Available online at www.sciencedirect.com
ScienceDirect

Comput. Methods Appl. Mech. Engrg. 388 (2022) 114244

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

An embedded approach for the solution of the full potential equation with finite elements

Marc Núñez[a,b,*], Iñigo López[c], Joan Baiges[a,b], Riccardo Rossi[a,b]

[a] Centre Internacional de Mètodes Numèrics a l'Enginyeria (CIMNE), C/ Gran Capità s/n Campus Nord UPC, 08034, Barcelona, Spain
[b] Universitat Politècnica de Catalunya (UPC), Campus Nord UPC, Barcelona, 08034, Spain
[c] Technische Universität München (TUM), Arcisstraße 21, München, 80333, Germany

Received 9 April 2021; received in revised form 29 July 2021; accepted 11 October 2021
Available online 9 November 2021

Abstract

Recent developments in numerical simulations in the aerodynamics field are focused on reducing the computational cost of the solvers, targeting their use at the initial design steps. Under the right assumptions, potential solvers can be exploited to accomplish a valid solution of the flow field for streamlined bodies in subsonic and close to transonic flows. A fully embedded approach to solve the full-potential equation is presented, where both the geometry and the wake are defined implicitly with a level set function. Embedded methods allow simplifying the mesh generation process, as only a background mesh is required to perform the analysis. The presented method gives an automatic and fast option to solve subsonic flows, which is especially desirable in optimization workflows.

Keywords: Finite elements; Embedded; Aerodynamics; Level set; Potential flow; Adaptive refinement

1. Introduction

Flow simulation is a powerful tool used in the early development of aerodynamic designs, especially if compared to expensive wind tunnel experiments, where the size of the object of study and the input conditions variability are limited. Nonetheless, Navier–Stokes solvers, while being highly accurate, require a great amount of computational power. Moreover, in order to yield meaningful results these methods require input parameters that are generally not available at early stages [1]. In the initial design phase of any engineering project, several design options are considered, and the project requirements may vary. For this reason, there is an interest from the industry in having a fast, yet sufficiently accurate aerodynamic solver. In this regard, full-potential formulations stand often as the best option in terms of accuracy-to-cost ratio [2], as only one single scalar partial–differential equation is required. This poses an advantage of more than one order of magnitude in computation time if compared to Euler solvers, where the momentum and energy equations are also considered. Full-potential formulations are irrotational and isentropic, and are suitable for subsonic and transonic flows, assuming that no strong shock waves exist in the

[*] Corresponding author.
E-mail addresses: mnunez@cimne.upc.edu (M. Núñez), inigo.lopez@tum.de (I. López), joan.baiges@upc.edu (J. Baiges), riccardo.rossi@upc.edu (R. Rossi).

https://doi.org/10.1016/j.cma.2021.114244

flow [3]. High mach number flows where strong shock waves are present invalidate the isentropic flow hypothesis. If compared to classic panel-methods, panel-methods are not able to deal with non-linear flows, so full-potential flows are expected to give better solutions in subsonic cases, since no linearization of the mass equation is performed [4]. As a disadvantage, the non-vorticity of the flow requires the addition of a discontinuity in the domain, which makes difficult other aspects such as mesh generation. This discontinuity represents an enforced vorticity that allows the potential flow to generate lift [5,6]. In order to deal with this discontinuity (also known as potential jump or wake), a physical separation of the domain is performed. Considering high Reynolds numbers, the wake can be assumed to be straight and to have the same direction as the free-stream velocity. A first approach to model the wake is by duplicating the surfaces of the wake [4,7]. This approach allows for a fast solution of subsonic flows, but the wake mesh generation still poses an important challenge, especially if several analyses are to be considered, or when dealing with moving geometries. A first step to deal with this issue is presented in [8], where the wake is defined implicitly in a classical body-fitted mesh using a level set function. Thus, no explicit mesh representation is needed for the wake. The potential jump required to model lift is accomplished by enforcing embedded discontinuities in the elements that are cut by the wake, by using new local degrees of freedom. In this case, changing the wake direction does not require to completely remesh the whole domain, allowing for fast analyses on different angle of attack configurations. Nonetheless, the use of classical body-fitted meshes to represent the geometry can be a limiting factor when considering the analysis of complex geometries or moving bodies, as it is the case in shape optimization. For this reason a full embedded approach is desired, where both the geometry of study and the wake are implicitly defined in a background mesh by signed-functions [9]. This approach would allow solving fast the subsonic and transonic flow around a complex structure, while also permitting changes or movements of the geometry without mesh deformation. Proper representation of the embedded geometry by the background mesh can be accomplished by means of adaptive remeshing [10,11]. Recent work on the usage of the full-potential equation can be highlighted for multidisciplinary [12] and aeroelastic [13] analyses, where the reduced computational cost of the potential flow equation is exploited. Regarding embedded formulations, there are already developments in the solution of the full-potential flow for finite-volume methods using Cartesian meshes [14]. Also, there are immersed methods to solve Navier–Stokes flows mixing finite-elements and finite-volumes [15]. In this document, a complete embedded finite-element method for the full-potential flow is presented, where the embedded-wake approach introduced in [8] is extended to model the aerodynamic shapes using embedded methods. Numerical terms are defined to robustly model the potential flow around the level set while also respecting the Kutta condition. The presented method has been implemented in the open source framework KratosMultiphysics [16,17], available under the BSD licence.

2. Embedded potential flow solver

Most aerodynamic applications involve streamlined bodies flying at very large Reynolds numbers, especially on classical commercial aircraft configurations at cruise speed. Under these conditions, the potential flow hypotheses can be assumed, where the outer flow is considered isentropic and irrotational. The potential flow assumptions are however not valid within the boundary layer and the wake, where viscous effects are not negligible. As the Reynolds number grows, the area enclosed by the boundary layer and the wake is reduced, as the inertial forces are considerably greater than the viscous forces. This allows performing accurate estimations of the flow over a streamlined body using potential flow assumptions over the whole domain. Starting from Navier–Stokes equations, assuming that the flow is irrotational and inviscid, and assuming that the velocity can be written as the gradient of a potential such that $\mathbf{v} = \nabla \varPhi$, the mass conservation equation reduces to:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \nabla \varPhi) = 0 \tag{1}$$

where the density, ρ, can be written in terms of far-field quantities:

$$\rho(\varPhi) = \rho_\infty \left[1 + \frac{\gamma - 1}{2} \frac{v_\infty^2}{a_\infty^2} \left(1 - \frac{\nabla \varPhi \cdot \nabla \varPhi}{v_\infty^2} \right) \right]^{\frac{1}{\gamma - 1}} \tag{2}$$

where ρ_∞, v_∞ and a_∞ denote the magnitude of the freestream density, velocity and speed of sound respectively. γ is the adiabatic index. This relation comes after applying the potential flow assumption in Euler's momentum equation. Pressure can be computed using the isentropic relation:

$$\frac{p}{p_\infty} = \left(\frac{\rho}{\rho_\infty} \right)^\gamma \tag{3}$$

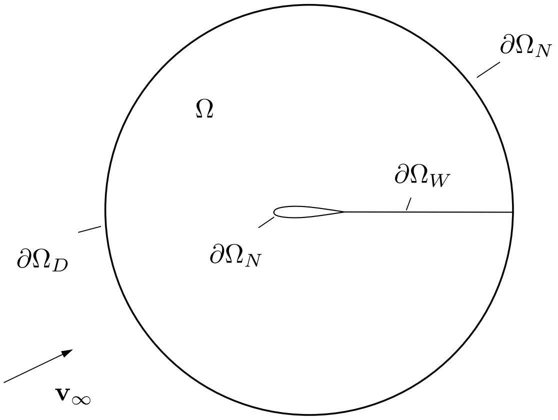


Fig. 1. Simplified representation of the domain Ω and its boundary $\partial\Omega$. The wake $\partial\Omega_W$ is modelled as a straight line.

where p_∞ is the freestream pressure. The assumption of inviscid and irrotational flow precludes the generation of lift, as no pressure difference can be created between the upper and lower surfaces of the aerodynamic body. The lack of viscosity prevents the flow from slowing down in contact with the solid surfaces [18]. It is therefore not possible to compute the lift force that the air generates on a solid body under potential flow assumptions. In order to do so, the addition of a discontinuity in the domain is needed, which will produce the vorticity required to generate lift [6]. This discontinuity models the so-called wake. The lift force per unit span for an airfoil with chord c, can be expressed in terms of the flow circulation Γ by the Kutta–Joukowski theorem $L = \rho v_\infty \Gamma$, which yields the following expression for the lift coefficient:

$$C_l = \frac{L}{\frac{1}{2}\rho v_\infty^2 c} = \frac{\rho v_\infty \Gamma}{\frac{1}{2}\rho v_\infty^2 c} = \frac{2}{v_\infty c}\Gamma \tag{4}$$

It can be shown that the circulation created by the wake is equivalent to the potential jump across the wake $\Gamma = \Phi^+ - \Phi^-$ [5] . The $+$ sign refers to the potential values above the wake, and the $-$ sign to the potential values below the wake.

