HAL
open science

An interpolation-free ALE scheme for unsteady inviscid flows computations with large boundary displacements over three-dimensional adaptive grids

Barbara Re, Cecile Dobrzynski, Alberto Guardone

To cite this version:

Barbara Re, Cecile Dobrzynski, Alberto Guardone. An interpolation-free ALE scheme for unsteady inviscid flows computations with large boundary displacements over three-dimensional adaptive grids. Journal of Computational Physics, 2017, 340, pp.26 - 54. 10.1016/j.jcp.2017.03.034 . hal-01633476

HAL Id: hal-01633476

https://inria.hal.science/hal-01633476

Submitted on 8 Jan 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Journal of Computational Physics 340 (2017) 26–54

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

An interpolation-free ALE scheme for unsteady inviscid flows computations with large boundary displacements over three-dimensional adaptive grids

B. Re [a,*], C. Dobrzynski [b,c], A. Guardone [a]

[a] Department of Aerospace Science and Technology, Politecnico di Milano, 20156 Milano, Italy
[b] Bordeaux INP, IMB, UMR 5251, F-33400, Talence, France
[c] Inria Bordeaux Sud-Ouest, Team CARDAMOM, F-33405 Talence, France

ARTICLE INFO

Article history:
Received 3 October 2016
Received in revised form 12 March 2017
Accepted 15 March 2017
Available online 6 April 2017

Keywords:
Mesh adaptation
ALE scheme
Finite volume discretization
Tetrahedral grid
Euler equations

ABSTRACT

A novel strategy to solve the finite volume discretization of the unsteady Euler equations within the Arbitrary Lagrangian–Eulerian framework over tetrahedral adaptive grids is proposed. The volume changes due to local mesh adaptation are treated as continuous deformations of the finite volumes and they are taken into account by adding fictitious numerical fluxes to the governing equation. This peculiar interpretation enables to avoid any explicit interpolation of the solution between different grids and to compute grid velocities so that the Geometric Conservation Law is automatically fulfilled also for connectivity changes. The solution on the new grid is obtained through standard ALE techniques, thus preserving the underlying scheme properties, such as conservativeness, stability and monotonicity. The adaptation procedure includes node insertion, node deletion, edge swapping and points relocation and it is exploited both to enhance grid quality after the boundary movement and to modify the grid spacing to increase solution accuracy. The presented approach is assessed by three-dimensional simulations of steady and unsteady flow fields. The capability of dealing with large boundary displacements is demonstrated by computing the flow around the translating infinite- and finite-span NACA 0012 wing moving through the domain at the flight speed. The proposed adaptive scheme is applied also to the simulation of a pitching infinite-span wing, where the bi-dimensional character of the flow is well reproduced despite the three-dimensional unstructured grid. Finally, the scheme is exploited in a piston-induced shock-tube problem to take into account simultaneously the large deformation of the domain and the shock wave. In all tests, mesh adaptation plays a crucial role.
© 2017 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

The numerical simulation of unsteady fluid dynamics and continuum mechanics problems often requires to deal with large deformations of the computational domain and, at the same time, to precisely represent the boundaries and/or the interfaces of multi-material systems. The capability to cope with both requirements is determined by the choice of the kinematic description, which expresses the relationship between the deforming continuum and the computational grid. The

* Corresponding author.
E-mail address: barbara.re@polimi.it (B. Re).

http://dx.doi.org/10.1016/j.jcp.2017.03.034
0021-9991/© 2017 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Arbitrary Lagrangian–Eulerian (ALE) method [1] is a popular framework used to tackle moving boundary problems. Within this formulation, the mesh is neither fixed in space (as in the Eulerian viewpoint) nor attached to the material particle (as in the Lagrangian one) but it moves independently so e.g. to follow the domain boundary or interfaces. The governing equations are therefore enforced over deforming control volumes. This feature allows both to accurately track boundaries and interfaces and to easily deal with distortions of the continuum domain, capabilities that make ALE approach widely used in fluid–structure interaction problems [2–8], in multi-material simulations [9–13] and in shock hydrodynamics [14–16].

The key points of standard ALE algorithms with fixed grid connectivity consist in a classical Lagrangian step in which the grid is moved to follow the continuum, a re-zone phase during which the grid points are displaced to enhance quality and the re-mapping phase wherein the solution is advected or interpolated from the old to the enhanced grid. Several contributions are available in literature about each single phases. For instance, a description of the most popular approaches for re-zoning can be found in [17,7] and references therein. For what concerns the re-map step, the solution interpolation must be conservative, accurate and must preserve monotonicity of the solution. A very effective approach that satisfies all this requirements is based on the integration of the volume swept by the cells during the re-zone phase, followed by a repair step in which mass is re-distributed between neighboring cells to ensure local bound preservation [18–20]. Other techniques exploit the Flux-Corrected Transport method to enforce monotonicity, by combining low-order fluxes with high-order ones [21,14,22].

In addition to algorithms that implement separately the three ALE phases, schemes that do not perform explicitly the re-map phase but embed it into the solution update are also available [2,23–25,15,26]. In this case, the governing equations are solved directly within the ALE framework, by taking into account the motion of the control volumes. Also high-order (more than second) schemes have been proposed, as for example [27,28].

Although it can be obtained quite straightforwardly from the standard Eulerian formulation, the ALE formulation includes some time-dependent geometric quantities involving positions and velocities of the grid nodes that have to be properly evaluated. To this end, the Geometric Conservation Law (GCL) and its discrete counterpart—the Discrete Geometric Conservation Law (DGCL) [3]—play a crucial role. This consistency condition states that a numerical scheme designed to enforce the governing equations over a moving grid should preserve a uniform flow. The GCL can be enforced by an appropriate computation of the geometric quantities. Fulfilling the DGCL is a sufficient condition to be at least first-order time-accurate [29], although it is neither a necessary nor a sufficient condition for high-order accuracy [23]. On the other hand, its fulfillment is a necessary and sufficient condition for preserving non-linear stability [5]. Therefore, it is generally accepted that satisfying the DCGL improves time accuracy, numerical stability and avoids spurious oscillations [30,24]. An updated and comprehensive review of this subject can be found in [31]. Note that the scheme conservativeness is of paramount importance in fluid–structure interaction and aeroelastic problems, where violating the DGCL may lead to an erroneous computation of the flutter velocity [3].

Most of the standard ALE schemes use fixed mesh topology, namely the re-zoning phase aims at modifying only the position of the grid nodes without changing the connectivity. This constraint results in a limitation on the deformation that the grid is able to tackle, because large displacements of the nodes may lead to entangled, and thus invalid, elements. Moreover, since the grid nodes are moved from one grid region to another, only a limited control over the mesh resolution can be generally achieved [32]. To overcome this drawback, some adaptive re-zone approaches have been proposed. Adaptive Mesh Refinement within the ALE framework has been used to insert (or delete) grid nodes in structured grids by Anderson et al. [33] and by Morrell et al. [34], but both approaches are based on an underlying fixed connectivity, because only the cell of the initial grid can be sub-divided [32]. In order to avoid mesh entanglement, the generation of a new dual mesh at each time step is proposed by Springel [35]. This techniques relies upon the fact that the nodes of the Voronoi mesh generated from a moving Delaunay triangulation experience always a continuous deformation, even if the mesh-generating points move discontinuously. Alternatively, local mesh adaptation techniques [36–39] may be exploited to enhance grid quality after mesh deformation and to improve solution accuracy in moving-boundary problems, as for instance in [40–43]. Unfortunately, this operation requires a non-trivial interpolation of the solution from the old grid to the new one, that may originate problems in enforcing conservativeness and monotonicity and complicate the implementation of multi-step time integration algorithms [44].

