maximum is approximately 0.9. On the other hand, the peaks on the suction side are slightly underestimated, as was the case for the rectangular wing. And, also in the present case, the discrepancy seems more evident for the sections closer to the tip, likely due to viscous effects associated with the tip vortex. Again, we can observe that the sharp trailing edge considered in the numerical simulations — as opposed to the experimental finite thickness one — results in a noticeable difference in correspondence with the last pressure probe measurement, at every section considered. Finally, we remark that also in the present case the effect of wake relaxation appears marginal, except on the trailing edge region, in which the relaxed wake solution predicts higher pressure recovery.

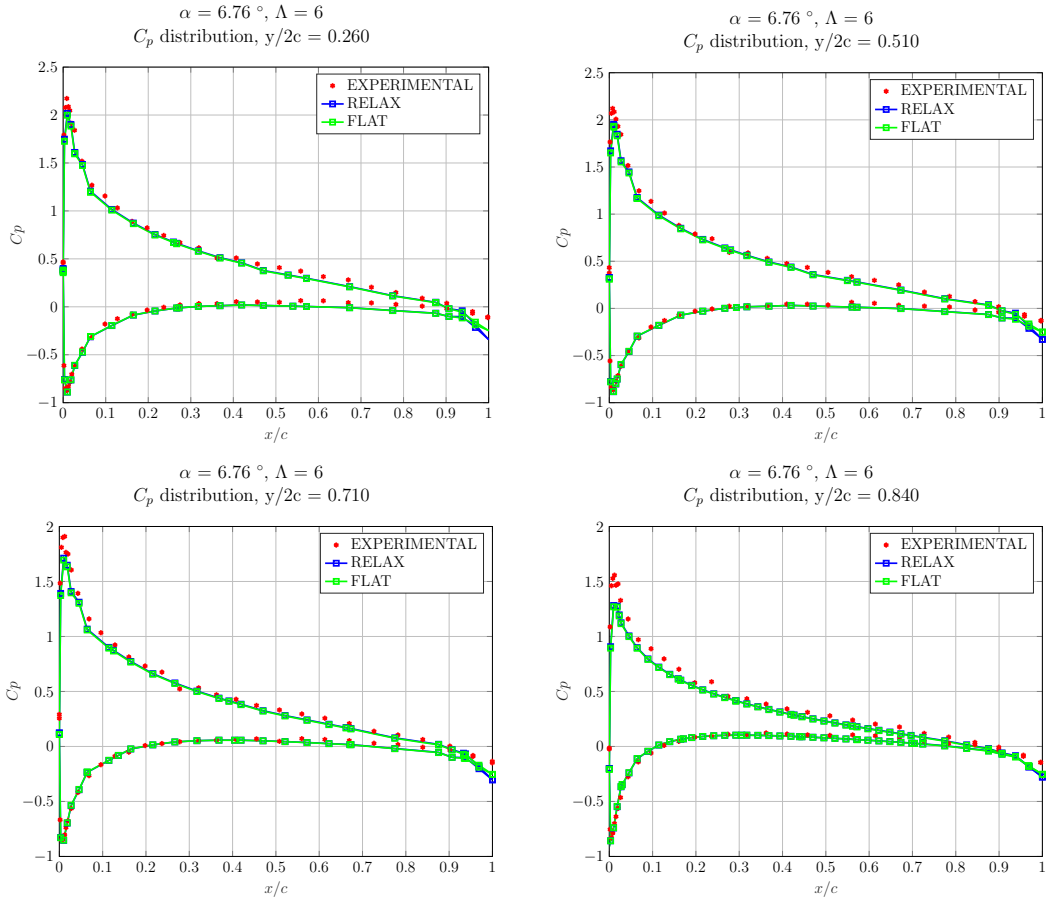

Figure 13: Comparative analysis between results obtained using Π-BEM and experimental data for a swept wing with a sweep angle of 20°. The plots refer to four planar sections normal to the wing axis, and located at coordinates $y/s = 0.260$, 0.510, 0.710, 0.840, as measured at their intersection with the trailing edge.