2.1. Boundary conditions

The boundary conditions defined in the problem are equivalent as those introduced in [8]. A scheme is shown in Fig. 1, where the far-field boundary is represented by the outer circle. A line denoted by $\partial\Omega_W$ represents the wake modelled in the problem. Depending on the flux passing through the domain, the imposed conditions are defined as $\partial\Omega_D$ or $\partial\Omega_N$, which will be considered the inlet and the outlet of the domain, respectively. If $\mathbf{n}$ is defined as the outer normal on the boundary, then $\partial\Omega_N$ and $\partial\Omega_D$ are defined as follows:

$$\partial\Omega_\infty = \begin{cases} \partial\Omega_D & \text{if} \quad \mathbf{v}_\infty \cdot \mathbf{n} < 0 \\ \partial\Omega_N & \text{if} \quad \mathbf{v}_\infty \cdot \mathbf{n} \geq 0 \end{cases} \tag{5}$$

At the inlet, a Dirichlet condition is imposed by fixing the potential values. For a given point $\mathbf{x}$ in the inlet, its potential is computed using the free stream velocity $\mathbf{v}_\infty$ and a given constant value, as shown in Eq. (6). A Neumann condition is imposed in Eq. (7) at the outlet by setting a mass flux g. At the walls, a no-penetration condition is set with $g = 0$. Also, the wake requires the imposition of two specific boundary conditions. Eq. (8) imposes mass conservation across the wake. Eq. (9) imposes pressure equality between the upper and lower parts of the wake.

$$\Phi(\mathbf{x}) = \mathbf{v}_\infty \cdot \mathbf{x} + \Phi_\infty \qquad \text{on} \quad \partial\Omega_D \tag{6}$$

$$\mathbf{n} \cdot (\rho \nabla \Phi) = g \qquad \text{on} \quad \partial\Omega_N \tag{7}$$

$$\mathbf{n} \cdot (\rho^+ \nabla \Phi^+ - \rho^- \nabla \Phi^-) = 0 \qquad \text{on} \quad \partial\Omega_W \tag{8}$$

$$\left|\nabla \Phi^+\right|^2 - \left|\nabla \Phi^-\right|^2 = 0 \qquad \text{on} \quad \partial\Omega_W \tag{9}$$

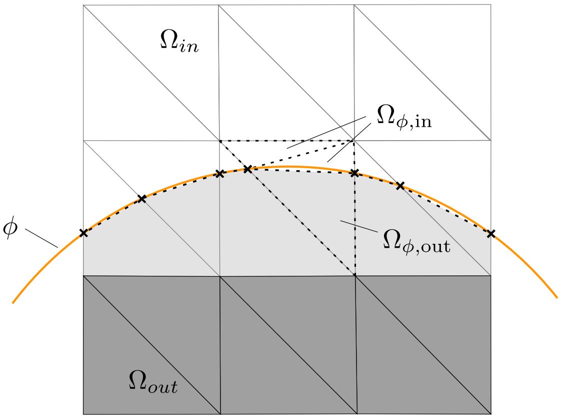


Fig. 2. Embedded approach. The geometry is represented by the level set function's zero-level and its intersections with the elemental edges are used to split the domain.

2.2. Embedded problem formulation and discretization

In order to solve the full-potential equation introduced in the previous sections, a complete embedded approach with a background mesh with finite elements is presented. The geometry of study is represented by a continuous level set function, denoted by ϕ, which is defined as a signed scalar field [19]. This is illustrated in Fig. 2., where the level set function's zero-level value is indicated as an orange line, and its intersections with the elements' edges are marked with black crosses. The dashed lines denote that the background mesh does not capture the geometry exactly, but the intersection points are used to perform a linear split of the elements. This zero-level intersection subdivides the computational domain into two partitions $\Omega = \Omega_{in} \cup \Omega_{out}$, depending on its sign. The positive part is considered as the region laying inside the fluid domain, and it is marked in white in Fig. 2. The negative part is considered as the region laying outside the fluid domain. Note that this convention is arbitrary. The elements intersected with ϕ are denoted with $\Omega_\phi = \Omega_{\phi,\text{out}} \cup \Omega_{\phi,\text{in}}$ for which splitting is performed, and only the part $\Omega_{\phi,\text{in}} \in \Omega_{in}$ is considered. The part $\Omega_{\phi,\text{out}} \in \Omega_{out}$ does not contribute to the system and it is marked in light grey in Fig. 2. The elements outside the fluid domain and not intersected by the level set function are deactivated. These elements are depicted in dark grey in Fig. 2. The function ϕ is defined as:

$$\phi(x) = \begin{cases} \phi(x) < 0 & \text{if} \quad x \in \Omega_{out} \\ \phi(x) \geq 0 & \text{if} \quad x \in \Omega_{in} \end{cases} \tag{10}$$

Similarly, a distance function $\phi_{\text{wake}}(x)$ is used to model the wake region $\partial\Omega_W$. This implies that both the airfoil and the wake cut the mesh elements in an arbitrary manner. The topological discontinuity of the domain at $\partial\Omega_W$ implies that the elements of the background mesh cut by $\partial\Omega_W$ (and their degrees of freedom) are duplicated and assigned to the discretization of the regions above and below $\partial\Omega_W$ respectively. The wake line will cut the elements in Ω in a region which is denoted as $\Omega_W = \Omega_{W+} \oplus \Omega_{W-}$, where the "+" and "−" sign refers to the upper and lower elements with respect to the wake. This is illustrated in Fig. 3, where a variable to account for the duplicated degrees of freedom on the wake is introduced, Φ^{ext}, which is represented by blue squares. The original degrees of freedom Φ are marked with red dots. This extra variable enables the definition of a discontinuity in the potential and the imposition of the boundary conditions of the wake, which are defined on the external side of the wake. Thus, the degrees of freedom above and below the wake are given by:

$$\Phi^+ = \begin{cases} \Phi & \text{if} \quad \phi_{\text{wake}} > 0 \\ \Phi^{\text{ext}} & \text{if} \quad \phi_{\text{wake}} < 0 \end{cases} \quad \forall \Omega^e \in \Omega_{W+} \tag{11}$$

$$\Phi^- = \begin{cases} \Phi^{\text{ext}} & \text{if} \quad \phi_{\text{wake}} > 0 \\ \Phi & \text{if} \quad \phi_{\text{wake}} < 0 \end{cases} \quad \forall \Omega^e \in \Omega_{W-} \tag{12}$$

The elements in Ω_{in} and their degrees of freedom define the finite element space for the problem, V_h, which also includes the cut volume integration and the duplicated degrees of freedom of the wake. The rest of elements outside

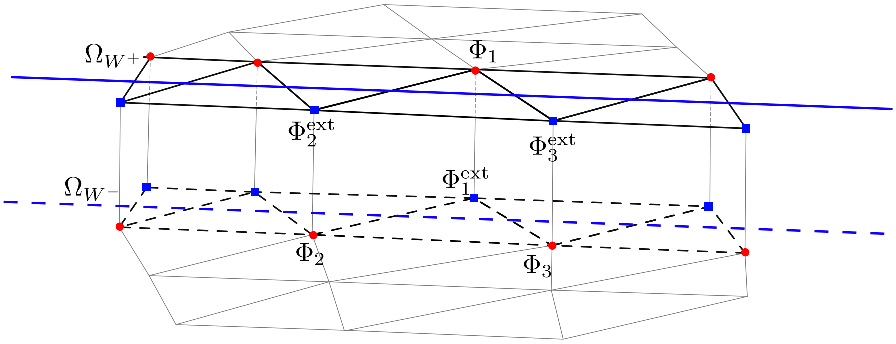


Fig. 3. Embedded wake visualization, where the wake is represented by a blue line. The elements intersected by the wake have duplicated degrees of freedom, which creates a discontinuity in the domain, as if the mesh was split in two parts. Note that the separation shown in the figure is simply a representation, as the mesh is not physically split in this approach. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the fluid (and inside the volume) are discarded. The finite element space V_h is decomposed into $V_h^{\text{main}} \oplus V_h^{\text{ext}}$. In this decomposition, V_h^{ext} is the finite element space spanned by the functions associated to the nodes of V_h which belong to the external part of the domain in the wake cut elements, while V_h^{main} is the finite element space spanned by the functions associated to the nodes of V_h which are part of the rest of the fluid domain (see Fig. 3).

Moreover, two auxiliary terms are added to ensure a robust handling of the cut element integration. On one hand, a stabilization term is added in the elements $\Omega_\phi \subset \Omega$, which corresponds to the set of elements intersected by the geometry. This term is used to improve the instabilities produced by the level set interpolation performed at the elements. This effect is visible in the pressure distribution plots, which is improved with the addition of the stabilization term, especially on coarser meshes. On the other hand, the region $\Omega_{\text{kutta}} \subset \Omega$ is defined, which corresponds to the set of elements that are intersected by both the geometry and the wake, and its neighbours. In these elements, a penalty term is added to enforce the Kutta condition for any arbitrary combination of the distance functions defining the wake and the geometry. Thus, the weak form of the problem can be introduced as:

$$B(\omega_h, \Phi_h)_{\Omega_{in}} + B_{\text{stab}}(\omega_h, \Phi_h)_{\Omega_\phi} + B_{\text{kutta}}(\omega_h, \Phi_h)_{\Omega_{\text{kutta}}} = F(\omega_h)_{\partial\Omega} \quad \forall \omega_h \in V_h^{\text{main}} \tag{13}$$

$$B_{\partial\Omega_{W+}}(\omega_h, \Phi_h^+, \Phi_h^-)_{\partial\Omega_{W+}} = 0 \quad \forall \omega_h \in V_h^{\text{ext}} \tag{14}$$

$$B_{\partial\Omega_{W-}}(\omega_h, \Phi_h^-, \Phi_h^+)_{\partial\Omega_{W-}} = 0 \quad \forall \omega_h \in V_h^{\text{ext}} \tag{15}$$

where the terms involved are defined next. The derivation of these terms is described in the following sections.