The extension of ALE schemes variable topology grids has been investigated instead in [32,45], where the edge swapping techniques has been exploited to modify the grid without changing the number of grid nodes. Furthermore, several techniques, see for instance [46,47] for 2D and [48–51] for 3D, propose to treat grid connectivity changes within the ALE framework thanks to an explicit interpolation of the solution from the old to the new grid, which however may leads to the same drawbacks highlighted before. An alternative approach, available only for 2D problems to authors' knowledge, consists in the so-called space–time meshes that allow to handle large boundary movements within the ALE formulation by treating the time as the third spatial dimension [52,53]. Finally, the recent work of Barral and co-workers [54] proposes an ALE formulation for moving boundary problems able to deal with connectivity changes due to mesh adaptation thanks to the interpolation scheme described in [55], which, however, is conservative only if the volume of the domain does not change, i.e. if its boundaries do not match between consecutive time steps an error in the global mass conservation is introduced.

A different approach has been recently proposed by Isola and co-workers [56,57], who developed a novel node-centered finite-volume ALE scheme for fully-adaptive bi-dimensional grids. Changes in both the grid connectivity and the number of nodes due to edge-swapping, grid refinement and coarsening are allowed. This strategy is here extended to adaptive three-dimensional grids. Within the proposed approach, the insertion or deletion of a new node is interpreted as a series of

fictitious continuous deformations of the finite volumes associated to the nodes involved in the modification. The additional deformations due to grid adaptation result in additional fictitious ALE fluxes, which can be easily taken into account by standard ALE techniques, without requiring any explicit interpolation of the solution over the new grid. Admittedly, as observed in [56], the ALE mapping is indeed equivalent to an interpolation, but it does not require any special treatment to guarantee the conservativeness, the monotonicity and accuracy of the scheme. Moreover, also geometric quantities involved in this fictitious deformation are evaluated so that the DGCL is automatically enforced even in presence of topology changes. No special treatments or interpolations are required to implement multi-step time-integration techniques.

A challenging aspect regarding the extension of the interpolation-free ALE scheme in [56,57] to unstructured three-dimensional meshes concerns the variable local topology of the tetrahedral grids, which requires one to handle a variable number of elements involved for the local grid modification (with the exception of the split of a single element in four). Consider for example a node insertion by edge split. In a triangular grid, only two elements share an edge, so when the edge is split only these two elements have to be modified. Conversely, in tetrahedral grid, an edge can be shared by an arbitrary number of elements, so an arbitrary number of finite volumes have to be included in the series of fictitious continuous deformations used to treat conservatively the connectivity change within the ALE framework. Furthermore, the three-dimensional version introduced in this work is capable to deal with a broader suite of adaptation procedures than the one described in [56], including node insertion by the Delaunay triangulation. This technique consists in re-generating a local portion of the triangulation and new elements inserted in this way show usually better qualities than the ones created through edge split [58]. In order to effectively deal with the different aspects that characterize mesh adaptation in three dimensions, as for instance also the multiple options that are available for an edge swapping, the external re-mesher Mmg3d [59] is exploited in the present work.

The paper is organized as follows. First, in Section 2 the finite-volume edge-base ALE scheme for dynamic grids with fixed connectivity is presented to introduce the relevant concepts and the notation. The evaluation of some key quantities is explained in Subsections 2.2 and 2.4. Section 3 describes the modification required for its extension to adaptive grid. In particular, Subsection 3.1 details the three-steps procedure that enables the interpretation of the grid connectivity changes as series of fictitious continuous deformations for a tetrahedral grid. Then, a brief outline of the mesh adaptation techniques used in the numerical experiments is given Section 4. Afterwards, the results obtained with the proposed approach are presented in Section 5. Subsections 5.4 and 5.3 illustrate the results for a flow around an infinite-span wing traveling through the domain at the flight velocity for one chord and experiencing a pitching motion, respectively. In Subsection 5.4, the experiment of a wing translating through the domain is presented for a finite-span wing. Subsections 5.5 and 5.6 show the results obtained in tests characterized by large boundary displacements. Finally, the conclusions of the present work are drawn in Section 6.

2. Finite-volume ALE scheme for dynamic meshes

The Euler equations describe the behavior of compressible inviscid fluid flows and for sufficiently high Reynolds number flows around aerodynamic bodies they provide an accurate representation, except in the thin, boundary layer regions close to solid walls. In the ALE framework for a control volume $\mathcal{C}(t) \in \Omega$ contained in the domain $\Omega(t)$ and moving at velocity $\boldsymbol{v}(\boldsymbol{x}, t)$, the Euler equations read

$$\frac{d}{dt}\int_{\mathcal{C}(t)} \mathsf{u}\, d\boldsymbol{x} + \oint_{\partial\mathcal{C}(t)} [\mathbf{f}(\mathsf{u}) - \mathsf{u}\boldsymbol{v}] \cdot \boldsymbol{n}\, d\boldsymbol{s} = 0, \tag{1}$$

where $\boldsymbol{x} \in \Omega \subseteq \mathbb{R}^3$ is the position vector, $t \in \mathbb{R}^+$ is the time, u is the vector of conservative variables, $\mathbf{f}(\mathsf{u})$ is the inviscid flux and $\boldsymbol{n}(\boldsymbol{s}, t) \in \mathbb{R}^3$ is the outward unit vector normal to the boundary $\partial\mathcal{C}$ of the control volume $\mathcal{C}$, which is function of the boundary coordinate $\boldsymbol{s} \in \partial\Omega \subseteq \mathbb{R}^2$ and of time. The vector of conservative variables $\mathsf{u} : \Omega \times \mathbb{R}^+ \to \mathbb{R}^5$ is defined as $\mathsf{u} = [\rho, \boldsymbol{m}, E^{\mathsf{t}}]^{\mathsf{T}}$, with ρ the density, $\boldsymbol{m}$ the momentum density and E^{t} the total (internal plus kinetic) energy density, while the flux function $\mathbf{f}$ is defined as

$$\mathbf{f}(\mathsf{u}) = \left[\boldsymbol{m}, \quad \boldsymbol{m} \otimes \boldsymbol{m}/\rho + \Pi(\mathsf{u})\mathbb{I}^3, \quad \left(E^{\mathsf{t}} + \Pi(\mathsf{u})\right)\boldsymbol{m}/\rho\right]^T \tag{2}$$

where $\Pi(\mathsf{u}) = \Pi(\rho, \boldsymbol{m}, E^{\mathsf{t}})$ is the pressure function, which depends on the thermodynamic model, and $\mathbb{I}^3$ is the 3×3 identity matrix. In the present work, the polytropic ideal gas model is adopted to describe the behavior of the fluid. To make the governing equations (1) complete, suitable initial and boundary conditions must be specified [60].