6. Conclusions and future perspectives

This work presented a numerical method for the simulation of quasi-potential flow past lifting surfaces. The solver combines the collocation Boundary Element Method formulation implemented in the open source library π-BEM [4] with a Galerkin formulation of the nonkinear Kutta condition imposed at the lifting body trailing edge. In addition, Galerkin formulation is used also for the Hypersingular Boundary Integral Equation solved to obtain the velocities at the wake collocation points needed for wake relax-

ation. The latter techniques, borrowed from the Finite Element Method, allow for the evaluation of the solution derivatives in a way that is independent of the local grid topology. As such, the proposed Boundary Element Method based solver allows for the user selection of the finite element degree at the start of each simulation. This is also made possible by the direct interface with CAD files, which are interrogated to obtain new points for high order elements grids on curved surfaces. Numerical experiments on rectangular and V-shaped wings with NACA 0012 airfoil sections reproduce in a satisfactory way experimental data. The results on rectangular wings also confirm the quality of the arbitrary order numerical approach implemented. In particular, the higher order solutions show less numerical dissipation in the leading edge region, which is characterized by the highest gradients. However, in BEM the polynomial shape functions are multiplied by singular Green functions so as to obtain the linear system matrices and vectors. Thus, because high order shape functions change sign multiple times within each cell, increased accuracy granted by high order elements can only be obtained at the price of a significant increase in the Gaussian quadrature nodes used for the BEM integrals. For such a reason, the combination of linear elements and additional local refinements appears to better balance accuracy and computational cost. In the near future, a possible way to best exploit the accuracy premium that higher order elements offer is that of implementing suitable *hp* refinement algorithms. Indeed, this paper showed that the methodology implemented possesses all the ingredients necessary for the implementation of such a strategy. Another interesting possible extension of this work, is that of using the CAD interface for the mapping of the reference cell to three dimensional cells in the calculation of the integrals involved in BEM. This approach, which is also discussed in [6], has never been applied to BEM. As isogeometric analysis, such a *heterogeometric* approach would allow for the calculation of integrals on the exact surface prescribed by the user, but without the need to use NURBS shape functions also for the solution. Finally, further developments of this work can come from an extension of the range of application of the model. In particular, the free of a surface in proximity of the wing can be initially introduced as a homogeneous Neumann boundary condition, and can be then treated with a linearized [18] or fully nonlinear [30] formulation. Including the presence of a free surface in close to the wing would make the solver able to estimate the forces generated by hydrofoils in different relative position with respect to the free surface, and contribute to the design of such artifacts.

7. Acknowledgments

We acknowledge the support of Piano Nazionale di Ripresa e Resilienza (PNRR) and ToolsPole for co-funding the PhD scholarship of Luca Cattarossi. Andrea Mola acknowledges project ROMEU: Reduced Order Models for Environmental and Urban flow, (CUP D53D230188800) funded by Ministero Università e Ricerca (MUR), Italy. We finally thank Istituto Nazionale di Alta Matematica "Francesco Severi" (INdAM), Gruppo Nazionale per il Calcolo Scientifico (GNCS) for the support.

References

[1] W. F. Phillips, D. O. Snyder, Modern adaptation of prandtl's classic lifting-line theory, Journal of Aircraft 37 (4) (2000) 662–670. arXiv:https://doi.org/10.2514/2.2649, doi:10.2514/2.2649.
URL https://doi.org/10.2514/2.2649

[2] D. D. Liu, P. C. Chen, Z. X. Yao, D. Sarhaddi, Recent advances in lifting surface methods, The Aeronautical Journal 100 (998) (1996) 327–340. doi:10.1017/S0001924000067038.