$$B(\omega_h, \Phi_h)_{\Omega_{in}} = \int_{\Omega_{in}} \rho \nabla\omega_h \cdot \nabla\Phi_h d\Omega \tag{16}$$

$$F(\omega_h)_{\partial\Omega_N} = \int_{\partial\Omega_N} \omega_h \mathbf{n} \cdot (\rho\nabla\Phi_h) d\partial\Omega \tag{17}$$

$$B_{\partial\Omega_{W+}}(\omega_h, \Phi_h^+, \Phi_h^-)_{\partial\Omega_{W+}} = \int_{\partial\Omega_{W+}} \nabla\omega_h[\nabla\Phi_h^+ - \nabla\Phi_h^-] d\partial\Omega \tag{18}$$

$$B_{\partial\Omega_{W-}}(\omega_h, \Phi_h^-, \Phi_h^+)_{\partial\Omega_{W-}} = -\int_{\partial\Omega_{W-}} \nabla\omega_h[\nabla\Phi_h^+ - \nabla\Phi_h^-] d\partial\Omega \tag{19}$$

$$B_{\text{kutta}}(\omega_h, \Phi_h)_{\Omega_{\text{kutta}}} = k_{\text{kutta}} \int_{\Omega_{\text{kutta}}} \nabla\omega_h \mathbf{n}_{\text{kutta}}^T \mathbf{n}_{\text{kutta}} \nabla\Phi_h d\Omega \tag{20}$$

$$B_{stab}(\omega_h, \Phi_h)_{\Omega_\phi} = k_{\text{stab}} \int_{\Omega_\phi} \nabla\omega_h(\nabla\Phi_h - \overline{\xi(\nabla\Phi_h)}) d\Omega \tag{21}$$

The terms in Eqs. (16) and (17) correspond to the terms from the Laplacian problem with the far-field acting as a forcing term, and they are presented in Section 2.3. The terms in Eqs. (18) and (19) correspond to the imposition of the boundary conditions on the wake, and they are introduced in Section 2.4. Eq. (20) is an auxiliary term used to enforce the Kutta condition with the penalty coefficient k_{kutta}. This term forces the velocity direction to locally match the trailing edge bisector line direction, whose normal is defined as $\mathbf{n}_{\text{kutta}}$, which is explained in detail in Section 2.6. Finally, Eq. (21) is a stabilization term used to smooth the gradient of the potential. To accomplish this, the gradient of the potential differing from the average of the gradients of the potential of the neighbouring elements is penalized with the factor k_{stab}. Here, the term $\overline{\xi(\nabla\Phi_h)}$ refers to the volume-based average of the gradient of the potential, which is described in depth in Section 2.7. From now on and for the sake of simplicity, the subscript $()_h$ will be omitted to refer to the discretized potential values $\Phi = \Phi_h$ and the test functions $\omega = \omega_h$.

2.3. Full-potential equation

The derivation of the terms introduced in Eq. (13) starts with the general expression of the potential formulation, as introduced in [4] and [8], extending it for embedded geometries. While a level set function is used in [8] to define the wake, the airfoil shapes considered are still body-fitted. In this paper, both the geometry of study and the wake are modelled using embedded methods.

From Eq. (1), the weak form of the system can be derived by applying the Galerkin method and the divergence theorem, for the test functions ω:

$$\int_{\Omega} \rho\nabla\omega\cdot\nabla\Phi d\Omega = \int_{\partial\Omega_N} \omega\mathbf{n}\cdot(\rho\nabla\Phi)d\partial\Omega \tag{22}$$

The left-hand side of the equation corresponds to Eq. (16), and the right-hand side of the equation to Eq. (17). The dependence of ρ on the potential makes the problem non-linear, so the system is written in residual form $R(\Phi) = 0$. The residual is written in terms of the M nodal shape functions N_j and the M corresponding nodal values Φ_j. Also, the new domains defined, Ω_{in} and Ω_{out} are considered. Naturally, only the fluid domain is modelled, so only the elements that belong to Ω_{in} are integrated. The elemental contribution to the residual for the elements in Ω_{in} is given by:

$$R_i^e(\Phi) = \sum_{j=1}^{M}\int_{\Omega_{in}^e} \rho\nabla N_i\cdot\nabla N_j d\Omega\Phi_j - \int_{\partial\Omega_N^e} N_i g d\partial\Omega \tag{23}$$

For the elements intersected by ϕ, only the region inside the fluid $\Omega_{\phi,\text{in}}$ is integrated after splitting. This operation is performed element-wise, when computing the local contribution of each subdivision of the domain. Also, a no-penetration condition ($\mathbf{v}\cdot\mathbf{n} = 0$) is imposed on the geometry by setting to zero the elemental contribution to the boundary term. The residual for the intersected elements by ϕ is therefore given by:

$$\int_{\partial\Omega_\phi} N_i g d\partial\Omega = 0 \quad \text{in} \quad \Omega_\phi \tag{24}$$

$$R_{i,\phi}^e(\Phi) = \sum_{j=1}^{M}\int_{\Omega_{\phi,\text{in}}^e} \rho\nabla N_i\cdot\nabla N_j d\Omega\Phi_j \quad \text{in} \quad \Omega_\phi \tag{25}$$

The elemental contributions to the Jacobian are denoted with $J_{i,j}^e(\Phi)$ which are computed as:

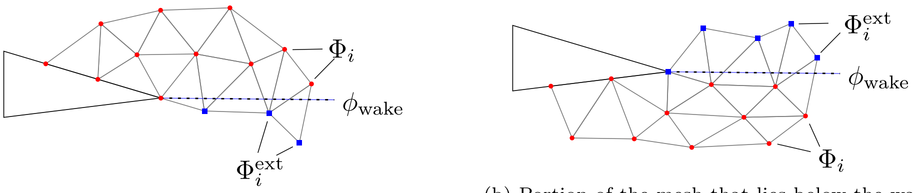
$$J_{i,j}^e(\Phi) = \frac{\partial R_i^e(\Phi)}{\partial\Phi_j} = \int_{\Omega_{in}^e} \rho\nabla N_i\cdot\nabla N_j + 2\frac{\partial\rho}{\partial|\mathbf{v}|^2}\left(\nabla N_j\cdot\nabla\Phi\right)\left(\nabla N_i\cdot\nabla\Phi\right) \tag{26}$$

where the derivative of the density with respect to the local velocity is obtained from (2):

$$\frac{\partial\rho}{\partial|\mathbf{v}|^2} = -\frac{\rho_\infty}{2a_\infty^2}\left(1+\frac{\gamma-1}{2}\frac{v_\infty^2}{a_\infty^2}\left(1-\frac{|\mathbf{v}|^2}{v_\infty^2}\right)\right)^{\frac{2-\gamma}{\gamma-1}} \tag{27}$$

2.4. Wake boundary conditions

As it is well-known, the definition of a wake is required in potential flow methods in order to model lift. Therefore, a method to account for the discontinuity of the wake and its boundary conditions is required. This discontinuity is imposed explicitly on a finite-element mesh in [4] by effectively duplicating the nodes on the wake.



(a) Portion of the mesh that lies above the wake.

(b) Portion of the mesh that lies below the wake.

Fig. 4. Representation of an embedded wake in a body fitted mesh introduced in [8]. The wake is defined implicitly through a distance function, and its intersection subdivides the mesh into two, shown in Fig. 4a and 4b. In this case, the trailing edge is clearly defined and it is used as the wake origin. Note that the duplication is only performed for visualization purposes, as only one mesh is utilized.

This wake would act as a standard solid wall if no additional conditions were imposed. Then, mass conservation and pressure equality across the wake is enforced with the boundary conditions presented in Eqs. (8) and (9) respectively. It must be noted that this approach requires the a priori definition of the wake in the meshing process. This is not a major drawback for stand-alone analyses with modern preprocessors, but it can pose a clear disadvantage if the same mesh is meant to be kept for different geometry configurations.

A viable alternative to model the wake is to implicitly define it with a level set function, as described in [8]. In order to create a domain discontinuity in this scenario, the elements intersected by the level set function ϕ_{wake} have duplicated degrees of freedom. This is illustrated in Fig. 3, where the discontinuity is accomplished by disconnecting the degrees of freedom from the upper and lower parts of the wake. The auxiliary degrees of freedom are also employed to enforce the wake boundary conditions. The wake is shed from the body-fitted airfoil as a straight line as shown in Fig. 4. In this paper, this approach is adapted to embedded geometries, where also the airfoil is defined using a level set function. The imposition of the boundary conditions on the wake elements on the auxiliary degrees of freedom is equivalent to [8], which is described next. Then, the specific required modifications for the current approach will be presented in Section 2.5, which will focus on the intersection of the geometry and the wake and the proper definition of the Kutta condition.

The main potential flow terms are evaluated in the wake elements on both the upper and lower side of the domain, which are included in the evaluation of Eqs. (16) and (17) with its corresponding degrees of freedom as depicted in Fig. 3. The residuals and Jacobians derived from these equations are rewritten for the elements that are intersected by the wake using its corresponding values of potential according to Eqs. (11) and (12). These elemental residuals and Jacobians are denoted with $R^e_{i,W+}$, $R^e_{i,W-}$ and $J^e_{i,j,W+}$, $J^e_{i,j,W-}$ respectively. When evaluating the upper terms of the wake in Ω_{W+}, Eq. (11) is used, assigning Φ_i and Φ_i^{ext} accordingly. If the lower terms are evaluated in Ω_{W-}, Eq. (12) is used instead:

$$R^e_{i,W+}(\Phi^+) = R^e_i(\Phi^+) = \sum_{j=1}^{M} \int_{\Omega^e_W} \rho^+ \nabla N_i \cdot \nabla N_j d\Omega \Phi^+_j - \int_{\partial\Omega^e_W} N_i g d\partial\Omega \tag{28}$$

$$R^e_{i,W-}(\Phi^-) = R^e_i(\Phi^-) = \sum_{j=1}^{M} \int_{\Omega^e_W} \rho^- \nabla N_i \cdot \nabla N_j d\Omega \Phi^-_j - \int_{\partial\Omega^e_W} N_i g d\partial\Omega \tag{29}$$

From these residuals, Jacobians for the wake elements are derived in the same manner as in (26), by computing $J^e_{i,j,W+} = \frac{\partial R^e_i(\Phi^+)}{\partial \Phi^+_j} = J^e_{i,j}(\Phi^+)$ and $J^e_{i,j,W-} = \frac{\partial R^e_i(\Phi^-)}{\partial \Phi^-_j} = J^e_{i,j}(\Phi^-)$.