As anticipated in Section 1, when solving the governing equations over moving grids, the fulfillment of the GCL is beneficial to avoid spurious oscillations and preserve non-linear stability [3,5,30,24]. To this end, the ALE scheme has to compute exactly the trivial solution of a uniform flow over a moving domain. Applying this requirement to Equations (1) leads to the following GCL:

$$\frac{d}{dt}\int_{\mathcal{C}(t)} d\boldsymbol{x} = \oint_{\partial\mathcal{C}(t)} \boldsymbol{v} \cdot \boldsymbol{n}\, d\boldsymbol{s}, \tag{3}$$

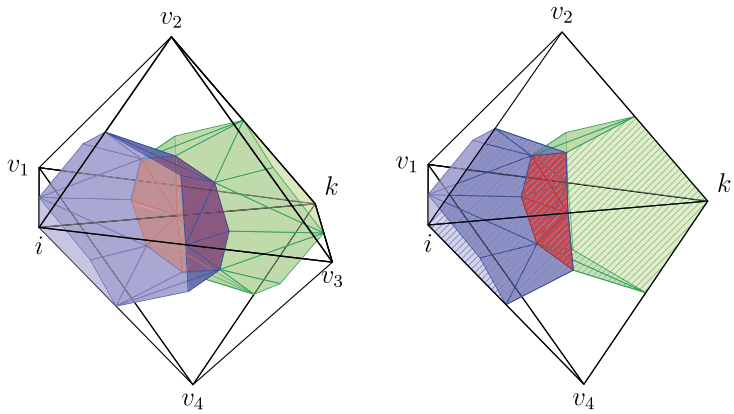

Fig. 1. Cell interface definition between two tetrahedral elements. The portion of the finite volumes surrounding the nodes i and k associated to the four tetrahedral elements defined by the grid nodes i, k, v_1, v_2, v_3, v_4 are shown. In the left picture, domain grid elements are considered, while in the right one the faces i–v_2–k and i–k–v_4 lie on the boundary $\partial\Omega$. Grid edges are drawn by black lines. The portions of $\mathcal{C}_i$ and $\mathcal{C}_k$ are painted respectively blue and green, while their boundaries $\partial\mathcal{C}_i$ and $\partial\mathcal{C}_k$ are darker, i.e. and . The red color identifies the cell interface $\partial\mathcal{C}_{ik} = \partial\mathcal{C}_i \cap \partial\mathcal{C}_k$, which is shown entirely. For the boundary case (at right), the patterns and indicate the boundary interfaces $\partial\mathcal{C}_i \cap \partial\Omega$ and $\partial\mathcal{C}_k \cap \partial\Omega$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

which can be viewed as a supplementary constraint to the flow equations (1) that the grid velocity has to fulfill during the motion.

2.1. Finite volume discretization within the node-pair representation

The spatially-discrete counterpart of the Euler equations (1) over a dynamic unstructured grid is obtained here through the finite volume formulation. A thorough description of this process can be found in [57,56]; here only the main steps are briefly recalled to introduce the nomenclature used in the following.

The domain is split into N_V non-overlapping volumes and a node-centered approach is adopted. As shown in Fig. 1, the control volume $\mathcal{C}_i(t)$ that surrounds the node i is composed by a portion of all grid elements sharing the node i. The portion of the boundary that it shares with its neighboring volume k, i.e. $\partial\mathcal{C}_{ik}(t) = \partial\mathcal{C}_i \cap \partial\mathcal{C}_k$, is usually referred to as cell interface and it is associated to the node-pair ik, that is the couple of interacting nodes i and k [61]. Then, given a general approximation Φ of the integral of the numerical flux across this interface, and a similar Φ^∂ for the eventual flux across the portion of $\partial\mathcal{C}_i$ that lies on the boundary, the node-pair representation of the Euler equations for each finite volume reads

$$\frac{d}{dt}[V_i \mathsf{u}_i] = \sum_{k\in\mathcal{K}_{i,\neq}} \Phi(\mathsf{u}_i, \mathsf{u}_k, \nu_{ik}, \hat{\boldsymbol{\eta}}_{ik}, \eta_{ik}) + \Phi^\partial(\mathsf{u}_i, \nu_i, \hat{\boldsymbol{\xi}}_i, \xi_i) \qquad \forall i \in \mathcal{K} \tag{4}$$

where V_i is the cell volume, u_i is the cell-averaged value of the solution, $\mathcal{K}$ is the set of all finite volumes and $\mathcal{K}_{i,\neq}$ is the subset of those finite volumes sharing a portion of their cell interface with i, i.e. $\mathcal{K}_{i,\neq} = \{k \in \mathcal{K}, k \neq i \mid \partial\mathcal{C}_i \cap \partial\mathcal{C}_k \neq \emptyset\}$. Moreover, the metric quantities $\boldsymbol{\eta}_{ik}$, ν_{ik}, $\boldsymbol{\xi}_i$, ν_i have been introduced [62]. They are the integrated normal $(\boldsymbol{\eta}_{ik}, \boldsymbol{\xi}_i)$ and the interface velocities (ν_{ik}, ν_i) of the interface $\partial\mathcal{C}_{ik}$ and of the boundary interface associated to the node i, respectively, and are defined as

$$\boldsymbol{\eta}_{ik} = \int_{\partial\mathcal{C}_{ik}} \boldsymbol{n}_i \, d\boldsymbol{x}, \qquad \nu_{ik} = \int_{\partial\mathcal{C}_{ik}} \boldsymbol{v}\cdot\boldsymbol{n}_i \, d\boldsymbol{x}, \qquad \boldsymbol{\xi}_i = \int_{\partial\mathcal{C}_i\cap\partial\Omega} \boldsymbol{n}_i \, d\boldsymbol{x} \qquad \nu_i = \int_{\partial\mathcal{C}_i\cap\partial\Omega} \boldsymbol{v}\cdot\boldsymbol{n}_i \, d\boldsymbol{x} \tag{5}$$

and, according to the standard vector notation, $\eta_{ik} = |\boldsymbol{\eta}_{ik}|$, $\hat{\boldsymbol{\eta}}_{ik} = \boldsymbol{\eta}_{ik}/\eta_{ik}$, $\xi_i = |\boldsymbol{\xi}_i|$ and $\hat{\boldsymbol{\xi}}_i = \boldsymbol{\xi}_i/\xi_i$. Although it is not explicitly written to lighten the notation, metric quantities depend on time as well as the cell interfaces because of grid points motion. Their expressions in terms of grid nodes position are given in Subsection 2.2.

In this work, the Total Variation Diminishing (TVD) approach is used to obtained the high-resolution expression for the integrated numerical domain fluxes [60,63]. The van Leer limiter [64] is used to switch from the second-order centered approximation to the first-order upwind scheme of Roe [65] near discontinuities. Boundary conditions are enforced in a weak form, i.e. by evaluating the boundary flux Φ^∂ in a suitable boundary state [56].

The same spatial discretization is applied now to the GCL condition (3), i.e.

$$\frac{dV_i(t)}{dt} = \int_{\partial\mathcal{C}_i(t)} \boldsymbol{v}\cdot\boldsymbol{n}_i \, d\boldsymbol{x} = \sum_{k\in\mathcal{K}_{i,\neq}} \nu_{ik}(t) + \nu_i(t) \qquad \forall i \in \mathcal{K}. \tag{6}$$

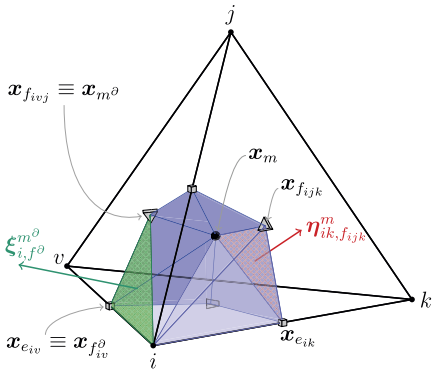

Fig. 2. Metrics definition in node-pair representation for an element with a face on the boundary. The six triangular facets defining the domain contribution of the considered element m to the $\partial\mathcal{C}_i$ are depicted in blue ■. Each interface portion has as vertices the barycenter of the element ●, of a face ◁ and of an edge ◻. The red pattern ※ highlights the triangular facet $\partial\mathcal{C}^m_{ik,f}$, related to the face i–j–k. Considering that the face i–v–j lies on the boundary and it forms the boundary element m^∂, the green pattern ⋇ depicts the boundary elemental portion over which the elemental contribution $\boldsymbol{\xi}^{m^\partial}_{i,f^\partial}$ is computed, where the considered face f^∂ is the edge i–v. Finally, the volume $V^m_{ik,f}$ of the sub-tetrahedron that has $\partial\mathcal{C}^m_{ik,f}$ as base and the node i as opposite vertex is shown in light blue ■. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The previous equation can be automatically satisfied by splitting the derivative of the cell volume into the volume swept by the interface $\partial\mathcal{C}_{ik}$ and $\partial\mathcal{C}_i \cap \partial\Omega$, respectively, namely $\frac{dV_i(t)}{dt} = \sum_{\mathcal{K}_{i,\neq}} \frac{dV_{ik}}{dt} + \frac{dV_{i,\partial}}{dt}$. The so-called Interface Velocities Consistency (IVC) conditions [66] are so obtained

$$\frac{dV_{ik}(t)}{dt} = \nu_{ik}(t) \qquad \forall k \in \mathcal{K}_{i,\neq} \qquad \text{and} \qquad \frac{dV_{i,\partial}(t)}{dt} = \nu_i(t) \qquad \text{if } i \in \mathcal{K}_\partial, \tag{7}$$

where $\mathcal{K}_\partial$ is the set of boundary nodes. Relations (7), given a known grid motion, allow to compute a value of the interface velocities that automatically satisfies the GCL constraint and they complement the System (4) made of $5 \times N_V$ Ordinary Differential Equations.