[3] J. Hess, A. Smith, Calculation of potential flow about arbitrary bodies, Progress in Aerospace Sciences 8 (1967) 1–138. doi:https://doi.org/10.1016/0376-0421(67)90003-6.
URL https://www.sciencedirect.com/science/article/pii/0376042167900036

[4] N. Giuliani, A. Mola, L. Heltai, π-bem: A flexible parallel implementation for adaptive, geometry aware, and high order boundary element methods, Advances in Engineering Software 121 (2018) 39 – 58. doi:https://doi.org/10.1016/j.advengsoft.2018.03.008.
URL http://www.sciencedirect.com/science/article/pii/S0965997818300371

[5] S. Gaggero, S. Brizzolara, Exact Modeling of Trailing Vorticity in Panel Method for Marine Propeller, 2nd International Conference on Marine Research and Transportation, 2007, ischia, 28-30 June.

[6] L. Heltai, W. Bangerth, M. Kronbichler, A. Mola, Propagating geometry information to finite element computations, ACM Trans. Math. Softw.

47 (4) (sep 2021). doi:10.1145/3468428.
URL https://doi.org/10.1145/3468428

[7] Boundary Integral Equation Methods for Aerodynamics, pp. 279–320. arXiv:https://arc.aiaa.org/doi/pdf/10.2514/5.9781600866180.0279.0320, doi:10.2514/5.9781600866180.0279.0320.
URL https://arc.aiaa.org/doi/abs/10.2514/5.9781600866180.0279.0320

[8] M. Gennaretti, F. Salvatore, L. Morino, Forces and moments in incompressible quasi-potential flows, Journal of Fluids and Structures 10 (3) (1996) 281–303. doi:https://doi.org/10.1006/jfls.1996.0017.
URL https://www.sciencedirect.com/science/article/pii/S0889974696900171

[9] C. Xu, Kutta condition for sharp edge flows, Mechanics Research Communications 25 (4) (1998) 415–420. doi:https://doi.org/10.1016/S0093-6413(98)00054-8.
URL https://www.sciencedirect.com/science/article/pii/S0093641398000548

[10] S. Chouliaras, P. Kaklis, K. Kostas, A. Ginnis, C. Politis, An isogeometric boundary element method for 3d lifting flows using t-splines, Computer Methods in Applied Mechanics and Engineering 373 (2021) 113556. doi:https://doi.org/10.1016/j.cma.2020.113556.
URL https://www.sciencedirect.com/science/article/pii/S0045782520307416

[11] A. Mola, L. Heltai, A. DeSimone, A stable and adaptive semi-Lagrangian potential model for unsteady and nonlinear ship-wave interactions, Eng. Anal. Bound. Elem. 37 (1) (2013) 128–143. doi:10.1016/j.enganabound.2012.09.005.
URL http://www.sciencedirect.com/science/article/pii/S0955799712001907http://linkinghub.elsevier.com/retrieve/pii/S0955799712001907

[12] A. Mola, L. Heltai, A. De Simone, Ship Sinkage and Trim Predictions Based on a CAD Interfaced Fully Nonlinear Potential Model, Vol. All Days of International Ocean and Polar Engineering Conference, 2016, iSOPE-I-16-438. arXiv:

https://onepetro.org/ISOPEIOPEC/proceedings-pdf/ISOPE16/All-ISOPE16/ISOPE-I-16-438/1337253/isope-i-16-438.pdf.

[13] F. Dassi, A. Mola, H. Si, Curvature-adapted remeshing of cad surfaces, Procedia Engineering 82 (2014) 253–265, 23rd International Meshing Roundtable (IMR23). doi:https://doi.org/10.1016/j.proeng.2014.10.388.
URL https://www.sciencedirect.com/science/article/pii/S1877705814016671

[14] D. J. Bernasconi, A higher-order potential flow method for thick bodies, thin surfaces and wakes, Ph.D. thesis, University of Southampton (2007).
URL https://eprints.soton.ac.uk/466428/

[15] M. Guiggiani, G. Krishnasamy, T. J. Rudolphi, F. J. Rizzo, A General Algorithm for the Numerical Solution of Hypersingular Boundary Integral Equations, Journal of Applied Mechanics 59 (3) (1992) 604–614. arXiv:https://asmedigitalcollection.asme.org/appliedmechanics/article-pdf/59/3/604/5462606/604_1.pdf, doi:10.1115/1.2893766.
URL https://doi.org/10.1115/1.2893766