Boundary conditions for both sides of the wake are imposed by a least-square finite-element approach, combining Eqs. (8) and (9) in a single two-dimensional vector equation, where the velocity on both sides of the wake is enforced to be equal:

$$\rho^+ \mathbf{v}^+ = \rho^- \mathbf{v}^- \quad \text{in} \quad \partial\Omega_W \tag{30}$$

Here, $\rho^+ = \rho(\Phi^+)$ and $\rho^- = \rho(\Phi^-)$ are defined using Eq. (2) with the degrees of freedom of the wake. This equation is added to the system by minimizing the functional Π, where the velocity vectors have been replaced by the gradients of the potential on each side of the wake, and it is given by:

$$\Pi(\Phi^+, \Phi^-) = \frac{1}{2} \int_{\partial\Omega_W} \|\rho^+ \nabla\Phi^+ - \rho^- \nabla\Phi^-\|^2 d\partial\Omega \tag{31}$$

This expression will act as a constraint in the element formulation. The residual terms and its Jacobians are found by deriving the functional in Eq. (31), where derivatives on the density are neglected due to the assumption of small velocity perturbations $\mathbf{u} = \mathbf{v} - \mathbf{v}_\infty \ll \mathbf{v}_\infty$, as only streamlined bodies are considered. Here, $\mathbf{u}$ represents the velocity difference between the free-stream unperturbed flow $\mathbf{v}_\infty$, and the actual velocity around the airfoil $\mathbf{v}$. Deriving with respect to the degrees of freedom Φ^+, Φ^- leads to Eqs. (18) and (19):

$$\frac{\partial \Pi(\Phi^+, \Phi^-)}{\partial \Phi^+} = B_{\partial\Omega_{W+}}(w, \Phi^+, \Phi^-)_{\partial\Omega_{W+}} \tag{32}$$

$$\frac{\partial \Pi(\Phi^+, \Phi^-)}{\partial \Phi^-} = B_{\partial\Omega_{W-}}(w, \Phi^-, \Phi^+)_{\partial\Omega_{W-}} \tag{33}$$

Since the wake is embedded in the elements, the above expression that is evaluated on the wake line is replaced by a volume integral. This assumes that the gradient of the potential is constant across the elements, which holds true for linear elements. The elemental contributions of the residual for the constraint are written with the subscript $R^e_{i,B}$, to refer to the terms enforcing the boundary conditions of the wake:

$$R^{e,+}_{i,B}(\Phi^+, \Phi^-) = \frac{\partial \Pi(\Phi^+, \Phi^-)}{\partial \Phi^+_i} = \int_{\Omega^e_W} \nabla N_i \left(\rho^+ \nabla \Phi^+ - \rho^- \nabla \Phi^-\right) d\partial\Omega \tag{34}$$

$$R^{e,-}_{i,B}(\Phi^+, \Phi^-) = \frac{\partial \Pi(\Phi^+, \Phi^-)}{\partial \Phi^-_i} = -\int_{\Omega^e_W} \nabla N_i \left(\rho^+ \nabla \Phi^+ - \rho^- \nabla \Phi^-\right) d\partial\Omega \tag{35}$$

which can be rewritten in terms of the original problem residual:

$$R^{e,+}_{i,B}(\Phi^+, \Phi^-) = R^e_{i,W+}(\Phi^+) - R^e_{i,W-}(\Phi^-) \tag{36}$$

$$R^{e,-}_{i,B}(\Phi^+, \Phi^-) = R^e_{i,W-}(\Phi^-) - R^e_{i,W+}(\Phi^+) \tag{37}$$

where Jacobians can also be written in terms of the original problem Jacobians:

$$J^{e,+}_{B,i,j} = \frac{\partial R^{e,+}_{i,B}}{\partial \Phi^+_j}(\Phi^+, \Phi^-) = J^e_{i,j,W+}(\Phi^+) - J^e_{i,j,W-}(\Phi^-) \tag{38}$$

$$J^{e,-}_{B,i,j} = \frac{\partial R^{e,-}_{i,B}}{\partial \Phi^-_j}(\Phi^+, \Phi^-) = J^e_{i,j,W-}(\Phi^-) - J^e_{i,j,W+}(\Phi^+) \tag{39}$$

The terms in the system of equations obtained in Eqs. (34) to (39) are equivalent to those defined by the full-potential equation in Eq. (26), only adapting the degrees of freedom according to Eqs. (11) and (12). If this is expressed in matrix form for a triangular element with the same cutting as the element shown in Fig. 3, the residuals and Jacobians on the auxiliary degrees of freedom are written as:

$$J^e_{B,i,j} = \begin{bmatrix} J^{e,+}_{B,i,j} & J^{e,-}_{B,i,j} \end{bmatrix} = \begin{bmatrix} J^+_{B,11} & J^+_{B,12} & J^+_{B,13} & J^-_{B,11} & J^-_{B,12} & J^-_{B,13} \\ J^+_{B,21} & J^+_{B,22} & J^+_{B,23} & J^-_{B,21} & J^-_{B,22} & J^-_{B,23} \\ J^+_{B,31} & J^+_{B,32} & J^+_{B,33} & J^-_{B,31} & J^-_{B,32} & J^-_{B,33} \end{bmatrix} \tag{40}$$

$$R^e_{i,B} = \begin{bmatrix} R^{e,+}_{i,B} \\ R^{e,-}_{i,B} \end{bmatrix} = \begin{bmatrix} R^+_{2,B} \\ R^+_{3,B} \\ R^-_{1,B} \end{bmatrix} \tag{41}$$

These terms replace the terms in the auxiliary degrees of freedom of Eqs. (28) and (29) and its Jacobians, acting as a constraint:

$$J^e_{i,j,W} = \begin{bmatrix} J^e_{i,j,W+} & J^e_{i,j,W-} \end{bmatrix} = \begin{bmatrix} J^+_{11} & J^+_{12} & J^+_{13} & 0 & 0 & 0 \\ J^+_{B,11} & J^+_{B,12} & J^+_{B,13} & J^-_{B,11} & J^-_{B,12} & J^-_{B,13} \\ J^+_{B,21} & J^+_{B,22} & J^+_{B,23} & J^-_{B,21} & J^-_{B,22} & J^-_{B,23} \\ J^+_{B,31} & J^+_{B,32} & J^+_{B,33} & J^-_{B,31} & J^-_{B,32} & J^-_{B,33} \\ 0 & 0 & 0 & J^-_{21} & J^-_{22} & J^-_{23} \\ 0 & 0 & 0 & J^-_{31} & J^-_{32} & J^-_{33} \end{bmatrix} \tag{42}$$

$$R^{e}_{i,W} = \begin{bmatrix} R^{e,+}_{i,W} \\ R^{e,-}_{i,W} \end{bmatrix} = \begin{bmatrix} R^{+}_{1} \\ R^{+}_{2,B} \\ R^{+}_{3,B} \\ R^{-}_{1,B} \\ R^{-}_{2} \\ R^{-}_{3} \end{bmatrix} \qquad \begin{bmatrix} \Phi^{+}_{i} \\ \Phi^{-}_{i} \end{bmatrix} = \begin{bmatrix} \Phi_{1} \\ \Phi^{\text{ext}}_{2} \\ \Phi^{\text{ext}}_{3} \\ \Phi^{\text{ext}}_{1} \\ \Phi_{2} \\ \Phi_{3} \end{bmatrix} \tag{43}$$

These elemental contributions are used on the elements intersected by the wake, now with the constraint of the boundary conditions of the wake applied in the auxiliary degrees of freedom. In summary, the above approach allows to model a discontinuity on the potential field, while respecting the wake boundary conditions, without the need to explicitly model the wake in the mesh.

2.5. Wake in the embedded approach

Defining the wake and the airfoil as embedded geometries requires dealing with its intersection, as both are modelled implicitly with a level-set function. On one hand, the geometry is represented as a closed volume for some scalar field, ϕ. On the other hand, the wake is represented as a volume-less geometry expressed as well with some scalar field, ϕ_{wake}. In the body-fitted case, the trailing edge node can be safely used as origin to start the definition of the wake, as shown in Fig. 4, where the degrees of freedom are duplicated and illustrated with red dots for the main potential values and with blue squares for the external potential values. In the embedded case, the origin of the wake is not clearly defined, and a consistent method to model the wake is needed, which works for any possible combination of elements near the trailing edge. This issue is solved in [14] in a finite-volume approach by generating first the wake on top of the trailing edge of the initial STL geometry. Then, the volume cells are generated from this configuration, cutting (physically splitting) the cells that are intersected by the wake during mesh generation. In this paper, the aim will be to start from an arbitrary finite-element mesh, and do this operation without explicitly cutting the elements.

As introduced in Eq. (13), the solver equations require to determine which elements belong to the fluid, to the intersected geometry or to the wake. The geometry elements are found using the level set function ϕ. The wake elements can be equivalently identified with the wake level set function ϕ_{wake}. Fluid elements are regular Laplacian elements for which the distance to the skin is positive on all their nodes. There are some set of elements, $\Omega_{\text{kutta}} = \Omega_{in} \cap \Omega_{W}$, that represent the intersection of geometry and the wake in the trailing edge, and for which the Kutta condition has to be enforced. These elements and their neighbours are illustrated in green in Fig. 5, where the wake is represented by a blue dotted line. The wake origin is set at the same location as the trailing edge from the original geometry and prolonged into the airfoil, following the chord. This allows identifying the intersected elements inside the geometry.