2.2. Metric computation on tetrahedral grids

The expression for the metric quantities and the cell volume in Equation (4) are now obtained in terms of the nodes positions for tetrahedral grids, on which the node-pairs correspond to the edges. From Fig. 1, it could be noticed that the generic cell interface $\partial\mathcal{C}_{ik}$ is composed by different contributions pertaining to all grid elements that share the edge ik. Considering only the elemental interface contribution $\partial\mathcal{C}^m_{ik}$ from the element m, it is composed by two triangular facets $\partial\mathcal{C}^m_{ik,f}$ stemming from the different faces f of the tetrahedron m to which the edge ik belongs. The three nodes of these facets are the barycenters of the element $\boldsymbol{x}_m$, of the edge $\boldsymbol{x}_{ik}$ and of the face $\boldsymbol{x}_f$ (Fig. 2). According to this partition of the interface, the integrated normals are computed on each triangular facets and then summed together. The elemental contribution to the integrated normal $\boldsymbol{\eta}_{ik}$ of the element m for the face f can be therefore computed as

$$\boldsymbol{\eta}^m_{ik,f} = -1/2\left(\boldsymbol{x}_f - \boldsymbol{x}_m\right) \times \left(\boldsymbol{x}_{ik} - \boldsymbol{x}_m\right) . \tag{8}$$

Similarly, the boundary elemental contribution is computed over the triangle formed by the boundary node i, the barycenters of the boundary element m^∂ and of its face f^∂, namely

$$\boldsymbol{\xi}^{m^\partial}_{i,f^\partial} = -1/2\left(\boldsymbol{x}_{f^\partial} - \boldsymbol{x}_{m^\partial}\right) \times \left(\boldsymbol{x}_i - \boldsymbol{x}_{f^\partial}\right).$$

The metric vectors $\boldsymbol{\eta}_{ik}$ and $\boldsymbol{\xi}_i$ can be then computed by summing all the elemental contributions

$$\boldsymbol{\eta}_{ik} = \sum_{m \in \varepsilon_i \cap \varepsilon_k} \sum_{f \in \mathcal{F}_{ik,m}} \boldsymbol{\eta}^m_{ik,f} \qquad \text{and} \qquad \boldsymbol{\xi}_i = \sum_{m^\partial \in \varepsilon^\partial_i} \sum_{f \in \mathcal{F}_{i,m^\partial}} \boldsymbol{\xi}^{m^\partial}_{i,f^\partial}, \tag{9}$$

where ε_i and ε_k are the set of the elements sharing the node i and k, respectively, ε^∂_i is the set of boundary elements sharing the nodes i, $\mathcal{F}_{ik,m}$ (and $\mathcal{F}_{i,m^\partial}$) is the set of faces of m (m^∂) sharing the edge ik (node i). The intersection $\varepsilon_i \cap \varepsilon_k$ denotes the set of all elements sharing the edge ik. When summing all contributions, particular care must be taken to preserve the correct orientation of the metric vectors, because Equation (8) assumes that the contribution $\boldsymbol{\eta}^m_{ik,f}$ is oriented as the local edge ik of the face f, which may be different to the global orientation of the edge. Conversely, the boundary elemental contributions are computed always as oriented outwards the domain.

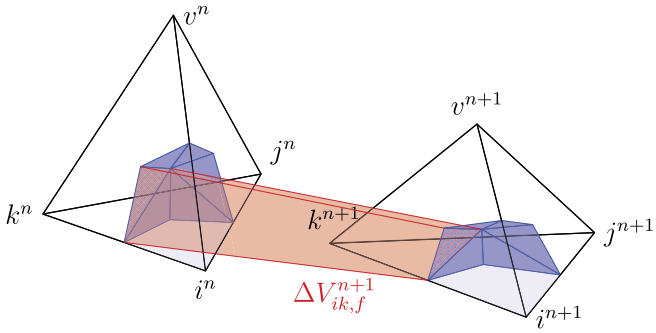

Fig. 3. Volume swept by the elemental interface $\partial\mathcal{C}^m_{ik,f}$, during the time interval $t^n \le t \le t^{n+1}$, as the result of the displacement of the nodes from their positions $\boldsymbol{x}^n$ to $\boldsymbol{x}^{n+1}$. The red pattern ※ highlights the elemental interface $\partial\mathcal{C}^m_{ik,f}$, while the red area ■ depicts the swept volume $\Delta V^{m,n}_{ik,f}$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Also the volume V_i of the finite volume $\mathcal{C}_i$ is computed by summing elemental contributions. According to the subdivision described above, each tetrahedron is split into 24 sub-tetrahedra, each having the triangular facet $\partial\mathcal{C}^m_{ik,f}$ as base and the node i or k as the opposite vertex. Thus, the volume of the finite volume $\mathcal{C}_i$ is computed as

$$V_i = \sum_{m\in\varepsilon_i}\sum_{k\in\mathcal{K}^m_{i,\neq}}\sum_{f\in\mathcal{F}_{ik,m}} V^m_{ik,f}\,, \qquad \text{with} \qquad V^m_{ik,f} = \frac{1}{3}\,(\boldsymbol{x}_m \times \boldsymbol{x}_i)\cdot\boldsymbol{\eta}_{ik,f}\,, \tag{10}$$

where $\mathcal{K}^m_{i,\neq}$ are the three vertices of the tetrahedron m different from i.

The proposed node-pair finite volumes discretization is very standard and straightforward to be implemented. After the median-dual mesh is generated and the edge structure computed, no information about elements is required by the flow solver [67,68]. Note that the edge-based representation here described is independent from the type of element of the initial grid and it is therefore very suitable for hybrid grids made of elements of different types [69]. Moreover, Selmin [61] and Selmin and Formaggia [62] proved that for fixed simplex grids (triangular in 2D and tetrahedral in 3D) the described finite volume discretization is equivalent to a finite element one with linear elements, except a different treatment of boundary terms.