[16] G. Grimberg, W. Pauls, U. Frisch, Genesis of d'alembert's paradox and analytical elaboration of the drag problem, Physica D: Nonlinear Phenomena 237 (14) (2008) 1878–1886, euler Equations: 250 Years On. doi:https://doi.org/10.1016/j.physd.2008.01.015.
URL https://www.sciencedirect.com/science/article/pii/S0167278908000225

[17] L. Morino, G. Bernardini, Singularities in bies for the laplace equation; joukowski trailing-edge conjecture revisited, Engineering Analysis with Boundary Elements 25 (9) (2001) 805–818. doi:https://doi.org/10.1016/S0955-7997(01)00063-7.
URL https://www.sciencedirect.com/science/article/pii/S0955799701000637

[18] N. Giuliani, A. Mola, L. Heltai, L. Formaggia, FEM SUPG stabilisation of mixed isoparametric BEMs : Application to linearised free surface flows, Engineering Analysis with Boundary Elements 59 (2015) 8–22.

doi:10.1016/j.enganabound.2015.04.006.
URL http://linkinghub.elsevier.com/retrieve/pii/S0955799715001058

[19] C. A. Brebbia, The Boundary Element Method for Engineers, Pentech Press, 1978.

[20] N. Giuliani, A. Mola, L. Heltai, pi-BEM: Parallel BEM Solver., https://github.com/mathLab/pi-BEM (2021).

[21] D. Arndt, W. B. M. Feder, M. Fehling, R. Gassmöller, T. Heister, L. Heltai, M. Kronbichler, M. Maier, P. Munch, J.-P. Pelteret, S. Sticko, B. Turcksin, D. Wells, The deal.II library, version 9.4, Journal of Numerical MathematicsAccepted (2022).
URL https://dealii.org/deal94-preprint.pdf

[22] W. Bangerth, D. Davydov, T. Heister, L. Heltai, G. Kanschat, M. Kronbichler, M. Maier, B. Turcksin, D. Wells, The deal.II library, Version 8.4, Journal of Numerical Mathematics 24 (3) (2016) 135–141.

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

[24] A. Mola, L. Heltai, A. DeSimone, A fully nonlinear potential model for ship hydrodynamics directly interfaced with CAD data structures, in: 24th International Ocean and Polar Engineering Conference, 2014.
URL http://digitallibrary.sissa.it/handle/1963/7311

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

[26] N. Giuliani, An hybrid boundary element method for free surface flows, Master's thesis, Politecnico di Milano (2013).

[27] S. T. Grilli, P. Guyenne, F. Dias, A fully non-linear model for three-dimensional overturning waves over an arbitrary bottom, Journal for Numerical Methods in Fluids 35 (7) (2015) 29–67. doi:https://doi.org/10.1002/1097-0363(20010415)35:7<829::AID-FLD115>3.0.CO;2-2.

URL https://onlinelibrary.wiley.com/doi/abs/10.1002/1097-0363(20010415)35:7%3C829::AID-FLD115%3E3.0.CO;2-2

[28] S. E. Mousavi, N. Sukumar, Generalized Duffy transformation for integrating vertex singularities, Computational Mechanics 45 (2-3) (2010) 127–140. doi:10.1007/s00466-009-0424-1.

[29] L. Yip, G. Shubert, Pressure distributions on a 1- by 3-meter semispan wing at sweep angles from 0° to 40° in subsonic flow, Tech. Rep. TN D-8307, NASA (1976).

[30] A. Mola, N. Giuliani, Óscar Crego, G. Rozza, A unified steady and unsteady formulation for hydrodynamic potential flow simulations with fully nonlinear free surface boundary conditions, Applied Mathematical Modelling 122 (2023) 322–349. doi:https://doi.org/10.1016/j.apm.2023.06.001.
URL https://www.sciencedirect.com/science/article/pii/S0307904X23002445