The algorithm to determine the elements in Ω_{kutta} is shown in Algorithm 1. An elemental loop is performed looking for a sign swap occurring at the nodes of the elements for both ϕ and ϕ_{wake}. If the swap is found, elements are listed and identified, which will belong to the geometry or the wake. Elements intersected by both distance functions and its nodes are identified. In these elements, the description of the geometry is likely inaccurate, as the trailing edge is a sharp edge. This is illustrated in Fig. 5, where the actual representation of the continuous level set function used is shown with a black dotted line. Due to this, the splitting coming from the function ϕ is ignored for these elements, as the no-penetration condition in Eq. (24) would implicitly enforce an incorrect direction of the local velocity. Since these elements are intersected by the wake, its degrees of freedom are duplicated, as shown in Fig. 6. As done previously, the main and external degrees of freedom are represented by red dots and blue squares respectively. However, as these elements are also part of the geometry, the terms imposing the wake boundary conditions in Eqs. (18) and (19) are omitted and only the main Laplacian terms are integrated, together with the penalty term used to enforce the Kutta condition introduced in Eq. (20).

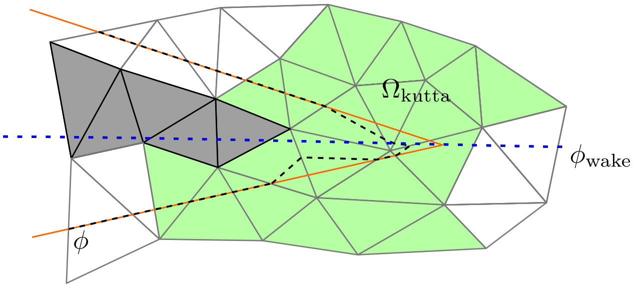


Fig. 5. Elements identified as Ω_{kutta} are illustrated in green, which are defined as the elements intersected by both the geometry and the wake, and its neighbours. The wake is represented by a dotted blue line and it is prolonged inside the airfoil, following the chord. Note that the shape of the airfoil, ideally represented by the orange line, is not correctly described in the mesh due to the sharp edge present in the trailing edge. This is exemplified with a dotted black line which shows the actual representation of the trailing edge performed by the mesh. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

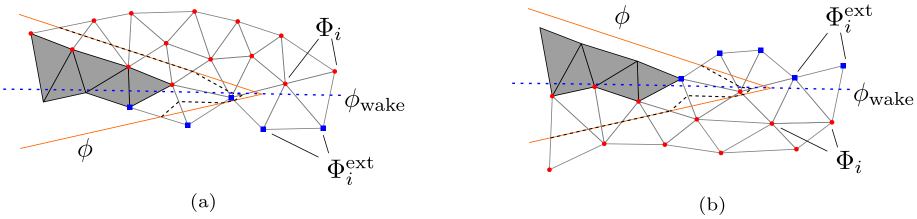


Fig. 6. Representation of an embedded wake with an embedded geometry. The wake is defined implicitly through a distance function, and its intersection subdivides the mesh into two, shown in Fig. 6a and 6b. Note that the duplication is only performed for visualization purposes, as only one mesh is utilized.

Algorithm 1: Algorithm to identify the Kutta elements Ω_{kutta}

```
for $\Omega^e \in \Omega$ do
    for $\Omega_i^e \in \Omega^e$ do
        $n_{W^+}$ += 1 if $\Omega_i^e \in \Omega_{W^+}$
        $n_{W^-}$ += 1 if $\Omega_i^e \in \Omega_{W^-}$
        $n_{\phi^+}$ += 1 if $\Omega_i^e \in \Omega_{in}$
        $n_{\phi^-}$ += 1 if $\Omega_i^e \in \Omega_{out}$
    end
    if $n_{W^+} \cdot n_{W^-} \cdot n_{\phi^+} \cdot n_{\phi^-} > 0$ then
        Set $\Omega^e \in \Omega_{\text{kutta}} = \Omega_\phi \cap \Omega_W$
    end
end
```

2.6. Penalty term to enforce Kutta condition

The so-called Kutta condition is defined in [6] as the fact that the velocity leaves smoothly the trailing edge. This condition is needed to find a unique solution in the potential flow equation, which gives the right amount of circulation in the solution. An infinite number of different values of circulation satisfy the potential flow equations, but the Kutta condition establishes the value that matches the behaviour that the flow physically follows. In a modelling sense, this can be defined as the velocity following the bisector line on the trailing edge, defined as $()_{TE}$. In order to do this, the following restriction is enforced in the flow:

$$\mathbf{n}_{\text{kutta}} \cdot \mathbf{v}_{TE} = \mathbf{n}_{\text{kutta}} \cdot (\nabla \Phi)_{TE} = 0 \tag{44}$$

where $\mathbf{n}_{\text{kutta}}$ is the normal to the bisector line of the trailing edge. In body-fitted meshes, the solid walls and the wake line near the trailing edge help to enforce this condition. In an embedded setting, one cannot rely on the no-penetration condition coming from the adjacent elements, as the shape of the trailing edge will not be properly defined in general. This is due to the fact that the trailing edge forms a sharp edge, which is a geometrical feature difficult to capture implicitly using a distance function. as illustrated in Fig. 5. For this reason, the equations above are used in conjunction with a penalty term, which ensures that the Kutta condition is satisfied even if the trailing edge is not perfectly described by ϕ. While local refinement and other techniques could be used to improve the description of sharp edges as presented in [19], a method that also yields a good enough solution for coarser meshes is desired.

Let $\psi(\Phi) = \mathbf{n}_{\text{kutta}} \cdot (\nabla \Phi)_{\text{TE}} = 0$ be a functional which represents the constraint in the trailing edge. The penalty method is written by building the functional $\Lambda(\Phi, k_{\text{kutta}})$ for some penalty k_{kutta}:

$$\Lambda(\Phi, k_{\text{kutta}}) := \frac{1}{2}\rho \left\|\nabla \Phi\right\|^2 - \frac{k_{\text{kutta}}}{2} \left\|\psi(\Phi)\right\|^2 \tag{45}$$

where Λ is minimized to find the update on Φ that complies with the constraint $\psi(\Phi)$ and the solution of the Laplacian problem. Deriving with respect to the potential and testing with the function ω, the original problem in Eq. (22) is recovered, as well as the term introduced in Eq. (20):

$$\frac{\partial \Lambda(\Phi, k)}{\partial \Phi} = \int_{\Omega_{\text{kutta}}} \rho \nabla \omega \cdot \nabla \Phi d\Omega - k_{\text{kutta}} \int_{\Omega_{\text{kutta}}} \nabla \omega \mathbf{n}_{\text{kutta}}^T \mathbf{n}_{\text{kutta}} \nabla \Phi d\Omega \tag{46}$$

After discretizing, the following terms represent the elemental contributions to the residual and Jacobians that are added to the elements on Ω_{kutta}:

$$R^e_{i,K}(\Phi, k_{\text{kutta}}) = -k_{\text{kutta}} \sum_{j=1}^{M} \int_{\Omega_{\text{kutta}}} \nabla N_i \mathbf{n}_{\text{kutta}}^T \mathbf{n}_{\text{kutta}} \nabla N_j d\Omega \Phi_j \tag{47}$$

$$J^e_{i,j,K}(\Phi, k_{\text{kutta}}) = -\frac{\partial R^e_{i,K}}{\partial \Phi} = k_{\text{kutta}} \int_{\Omega_{\text{kutta}}} \nabla N_i \mathbf{n}_{\text{kutta}}^T \mathbf{n}_{\text{kutta}} \nabla N_j d\Omega \tag{48}$$

Substituting with these terms the elemental contributions in Eqs. (47) and (48) on the elements Ω_{kutta} marked by Algorithm 1, ensures that the Kutta condition is satisfied and the correct pressure distribution is obtained in the trailing edge, as shown in Fig. 7. In general terms, the solution pressure distribution is correct without the penalty terms, but if the trailing edge is closely analysed, it can be seen that without penalty the pressure distribution on the embedded geometry does not match the body-fitted solution. Applying the penalty on the embedded solver, gives an improved solution that is clearly visible on the trailing edge, but also on the final value of the lift coefficient, that is closely dependent on the proper definition of the trailing edge flow.

2.7. Stabilization term for the embedded formulation

For linear elements, the gradient of the potential field shows some instabilities that can be attributed to the element splitting with the distance function. This is specially important in coarser meshes where the geometry representation is worse. The effect is appreciable in the pressure distribution plots, as the pressure depends directly on the potential gradient. In order to improve these instabilities, a stabilization term is added to the formulation to smooth the gradient of the potential. This term is added on the split elements, using the average of the gradients of the surrounding elements. This results in a smoothing of the pressure distribution curve. The added term was presented in Eq. (21) with some factor k_{stab}. This term corresponds to the difference of the gradient of the potential and the average of the nodal gradient on the surrounding elements. Thus, local jumps in the gradient of the potential are penalized. The average of the nodal gradient is expressed as:

$$\overline{\xi\left(\nabla \Phi\right)} = \frac{1}{n_{\phi^+}} \sum_{\phi_{i>0}} \xi\left(\nabla \Phi\right)_i \tag{49}$$

where n_{ϕ^+} is the number of nodes in the elements for which the distance function is positive $\phi > 0$. The term $\nabla \Phi^n_i$ represents the projected elemental gradient of the potential in the nodes. This projection can be estimated