2.3. Time integration of the ALE equations

Equations (4) and (6) are integrated in time with the Backward Euler scheme. The fully-discrete system of the governing ALE equations therefore, reads

$$\begin{cases} V_i^{n+1}\mathsf{u}_i^{n+1} - V_i^n\mathsf{u}_i^n = \Delta t\Big[\sum\limits_{k\in\mathcal{K}_{i,\neq}} \Phi(\mathsf{u}_i,\mathsf{u}_k,\nu_{ik},\hat{\boldsymbol{\eta}}_{ik},\eta_{ik})^{n+1} + \Phi^{\partial}(\mathsf{u}_i,\nu_i,\hat{\boldsymbol{\xi}}_i,\xi_i)^{n+1}\Big], & i\in\mathcal{K} \\ \Delta V_{ik}^{n+1} = \Delta t\ \nu_{ik}^{n+1}\,, & k\in\mathcal{K}_{i,\neq} \\ \Delta V_{i,\partial}^{n+1} = \Delta t\ \nu_i^{n+1}\,, & i\in\mathcal{K}_{\partial} \end{cases} \tag{11}$$

where ΔV_{ik}^{n+1} and $\Delta V_{i,\partial}^{n+1}$ are the volume swept by the interfaces $\partial\mathcal{C}_{ik}$ and $\partial\mathcal{C}_i\cap\partial\Omega$ during the time step $\Delta t^n = t^{n+1} - t^n$, see Fig. 3. Explicit expressions for the swept volumes are given in the following Subsection 2.4.

In the previous system, all quantities at time t^n are known, as well as all the grid-dependent quantities since grid motion is known. Thus, the two last equations are not coupled to the Euler equations. However, the coupled form of this system is preferred here to highlight the existence of the consistency constraints, see Equation (7), on the interface velocities.

A pseudo-time step method is used to solve the non-linear System (11) [70]. At each pseudo step, a modified Newton method is used to solve the linear system, in which, according to the defect-correction approach, the exact Jacobian of the integrated flux function is replaced by an approximated one [71].

2.4. Interface velocity computation on tetrahedral grids

The last two equations in System (11) are the so-called Discrete Geometric Conservation Laws (DGCL) [5], which are obtained by integrating in time the GCL condition (3). The DGCL states that the variation in volume in a certain time interval should balance the volume swept by its boundary during the same interval. In System (11), this constraint is matched by computing each interface velocity, ν_{ik} and ν_i, in terms of the volume swept by the corresponding portion of the cell interface, i.e. ΔV_{ik} and $\Delta V_{i,\partial}$ respectively.

The swept volumes are computed by recasting to the same interface partition introduced in Subsection 2.2. Thus, the volume swept by a domain interface is

$$\Delta V_{ik} = \int_{t^n}^{t^{n+1}} \int_{\partial \mathcal{C}_{ik}} \boldsymbol{v}(t) \cdot \boldsymbol{n}_i(t)\, d\boldsymbol{x}\ dt = \sum_{m \in \varepsilon_i \cap \varepsilon_k} \sum_{f \in \mathcal{F}_{ik,m}} \Delta V^m_{ik,f}\,,$$

where $\Delta V^m_{ik,f}$ is the volume swept by $\partial \mathcal{C}^m_{ik,f}$, as shown in Fig. 3. Assuming a constant velocity of the grid points during the time step, it can be evaluated as

$$\Delta V^{m,n}_{ik,f} = \frac{\boldsymbol{v}^n_m + \boldsymbol{v}^n_f + \boldsymbol{v}^n_{ik}}{3} \cdot \int_{t^n}^{t^{n+1}} \boldsymbol{\eta}^m_{ik,f}(t)\, dt \tag{12}$$

where $\boldsymbol{v}^n_j = (\boldsymbol{x}^{n+1}_j - \boldsymbol{x}^n_j)/\Delta t^n$ are the velocities of the barycenters of the element $\boldsymbol{x}_m$, of the face $\boldsymbol{x}_f$ and of the edge $\boldsymbol{x}_{ik}$, for $j = m, f, ik$, respectively. Considering a linear function for the point position, i.e. $\boldsymbol{x}_j(t) = \boldsymbol{x}^n_j + \boldsymbol{v}^n_j(t - t^n)$, the integrated normal portion $\boldsymbol{\eta}^m_{ik,f}(t)$ results

$$\boldsymbol{\eta}^m_{ik,f}(t) = \boldsymbol{\eta}^n_{ik} + \frac{t - t^n}{\Delta t^n}\left[-3\boldsymbol{\eta}^n_{ik} + 4\boldsymbol{\eta}^{n+\frac{1}{2}}_{ik} - \boldsymbol{\eta}^{n+1}_{ik}\right] + 2\left(\frac{t - t^n}{\Delta t^n}\right)^2 \left[\boldsymbol{\eta}^n_{ik} - 2\boldsymbol{\eta}^{n+\frac{1}{2}}_{ik} + \boldsymbol{\eta}^{n+1}_{ik}\right] \tag{13}$$

where the indices f and m are omitted in the right side to simplify the notation and $\boldsymbol{\eta}^{n+\frac{1}{2}}_{ik}$ is defined as

$$\boldsymbol{\eta}^{n+\frac{1}{2}}_{ik} = 1/4(\boldsymbol{\eta}^n_{ik} + \boldsymbol{\eta}^{n+1}_{ik}) - 1/8\left[(\boldsymbol{x}^n_f - \boldsymbol{x}^n_m) \times (\boldsymbol{x}^{n+1}_{ik} - \boldsymbol{x}^{n+1}_m) + (\boldsymbol{x}^{n+1}_f - \boldsymbol{x}^{n+1}_m) \times (\boldsymbol{x}^n_{ik} - \boldsymbol{x}^n_m)\right]. \tag{14}$$

Finally, integrating in time and substituting the expressions for the velocities of the points, the equation for the volume swept by the interface $\partial \mathcal{C}^m_{ik,f}$ during the time step Δt^n is obtained:

$$\Delta V^{m,n}_{ik,f} = \frac{1}{18}(\boldsymbol{x}^{n+1}_m - \boldsymbol{x}^n_m + \boldsymbol{x}^{n+1}_f - \boldsymbol{x}^n_f + \boldsymbol{x}^{n+1}_{ik} - \boldsymbol{x}^n_{ik}) \cdot (\boldsymbol{\eta}^{m,n}_{ik,f} + \boldsymbol{\eta}^{m,n+1}_{ik,f} + 4\boldsymbol{\eta}^{m,n+\frac{1}{2}}_{ik,f})\,. \tag{15}$$

With a similar procedure, the following expression for the volume swept by a boundary interface is obtained

$$\Delta V^{m^\partial,n}_{i,f^\partial} = \frac{1}{18}\left(\boldsymbol{x}^{n+1}_{m^\partial} - \boldsymbol{x}^n_{m^\partial} + \boldsymbol{x}^{n+1}_{f^\partial} - \boldsymbol{x}^n_{f^\partial} + \boldsymbol{x}^{n+1}_i - \boldsymbol{x}^n_i\right) \cdot \left(\boldsymbol{\xi}^{m^\partial,n}_{i,f^\partial} + \boldsymbol{\xi}^{m^\partial,n+1}_{i,f^\partial} + 4\boldsymbol{\xi}^{m^\partial,n+\frac{1}{2}}_{i,f^\partial}\right), \tag{16}$$

with $\boldsymbol{\xi}^{n+\frac{1}{2}}_{i,f^\partial} = 1/4(\boldsymbol{\xi}^n_i + \boldsymbol{\xi}^{n+1}_i) \pm 1/8\left[(\boldsymbol{x}^n_m - \boldsymbol{x}^n_f) \times (\boldsymbol{x}^{n+1}_i - \boldsymbol{x}^{n+1}_f) + (\boldsymbol{x}^{n+1}_m - \boldsymbol{x}^{n+1}_f) \times (\boldsymbol{x}^n_i - \boldsymbol{x}^n_f)\right]$ where the sign $\pm$ depends on whether i is the first $(-)$ or second $(+)$ node of the node-pair corresponding to the face f^∂.