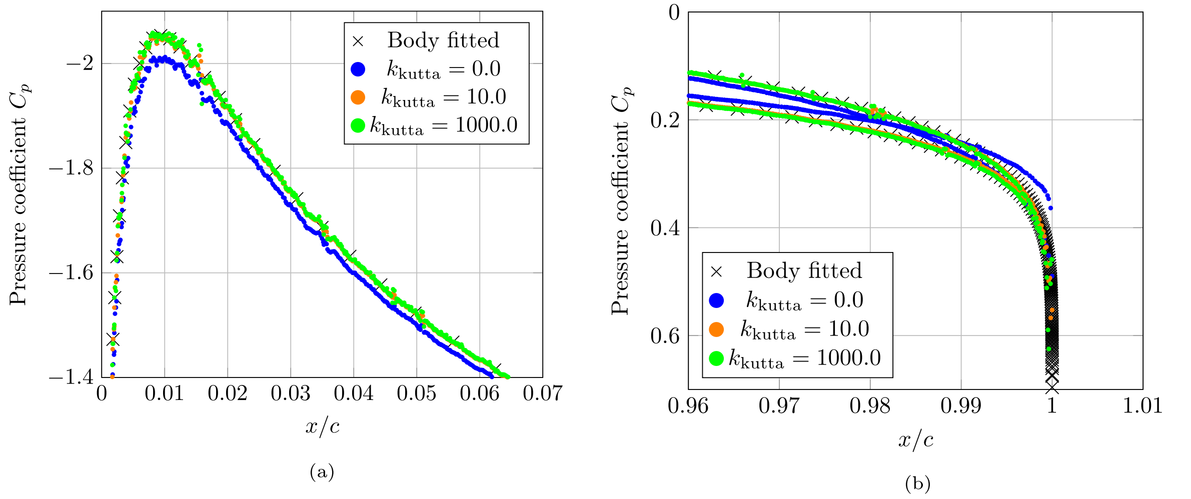


Fig. 7. Pressure distribution of the incompressible potential flow around a NACA0012 profile at 5° for different values of the penalty coefficient used to enforce the Kutta condition. Fig. 7a and 7b show a close up to the leading and trailing edge, respectively. Note that for the pressure distribution for no penalty, $k_{\text{kutta}} = 0$, the trailing edge is badly represented.

numerically as follows, where n_{elem} is the number of neighbouring elements for an element Ω^e:

$$\xi\left(\nabla\Phi\right)_i = \frac{\sum_1^{n_{elem}} \int_{\Omega^e} N_i \nabla N_j d\Omega \Phi_j}{\sum_1^{n_{elem}} \int_{\Omega^e} N_i d\Omega} \tag{50}$$

After discretization, the elemental contributions to the residual on the potential for the intersected terms R_ϕ is rewritten with the stabilization term:

$$R^e_{i,\phi}(\Phi) = \sum_{j=1}^{M} \int_{\Omega^e_{in}} \rho \nabla N_i \cdot \nabla N_j d\Omega \Phi_j + k_{\text{stab}} \left[\int_{\Omega^e} \nabla N_i \cdot \nabla N_j d\Omega \Phi_j - \overline{\xi\left(\nabla\Phi\right)} \right] \tag{51}$$

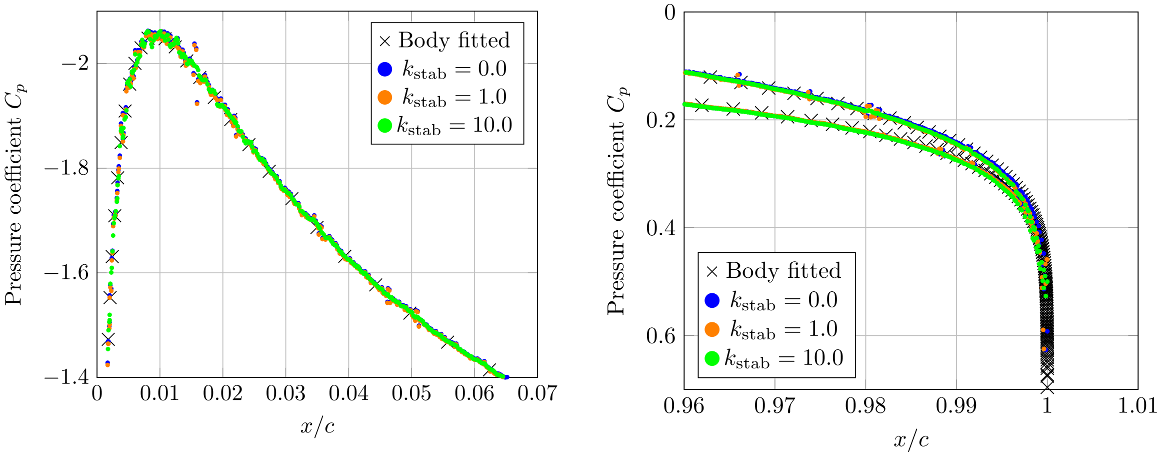
Thus, the new term penalizes big differences of the potential gradient of a given element with its neighbours, if its value is different than the neighbour average. The added term is non-linear on the potential, but this is not a major drawback since the formulation is already non-linear due to the compressibility hypothesis. The effect of this term is shown in Fig. 8, for the incompressible flow around a NACA0012 profile at an angle of attack of $\alpha = 5°$. Different pressure distributions are plotted for different values of the stabilization factor. For these plots, a penalty coefficient of $k_{\text{kutta}} = 1000$ was used. It can be seen that some pressure values are out of the reference outline for the embedded formulation with no stabilization term. Adding the stabilization term smooths these pressure jumps, accomplishing a better distribution in the embedded formulation that matches the body-fitted solution.

3. Adaptive refinement

Metric-based refinement has been employed to automatically adapt the meshes using the Mmg software [20], in order to solve the embedded potential flow equations. Among other advantages such as mesh quality or speed, a metric-based approach allows to freely adapt the remeshing operation depending on the user needs. The approach is based on the workflow presented by [21], which combines the metric definition from a level set function and Hessian based refinement [10]. This is exploited to use the information of the level set field describing the geometry ϕ and previous potential field solutions, as shown in Fig. 9 .

On one hand, an a priori (before solution) strategy is defined, which depends on the level set field defined by the geometry of study. For this purpose, the gradient of the level set function is computed, whose components are denoted by $\nabla\phi_i$. Then, the metric tensor computed at every node is defined as:

$$\mathcal{M}(x) = \begin{pmatrix} c_0\left(1 - \nabla\phi_x^2\right) + c_1\nabla\phi_x^2 & \left(c_1 - c_0\right)\nabla\phi_x\nabla\phi_y & \left(c_1 - c_0\right)\nabla\phi_x\nabla\phi_z \\ \left(c_1 - c_0\right)\nabla\phi_x\nabla\phi_y & c_0\left(1 - \nabla\phi_y^2\right) + c_1\nabla\phi_y^2 & \left(c_1 - c_0\right)\nabla\phi_y\nabla\phi_z \\ \left(c_1 - c_0\right)\nabla\phi_x\nabla\phi_z & \left(c_1 - c_0\right)\nabla\phi_y\nabla\phi_z & c_0\left(1 - \nabla\phi_z^2\right) + c_1\nabla\phi_z^2 \end{pmatrix} \tag{52}$$



(a)

(b)

Fig. 8. Pressure distribution of the incompressible potential flow around a NACA0012 profile at 5° for different values of the stabilization factor used to smooth the gradient of the potential. A penalty coefficient of $k_{\text{kutta}} = 1000$ was used in these figures for all curves. Fig. 8a shows the pressure distribution in the upper side of the trailing edge, while Fig. 8b shows a close up at the trailing edge.

where the coefficients c_0 and c_1 are defined for a user defined element size h and an anisotropic ratio r:

$$c_0 = \frac{1}{h^2} \tag{53}$$

$$c_1 = \frac{c_0}{r^2} \tag{54}$$

With this metric definition, the newly generated mesh can be controlled by choosing the size h as a function of ϕ. The size is controlled with:

$$h(\phi(x)) = h_{min} + \frac{\phi(x)}{L_b}\left(h_{max} - h_{min}\right) \quad \text{if} \quad \phi(x) < L_b \tag{55}$$

where a minimal size h_{min} is set for nodes at the zero-isosurface of the level set $\phi(x) = 0$, and a maximal size h_{max} is set for nodes at some user-defined isosurface at $\phi = L_b$. Values larger than this boundary isosurface take the element size of the initial mesh. On the other hand, a metric is defined using the solution of a previous mesh, which is written as:

$$\mathcal{M} = \mathcal{R}\hat{\Lambda}^t\mathcal{R} \qquad \text{where} \qquad \hat{\Lambda} = diag(\hat{\lambda}_i) \tag{56}$$

$$\hat{\lambda}_i = min\left(max\left(\frac{c_d|\lambda_i|}{\varepsilon}, h_{max}^{-2}\right), h_{min}^{-2}\right) \tag{57}$$

where λ_i are the eigenvalues and $\mathcal{R}$ the matrix of eigenvectors of the Hessian $\mathcal{H}_u$ of a given variable u. The eigenvalues are truncated for some user-defined minimal and maximal sizes, h_{min} and h_{max}. The metric depends on a constant c_d and the interpolation error ε. This error is defined as the error that is committed by discretizing the domain and representing the variable u as u_h:

$$\varepsilon = \|u - u^h\| \tag{58}$$

The Hessian is estimated numerically by using the shape function gradients of the elements and by performing a volume-based nodal projection twice, on a similar manner as in Eq. (50). For a given scalar quantity, the gradient of the shape functions can be used to compute the gradient on the Gauss points of the scalar quantity. This gradient can be computed at the nodes by performing an area-based average using the gradient of the Gauss points from the neighbouring elements. Performing the same operation again using each component of the computed nodal gradient will yield the final estimated Hessian matrix. This can be expressed for some scalar quantity u as:

$$\nabla u_i^n = \frac{\sum_1^{n_{elem}} \int_{\Omega^e} N_i \nabla N_j d\Omega u_j}{\sum_1^{n_{elem}} \int_{\Omega^e} N_i d\Omega} \tag{59}$$

Table 1
Parameters used and lift coefficient obtained after solving the compressible potential flow around a NACA0012 airfoil at an angle of attack of 2°. The lift coefficient obtained lies within a 1% error if compared to the reference value from [22].