3. Conservative ALE scheme for adaptive meshes

In the previous section the IVC conditions, solved along with the Euler equations in System (11), were shown to guarantee the fulfillment of the DGCL on dynamic grids with fixed connectivity. More precisely, the change in position and shape of the finite volumes can be easily taken into account by computing the interface velocities as the sum of the volume swept by the interfaces during the grid deformation. This technique is now extended to adaptive grids with variable connectivity by means of the three-steps procedure proposed by Guardone and co-workers for bi-dimensional problems [57,56], that allows to compute the swept volume also in presence of topology changes. This contribution, labeled $\Delta V^{\mathsf{A}}_{ik}$ to highlight that it originates from a connectivity modification, is then summed to the one due to simple deformation, labeled $\Delta V^{\mathsf{D}}_{ik}$. Then the IVC-compliant interface velocities are computed from the last two relations of System (11) considering $\Delta V_{ik} = \Delta V^{\mathsf{D}}_{ik} + \Delta V^{\mathsf{A}}_{ik}$. A thorough description of this procedure for triangular grids in two dimensions can be found in [57,56], while here its application to tetrahedral grids is detailed, followed by the final systems of ALE governing equations for adaptive grids.

3.1. Continuous interpretation of topology modifications

The fundamental idea of the three-steps procedure is to describe the grid modification as a sequence of fictitious continuous deformations for which the volumes swept by cell interfaces can be computed from Equations (15) and (16). This can be accomplished by the following three-steps procedure, in which a fictitious time ζ is used to better distinguish the different phases, with the initial configuration at $\zeta = 0$ and the final one at $\zeta = 1$.

1. *Collapse*, $0 < \zeta < 0.5$: all elements involved in the local modification collapse into an arbitrary point.

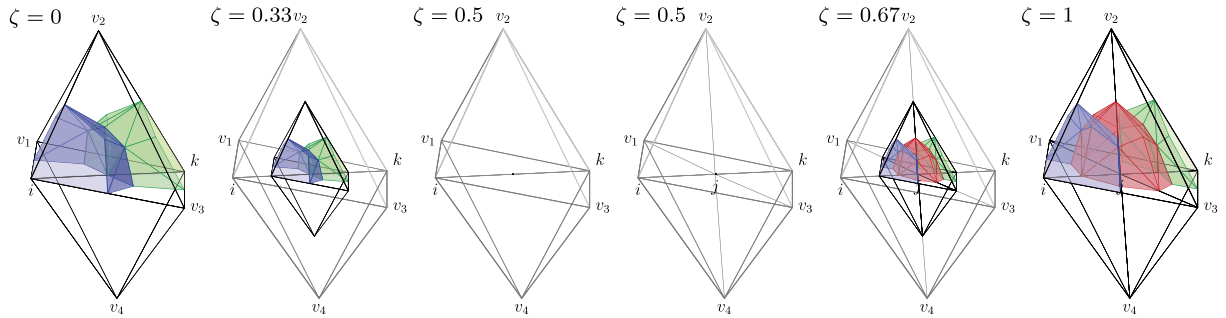

Fig. 4. Exemplary three-steps procedure applied to the splitting of the edge ik. The portions of the finite volumes pertaining to the polyhedron i–k–v_1–v_2–v_3–v_4 associated to the nodes i, k and j are shown, respectively, with blue /, green /, and red / (the darker colors refer to the boundaries of the finite volumes). The procedure begins with the collapse phase. The connectivity change, shown in the third and fourth pictures, occurs at $\zeta = 0.5$ when the surface of the involved interfaces has null area. Then, the expansion takes place: the nodes i, k, v_1, v_2, v_3, v_4 return to their original positions and the new finite volume $\mathcal{C}_j$ expands to reach the final configuration (at $\zeta = 1$). For $0 < \zeta < 1$, the grey lines show the grid connectivity in the original/final configuration as a reference. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

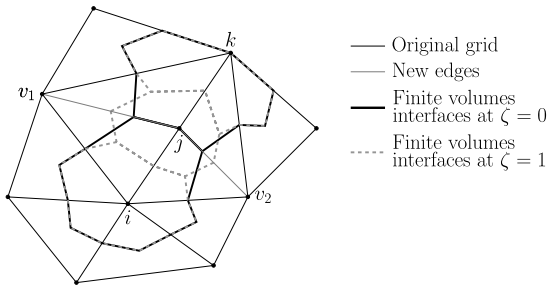

Fig. 5. Local character of connectivity modifications, example for an edge split in 2D. The initial and the final configuration for the split of the edge ik are shown overlapped. The boundaries of the finite volumes associated to i and k are drawn with different lines to display the modifications due to the edge split. The polyhedron formed by the union of the involved elements consists in the quadrilateral i–k–v_1–v_2. Outside this region, there are no differences between the initial and the final configuration.

2. *Connectivity change*, $\zeta = 0.5$: when all involved elements reach null volumes, nodes may be inserted or deleted.
3. *Expansion*, $0.5 < \zeta < 1$: all elements still active (i.e. not deleted at $\zeta = 0.5$) expand to the final configuration.

Fig. 4 shows the application of the three-steps procedure to an edge split, which consists in the insertion of a new node along a grid edge. With reference to Fig. 4, consider that the edge between node i and k has to be split to insert the new node j. This modification involves all the elements that share initially the edge ik, that are the four elements shown in the picture, because the finite volumes associate to nodes i, k, v_1, v_2, v_3, v_4 have to be modified to generate the new volume associated with the new node. Once the involved elements have been identified, the collapse phase can be carried out. At this point, the new node is inserted and connected with the existing ones to generate new tetrahedra. Finally, the elements expand towards the final configuration, which comprehends the new finite volume associated to the node j.

The collapse and the expansion phase are simply continuous deformations, therefore the swept volumes can be computed from Equations (15) and (16). Conversely, since the connectivity change occurs while the elements have null volume—and the interfaces have null area—no volume is swept by any interface in this phase. Thus, the connectivity change has no effects in terms of interface velocities or DGCL satisfaction and it does not generate any numerical fluxes. The complete contribution to the swept volume due to adaptation $\Delta V_{ik}^{\mathsf{A}}$ is given by the sum of the volume swept during the collapse and the expansion phases. It is important to notice the local character of the modification: the external faces of the polyhedron formed by the union of the involved elements remain unchanged, see Fig. 5 for a 2D example. Indeed, since no variation occurs in the grid elements outside the polyhedron, no variation occurs in the finite volumes either. Then, the volume swept by the interfaces located outside the polyhedron during the whole procedure is null, because the interfaces sweep the same volume but with opposite signs during the collapse and the expansion steps.

The three-steps procedure is general and it can be exploited to compute the IVC-compliant interface velocities in case of different local mesh adaptation techniques, like node insertion, node deletion and edge swap. Clearly, the elements involved are different depending on the grid modification. Table 1 describes the elements involved and the collapse point for all the local adaptation techniques used in the present work.

For instance, Fig. 6 shows how this procedure is applied to the edge collapse, during which the node j is collapsed over the node i. In this case the initial configuration contains elements that have to be removed to comply with the new

Table 1
Three-steps procedure details for the considered local mesh adaptation techniques. For each technique adopted in the present work, the elements composing the polyhedron that delimits the finite volumes influenced by the grid modification are indicated, along with the chosen collapse points.