Mesh	Refinement type	$h_{\min}$	h_{max}	L_b	ε	n_{nodes}
Initial mesh	None	5×10^{-2}	1.0	–	–	14 042
Intermediate mesh	Level set	1×10^{-3}	5×10^{-2}	0.1	–	23 005
Final mesh	Hessian	1×10^{-9}	1×10^{3}	–	5×10^{-3}	81 668

$$\mathcal{H}_{u,i,j} = \frac{\sum_1^{n_{elem}} \int_{\Omega^e} N_i \nabla N_k d\Omega \nabla u^n_{k,i}}{\sum_1^{n_{elem}} \int_{\Omega^e} N_i d\Omega} \tag{60}$$

The scalar variable u can be assigned as any quantity coming from the solver. For instance, the values of the potential Φ, or each component of the velocity $v_k = \nabla\Phi_k$. For the wake elements $\Omega^e \in \Omega_W$, the values of the potential used to estimate the nodal gradient in Eq. (59) are replaced by Φ^+. Either side of the wake can be used to compute the gradient on the wake elements. Here, the upper side, denoted with the $+$ sign, is arbitrarily chosen:

$$\nabla\Phi^n_i = \frac{\sum_1^{n_{elem}} \int_{\Omega^e} N_i \nabla N_j d\Omega \Phi^+_j}{\sum_1^{n_{elem}} \int_{\Omega^e} N_i d\Omega} \quad \text{if} \quad \Omega^e \in \Omega_{W+} \tag{61}$$

Using this metric allows refining adaptively the mesh while prescribing the interpolation error on the potential field to a fixed value. In practice, this generates more nodal presence on the leading and trailing edge. If this metric is combined with the level set metric, it allows to generate a highly accurate mesh with a reduced number of nodes.

4. Results

In order to validate the method proposed, a test case from [22] is selected for a NACA0012 airfoil with chord $c = 1$ m at an angle of attack $\alpha = 2°$ and free stream Mach $M_\infty = 0.63$. The far-field has been modelled with a 50×50 m squared mesh. The compressible potential flow for this case is solved starting from a generic unstructured mesh which is adaptively refined in two steps. Detailed information of the three meshes used is presented in Table 1.

The initial mesh is shown in Fig. 9a which is uniformly meshed with a classical pre-processor, using a bounding box at the airfoil location with an element size of $h_{min} = 5\times10^{-2}$. The outer domain is meshed with an element size $h_{max} = 1.0$. Note that this initial mesh could be reused for any geometry configuration.

Next, the metric in Eq. (52) is used to adaptively refine the mesh using the level set field, and the case is solved. The settings involved in this refinement are also reported in Table 1, where the refinement is performed prescribing the element sizes to $h_{min} = 1\times10^{-3}$ and $h_{max} = 5\times10^{-2}$. These sizes are imposed at the level set zero-isosurface and at the isosurface $\phi = L_b$, respectively. The sizes at the points lying in-between these isosurfaces are linearly interpolated according to Eq. (55). The mesh obtained in this refinement step is shown in Fig. 9b.

Each component of the velocity field obtained after solving the intermediate mesh is then used with the metric in Eq. (56) to adapt the mesh according to the solution, while keeping the geometrical information from the previous step by performing the metric intersection [11]. The metric intersection allows keeping the optimal sizes from both approaches. The settings used in this step are also reported in Table 1, which take part in Eq. (57) for this approach. The interpolation error is set to $\varepsilon = 5\times10^{-3}$, and the element size truncating values are deactivated so that the refinement is driven purely by the interpolation error. For that reason, the minimal and maximal sizes are prescribed to small and large values respectively in Table 1. This results in the computational mesh shown in Fig. 9c, which has been adaptively refined using both the geometrical information and the flow characteristics. A summary of the settings and computational information used in the case is presented in Table 1.

Although the final solution for this test case is subcritical, the case is close to transonic in some points of the domain which makes the Jacobian in Eq. (26) ill-defined. This is pointed out by [4], where a limit in the local Mach number and the isentropic ratio for the density is proposed to overcome this issue. For local Mach numbers exceeding $M_{\text{local}} = 0.99$, the Mach number is fixed to 0.99 and the density isentropic ratio is set to $\rho/\rho_\infty = 10^{-5}$. Some of the parameters and computational details used to solve this case are reported in Table 2. The problem is solved first on the intermediate mesh, whose solution is used to generate the final mesh. A penalty coefficient of $k_{\text{kutta}} = 10.0$ is selected to enforce the Kutta condition. For these meshes, a stabilization factor $k_{\text{stab}} = 1.0$ is used, for which the instabilities in the pressure distribution are unnoticeable. The number of degrees of freedom n_{dof} is

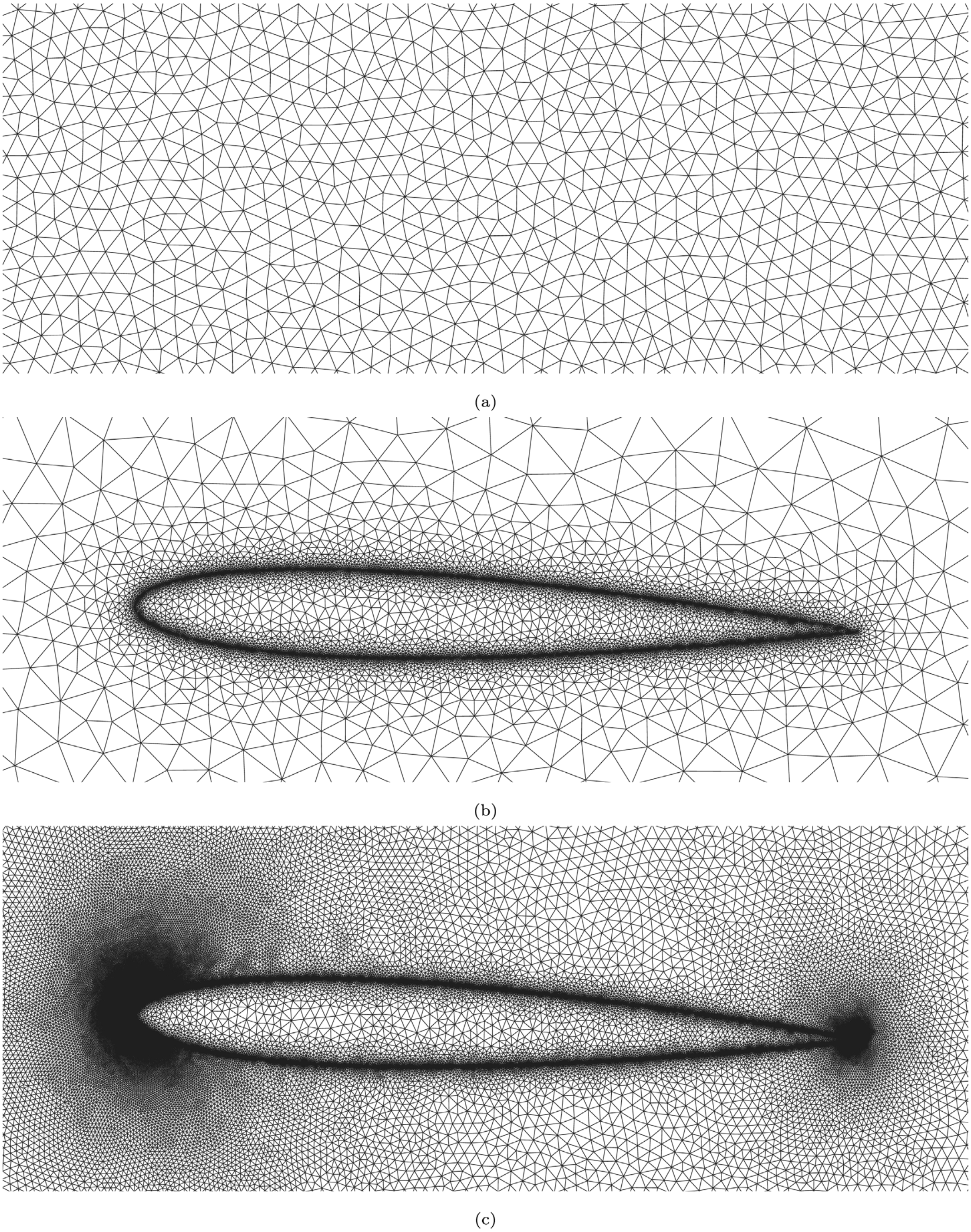


Fig. 9. Snapshot of the adaptively refined mesh in two steps. First, the level set function is used to adaptively refine the mesh in Fig. 9a according to the distance function defining the geometry of study, resulting in the mesh shown in Fig. 9b. After solving the problem in this mesh, another iteration is performed by adaptively refining the mesh according to the potential solution, obtaining the mesh in Fig. 9c.

Table 2
Computational details and results for the intermediate and final meshes used in the analysis to solve the compressible potential flow around a NACA0012 airfoil at an angle of attack of 2°. The same settings for the far-field are applied on both meshes. The final mesh is generated from the solution of the intermediate mesh by using a Hessian adaptive refinement. The lift coefficient obtained on the final mesh matches the reference value from [22].

Mesh	M_∞	α	k_{kutta}	k_{stab}	n_{dof}	C_l	C_l^{ref} [22]	n_{iter}	T_M	T_S
Intermediate mesh	0.63	2°	10.0	1.0	18 957	0.293	0.335	17	3 s	5 s
Final mesh	0.63	2°	10.0	1.0	71 741	0.335	0.335	13	7 s	13 s

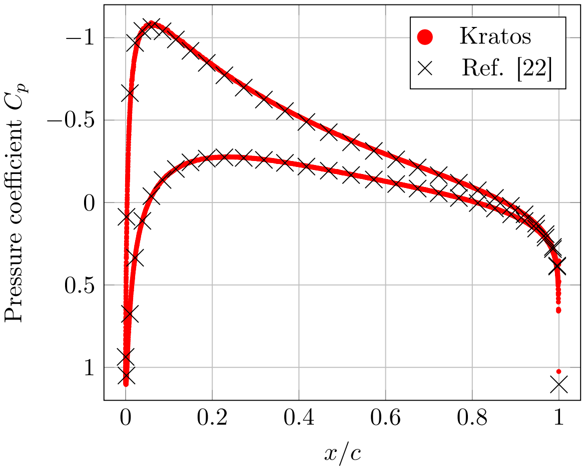


Fig. 10. Pressure distribution of a NACA0012 airfoil at an angle of attack $\alpha = 2°$ at $M_\infty = 0.63$, solved in an unfitted mesh. The result is compared to the values reported in [22].

lower than the number of nodes reported in Table 1 for two reasons. First, the elements intersected by the wake have duplicated degrees of freedom. Second, the elements completely immersed by the level set are deactivated, and the degrees of freedom of their nodes do not contribute to the system. For this reason, the total number of degrees of freedom is lower than the number of nodes. The number of non-linear iterations n_{iter} of the Newton–Raphson strategy and the time T_S spent solving the system is also reported in Table 2. Both cases are solved on the four cores of a 3.6 GHz Intel Core i7-4790 CPU with OpenMP parallelization with an algebraic multigrid linear solver [23]. The modelling time T_M is reported separately from the linear solver time, as it is the time that includes both the remeshing and the level set calculation operations, which are not performed in classical body-fitted analyses. This can be compared to the time that the user spends creating and generating the body-fitted mesh, which is done automatically in this approach. Most of the modelling time is spent remeshing, which is a serial operation that does not benefit from OpenMP parallelization. The final pressure distribution obtained on this mesh is shown in Fig. 10, matching the reference solution in [22].

5. Conclusions

A method to solve the full potential equation on a fully embedded approach has been presented. The method allows defining both the geometry and the wake implicitly, reducing considerably the preprocessing effort from the user. Adaptive mesh refinement can be employed to improve a generic background mesh to the specific case, allowing for an accurate description of the geometry and the flow. The formulation depends on two hyper-parameters, but the solver performs well for a wide range of values. The accuracy accomplished depends on the adaptive refinement settings selected by the user, which allows trading off accuracy for speed. Although simpler methods exist to compute the potential flow for streamlined bodies, the automatic nature of the immersed approach makes it a perfect choice for optimization analyses. In this scenario, the number of solver evaluations is generally high, and the geometry is likely changing at every step. The embedded approach ensures that the solver evaluations can be performed on the same mesh without mesh deformation, which can be adapted for every new geometry configuration.

The method and results presented in this document have been verified for the two-dimensional analysis of airfoil shapes. The extension of the embedded solver to three-dimensional problems is currently under development. The methodology described here to embed bodies can be directly extended to three dimensions, but certain care needs to be taken into account regarding the imposition of the wake boundary conditions. In the two-dimensional case, Eq. (30) simplifies both Eq. (8) and (9) into a single equation as this is equivalent for vectors in two dimensions. Instead, in three dimensions Eqs. (8) and (9) need to be imposed separately. Here, Eq. (9) can be linearized using the freestream velocity as presented by [4]. Also, the wake is modelled as a plane instead of a line, which intersects the trailing edge line of the wing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work has been supported by the European Commission through the Horizon 2020 Research and Innovation program under contract 800898 (ExaQUte project). The authors also acknowledge financial support from the Spanish Ministry of Economy and Competitiveness, through the “Severo Ochoa Programme for Centres of Excellence in R&D” (CEX2018-000797-S).

References

[1] S. Deinert, Shape and Sizing Optimization of Aircraft Structures with Aeroelastic and Induced Drag Requirements (Ph.D. thesis), Technische Universität München, 2016.

[2] T.L. Holst, W.F. Ballhaus, Fast, conservative schemes for the full potential equation applied to transonic flows, AIAA J. 17 (2) (1979) 145–152, http://dx.doi.org/10.2514/3.61088.

[3] T.L. Holst, Transonic flow computations using nonlinear potential methods, Prog. Aerosp. Sci. 36 (1) (2000) 1–61, http://dx.doi.org/10.1016/S0376-0421(99)00010-X.

[4] D. Eller, Fast, unstructured-mesh finite-element method for nonlinear subsonic flow, J. Aircr. 49 (5) (2012) 1471–1479, http://dx.doi.org/10.2514/1.c031738.

[5] B. Nishida, Fully simultaneous coupling of the full potential equation and the integral boundary layer equations in three dimensions, 1996.

[6] J. Katz, A. Plotkin, Low-Speed Aerodynamics, Cambridge University Press, 2001, http://dx.doi.org/10.1017/CBO9780511810329, URL https://www.cambridge.org/core/product/identifier/9780511810329/type/book.

[7] B. Nishida, M. Drela, Fully simultaneous coupling for three-dimensional viscous/inviscid flows, in: 13th Applied Aerodynamics Conference, American Institute of Aeronautics and Astronautics Inc, AIAA, 1995, pp. 355–361, http://dx.doi.org/10.2514/6.1995-1806, URL https://arc.aiaa.org/doi/abs/10.2514/6.1995-1806.

[8] M. Davari, R. Rossi, P. Dadvand, I. López, R. Wüchner, A cut finite element method for the solution of the full-potential equation with an embedded wake, Comput. Mech. 63 (5) (2019) 821–833, http://dx.doi.org/10.1007/s00466-018-1624-3.

[9] E. Burman, S. Claus, P. Hansbo, M.G. Larson, A. Massing, Cutfem: Discretizing geometry and partial differential equations, Internat. J. Numer. Methods Engrg. 104 (7) (2015) 472–501, http://dx.doi.org/10.1002/nme.4823, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.4823.

[10] P. Frey, F. Alauzet, Anisotropic mesh adaptation for CFD computations, Comput. Methods Appl. Mech. Engrg. 194 (48–49) (2005) 5068–5082, http://dx.doi.org/10.1016/j.cma.2004.11.025.

[11] C. Bui, C. Dapogny, P. Frey, An accurate anisotropic adaptation method for solving the level set advection equation, Internat. J. Numer. Methods Fluids 70 (7) (2012) 899–922, http://dx.doi.org/10.1002/fld.2730.

[12] M.C. Galbraith, S.R. Allmaras, R. Haimes, Full potential revisited: A medium fidelity aerodynamic analysis tool, in: AIAA SciTech Forum - 55th AIAA Aerospace Sciences Meeting, American Institute of Aeronautics and Astronautics Inc., Reston, Virginia, 2017, http://dx.doi.org/10.2514/6.2017-0290, URL http://arc.aiaa.org/doi/10.2514/6.2017-0290.

[13] A. Crovato, R. Boman, H. Guner, V.E. Terrapon, G. Dimitriadis, H.S. Almeida, A.P. Prado, C. Breviglieri, P.H. Cabral, G.H. Silva, A full potential static aeroelastic solver for preliminary aircraft design, in: International Forum on Aeroelasticity and Structural Dynamics 2019, IFASD 2019, 2019.

[14] F. Lyu, T. Xiao, X. Yu, A fast and automatic full-potential finite volume solver on cartesian grids for unconventional configurations, Chin. J. Aeronaut. 30 (3) (2017) 951–963, http://dx.doi.org/10.1016/j.cja.2017.03.001.

[15] R. Abgrall, H. Beaugendre, C. Dobrzynski, An immersed boundary method using unstructured anisotropic mesh adaptation combined with level-sets and penalization techniques, J. Comput. Phys. 257 (PA) (2014) 83–101, http://dx.doi.org/10.1016/j.jcp.2013.08.052, URL www.elsevier.com/locate/jcp.

[16] P. Dadvand, R. Rossi, E. Oñate, An object-oriented environment for developing finite element codes for multi-disciplinary applications, Arch. Comput. Methods Eng. 17 (3) (2010) 253–297, http://dx.doi.org/10.1007/s11831-010-9045-2, URL https://link.springer.com/article/10.1007/s11831-010-9045-2.

[17] P. Dadvand, R. Rossi, M. Gil, X. Martorell, J. Cotela, E. Juanpere, S. Idelsohn, E. Oñate, Migration of a generic multi-physics framework to HPC environments, Comput. & Fluids 80 (1) (2013) 301–309, http://dx.doi.org/10.1016/j.compfluid.2012.02.004.

[18] J. Anderson, Fundamentals of aerodynamics, 1984, http://dx.doi.org/10.2514/152157.

[19] D. Baumgärtner, J. Wolf, R. Rossi, P. Dadvand, R. Wüchner, A robust algorithm for implicit description of immersed geometries within a background mesh, in: Advanced Modeling and Simulation in Engineering Sciences, vol. 5, Springer International Publishing, 2018, http://dx.doi.org/10.1186/s40323-018-0113-8.

[20] C. Dapogny, C. Dobrzynski, P. Frey, Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems, J. Comput. Phys. 262 (2014) 358–378, http://dx.doi.org/10.1016/j.jcp.2014.01.005.

[21] V. Mataix, in: U.P. de Cataluña (Ed.), Innovative mathematical and numerical models for studying the deformation of shells during industrial forming processes with the finite element method (Ph.D. thesis), Universitat Politècnica de Catalunya, 2020, URL http://hdl.handle.net/2117/328952.

[22] R. Lock, Test cases for numerical methods in two-dimensional transonic flows, AGARD (1970).

[23] D. Demidov, AMGCL —A C++ library for efficient solution of large sparse linear systems, Softw. Impacts 6 (2020) 100037, http://dx.doi.org/10.1016/J.SIMPA.2020.100037.