Adaptation	Polyhedron made by	Collapse point
Edge split	Elements sharing the edge	New node
Element split	Element to split	New node
Delaunay insertion	Elements in the cavity	New node
Edge collapse	Elements sharing the node to be deleted	Node to be deleted

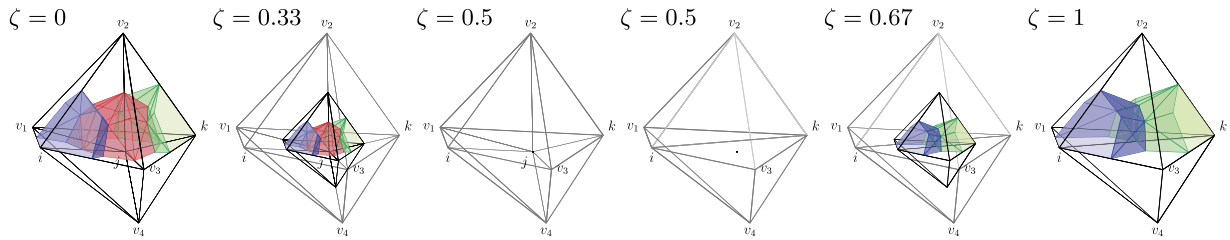

Fig. 6. Three-steps procedure applied to the collapse of the edge ij. The portions of the finite volumes pertaining to the polyhedron i–k–v_1–v_2–v_3–v_4 and associated to the nodes i, k and j are shown, respectively, with blue ■/■, green ■/■, and red ■/■ (the darker colors refer to the boundaries of the finite volumes). At $\zeta = 0.5$ (third and fourth pictures), the node j is collapsed over the node i. The elements that share the edge ij are deleted, while for the other elements the node j is substituted by the node i. Finally, only the elements that are not deleted are expanded to reach the final configuration (at $\zeta = 1$). The grey lines show the grid connectivity in the original/final configuration, for $0 < \zeta < 1$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

topology. Indeed, these elements are deleted at $\zeta = 0.5$ and they do not take part to the expansion phase. Also the finite volume connected to the node j is deleted at $\zeta = 0.5$. Nevertheless, it should be noticed that during the collapse phase the interfaces of $\mathcal{C}_j$ sweep a non-null volume, which generates non-null interface velocities. This fact has to be carefully taken into account in the flow solver, as described in next subsection.

3.2. ALE scheme with variable topologies

In this subsection, System (11) is extended to adaptive grids with variable connectivity. Before presenting the final system of equations, the reader's attention is drawn to some peculiarities of the present approach. A detailed description of the whole procedure can be found in [57,56,72].

A first remark concerns the time dependence of the grid connectivity. During each time step, a variable number of grid nodes can be added or deleted. Consequently, both the set of grid nodes and the set of the finite volumes change in time. In the following, a superscript is added to their symbols to indicate the time step or time interval at which they should be considered. For instance, $\mathcal{K}^{n+1}$ denotes the set of nodes that compose the grid at time level t^{n+1}.

Second, when a new node is added to the triangulation, an equation for the balance of the conservative variables over the new finite volume has to be added to the governing equations system and additional IVC conditions are required to compute the new interface velocities. These equations do not differ from the ones in System (11), however they are simpler since if the node i is inserted between the time step t^n and t^{n+1}, the volume V_i and the volume swept by its interface are null for all time steps previous than t^{n+1}.

Third, special care must be taken in dealing with the treatment of the deleted interfaces. When the grid edge ik is deleted, during the three-steps procedure the associated interface $\partial\mathcal{C}_{ik}$ is collapsed to reach a null area at $\zeta = 0.5$; from then on, it is considered as deleted. However, during the collapse phase it sweeps a non-null volume, which has to be included into the computation of the associated interface velocity ν_{ik} even if the edge ik is no longer present in the mesh. Thus, supposing that this modification occurs during the adaptation phase performed between t^n and t^{n+1}, the interface velocity is not null also for the next time step, i.e. $\nu_{ik}^{n+1} \neq 0$. As a consequence, the associated numerical flux is not-null and it is included in the system of governing equations by adding the term $\Phi_{ik}^{n+1} = \Phi(\mathrm{u}_i^{n+1}, \mathrm{u}_k^{n+1}, \nu_{ik}^{n+1}, \hat{\boldsymbol{\eta}}_{ik}^{n+1}, 0)$ where $\eta_{ik}^{n+1} = 0$ because of the null area of integration while the normal unit vector is defined as $\hat{\boldsymbol{\eta}}_{ik}^{n+1} = \lim_{t \to t^{n+1}} \hat{\boldsymbol{\eta}}_{ik}(t)$ [57]. Furthermore, the associated IVC conditions have to be enforced until the interface velocity is identically null. As explained by Isola et al. [56], the fact that the interface velocity does not necessarily become null while the interface collapses and is removed from the mesh is a consequence of the differential nature of the GCL constraint and it does not depend on the time integration scheme.

According to the previous remarks, the system of conservative governing equations for inviscid flows within the ALE framework over adaptive grids can be written as

$$
\begin{cases}
\dfrac{V_i^{n+1}\mathsf{u}_i^{n+1} - V_i^{n}\mathsf{u}_i^{n}}{\Delta t} = \displaystyle\sum_{k\in\mathcal{K}_{i,\neq}^{n+1}} \Phi(\mathsf{u}_i, \mathsf{u}_k, \nu_{ik}, \hat{\boldsymbol{\eta}}_{ik}, \eta_{ik})^{n+1} + \Phi^{\partial}(\mathsf{u}_i, \nu_i, \hat{\boldsymbol{\xi}}_i, \xi_i)^{n+1} & \\
\qquad\qquad \displaystyle\sum_{k\in\mathcal{K}_{i,\neq}^{[n-p,n+1)}} \Phi(\mathsf{u}_i, \mathsf{u}_k, \nu_{ik}, \hat{\boldsymbol{\eta}}_{ik}, 0)^{n+1}, & i \in \mathcal{K}^{n+1} \\
\dfrac{V_i^{n}\mathsf{u}_i^{n}}{\Delta t} = \displaystyle\sum_{k\in\mathcal{K}_{i,\neq}^{[n-p,n+1)}} \Phi(\mathsf{u}_i, \mathsf{u}_k, \nu_{ik}, \hat{\boldsymbol{\eta}}_{ik}, 0)^{n+1} + \Phi^{\partial}(\mathsf{u}_i, \nu_i, \hat{\boldsymbol{\xi}}_i, 0)^{n+1}, & i \in \mathcal{K}^{[n,n+1)} \\
\Delta V_{ik}^{n+1} = \Delta t\ \nu_{ik}^{n+1}, & k \in \mathcal{K}_{i,\neq}^{[n,n+1]} \\
\Delta V_{i,\partial}^{n+1} = \Delta t\ \nu_{i}^{n+1}, & i \in \mathcal{K}_{\partial}^{[n,n+1]}
\end{cases}
\tag{17}
$$

where $\mathcal{K}_{i,\neq}^{[n,n+1)} = \{k \in \mathcal{K},\ k \notin \mathcal{K}_{i,\neq}^{n+1} \mid \nu_{ik}^{n+1} \neq 0\}$ is the set of finite volumes that have shared a portion of their boundary with $\mathcal{C}_i$ in the interval from t^n to t^{n+1} but that do not share any portions at t^{n+1}, while $\mathcal{K}_{i,\neq}^{[n,n+1]} = \mathcal{K}_{i,\neq}^{[n,n+1)} \cup \mathcal{K}_{i,\neq}^{n+1}$ includes all finite volumes that share or have shared their boundary with $\mathcal{C}_i$.

The first and the second equations of System (17) express the balance of conservative variables for the finite volumes associated to all nodes of the triangulation at time t^{n+1} and to the nodes removed between t^n and t^{n+1}, respectively. The computation of the solution on nodes removed during the previous steps is essential to the correct evaluation of the solution on the existing nodes. Indeed, if the removed node j was connected to the existing node i by an edge that is no longer present in the grid, the flux Φ_{ij}^{n+1} is required to compute the actual solution at node i, but its evaluation requires the value of u_j^{n+1}, which is computed via the second equation of System (17). Then, since the interface velocity ν_{ij} remains different from zero only for the step next to its deletion, for $t > t^{n+1}$ the equation for the node j is discarded from the system and the solution u_j is not computed.

Finally, the last two rows of System (17) express the IVC conditions that allow to compute the cell volume changes due to mesh deformation and adaptation so that the DGCL is matched. Thanks to the three-steps procedure described in Subsection 3.1, the solution can be conservatively computed within the ALE framework even if topology changes occur due to grid adaptation.

3.3. Multiple modifications involving the same node-pair

During a single time step diverse topology modifications may affect the same interface. The proposed approach enable to cope with this situation by summing the contributions of all the collapse and expansion phases experienced by the interface. Although this choice is not mandatory, it can be implemented in an extremely straightforward way in the node-pair representation detailed in Section 2. Indeed, after each local modification, a loop on all involved interfaces is performed and the volume swept by each one is computed and added to the corresponding node-pair. At the end of the time level, the swept volume due to adaptation is summed to the one due to deformation and to obtain the total amount ΔV_{ik}^{n+1}.

Finally, particular care is devoted to the update of the node-pair spatial discretization after a mesh modification. Although in a tetrahedral finite-volume dual-mesh discretization node-pairs are naturally associated to grid edges, the adaptive scheme requires one to include also some node-pairs that are associated to deleted edges. In other words, two grid nodes may interact with each other even if they are not connected by an edge. To this end, when the topology re-construction requires the creation of a new node-pair, it should be checked if in the set of the "deleted-but-still-active" node-pairs already exists the one connecting these nodes. If this case, the node-pair is restored and the volume swept by the associated interface is taken into consideration.

4. Overview of the computational procedure for adaptive simulations

The proposed conservative ALE scheme for adaptive grids has been implemented in the FlowMesh solver that is currently under development at the Department of Aerospace Science and Technology of Politecnico di Milano. This numerical tool is able to perform unsteady three-dimensional simulations over moving-boundary problems, even when the domain boundaries experience large deformations [72]. This section gives an overview of the different techniques and of the complete procedure used to carried out such simulations. Additional details can be found in [72,73].

Moving-boundary problems require a mesh deformation strategy to update the computational domain to follow the body motion or the deformation of some of its surfaces, which in this work are assumed to be known. Therefore, at each time step, after the boundary nodes are moved on the new positions, the internal (or domain) nodes have to be displaced to obtain a new mesh that is conform with their movement. A mesh displacement strategy is developed on the basis of the elastic analogy proposed by Batina [74], which is easy to be implemented and requires a reasonable computational cost. The grid is treated as the union of solid elastic bodies which correspond to grid elements and have a stiffness inversely proportional to their volume. The displacement field is then obtained by solving the stress equilibrium imposing the prescribed boundary motion as Dirichlet boundary conditions [75]. Thanks to this approach, the deformation occurring at the

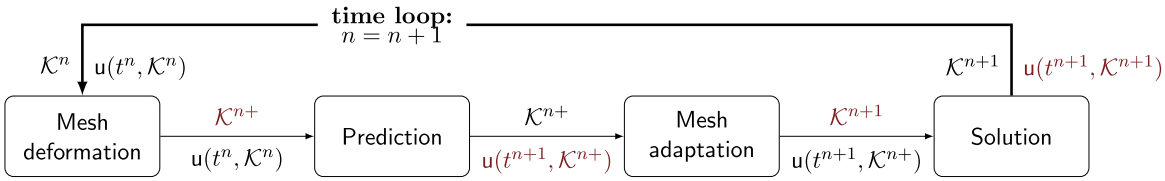

Fig. 7. Computational procedure for adaptive unsteady problems. At the beginning of the time step, the mesh deformation strategy is applied to update the grid $\mathcal{K}^n$ to the grid $\mathcal{K}^{n+}$ that complies with the boundary displacement. A prediction phase is then performed to compute the solution at time t^{n+1} over this new grid. Mesh adaption is then carried out and the target grid spacing is built according to the predicted solution $\mathsf{u}(t^{n+1}, \mathcal{K}^{n+})$. Finally, the solution at the time level t^{n+1} over the adapted grid $\mathcal{K}^{n+1}$ is computed. The red color highlights the new output of each computational step. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

boundary is redistributed among internal grid elements: the largest elements account for the major part while the smallest ones move almost rigidly, avoiding so possible entanglement.

If the above procedure does not succeed in producing a displacement field that results in a valid mesh, i.e. without poor-quality or negative-volume elements, the target displacement is subdivided in more intervals until a valid mesh is obtained through elastic analogy or a user-defined maximum number of subdivisions is reached. In the former case the remaining portion of the displacement is carried out, while in the latter case mesh optimization is performed to restore grid quality [73]. To efficiently modify three-dimensional grids, the capabilities offered by the automatic tetrahedral re-mesher Mmg3d are exploited [59,42]. This library is part of the wider open-source software Mmg which offers different tools for surface and volume mesh adaptation, optimization and generation. In addition to a series of local grid modifications which can be performed both on the interior and on the surface of the grid, Mmg3d implements also the gradation control technique [76] to limit the variation in size among adjacent grid element and the validity of the boundary representation is guaranteed by the Hausdorff criterion [77]. A detailed description of all the techniques exploited in Mmg3d can be found in [78,42]. The local operators exploited in this work are quite standard and include the node insertion through element split, edge split or Delaunay triangulation, the node deletion through edge collapse, the edge swapping and the barycentric regularization, see Table 1.

Remark: it is important to highlight that the application of the conservative ALE scheme here proposed is not limited to these operators, but it can be extended to all local modifications that can be described by the three-steps procedure.

During mesh optimization, each local modification aims at improving the quality of the involved elements which is defined as

$$Q_m = \alpha\, V_m / \left(\sum_{i=1}^{6} \ell_i^2 \right)^{3/2}, \tag{18}$$

where V_m is the volume of the element m, ℓ_i is the edge length and α is a constant parameter introduced to give $Q_m = 1$ for regular tetrahedra, to which corresponds the maximum quality.

In this work, the aim of remeshing is not limited to the enhancement of grid quality after mesh deformation, but it is exploited also to increase solution accuracy and to detect all relevant flow features whose exact location is not known a priori or change in time [37,36]. The key point in this process is the definition of a suitable criterion able to efficiently identify where grid modifications are required and to prescribe a proper size map. Since the main scope of the work concerns the assessment of the proposed conservative ALE scheme over adaptive grids, local criteria based on solution variations are adopted because of their simplicity and easy implementation. The underlying idea of this kind of adaptation criteria is that the largest errors occur where the solution exhibits the largest gradients, therefore the solution accuracy can be increased by gathering mesh points in these regions but it is not negatively affected by a coarsening of the grid regions where the solution exhibits a smooth behavior. Thus, a local error indicator can be obtained from the first- or second-order derivatives of a representative solution variable, as for instance density, pressure, Mach number, or of their combination and the grid spacing used as target during mesh adaptation can be built according to it. More specifically, the grid spacing is reduced where the error estimator exceeds a refinement threshold and it is enlarged where the estimator is below a coarsening threshold. These user-defined thresholds are prescribed in terms of the standard deviation and the mean of the error estimator over the domain. Only the edge size is prescribed and no information about the element shape or orientation is given since simple isotropic adaptation is performed. Mesh coarsening plays an important role especially in unsteady simulations where it avoids an excessively increase in the computational burden that may be caused by the insertion of a great number of new nodes [79].

During the mesh adaptation process, the re-mesher sends to the flow solver the information concerning each local adaptation immediately after it is performed, so that the swept volume ΔV^{A} of all involved interfaces can be update. Ad-hoc callback functions, similar to the ones proposed in [43], are used to this aim. Such as an implementation requires a strong link between the solver and the re-mesher, but avoids to store the history of the grid adaptation process.

Fig. 7 sketches the main steps of the computational procedure for unsteady simulations. The first block called *Mesh deformation* include the whole strategy described above, including the eventual mesh adaptation process carried out if

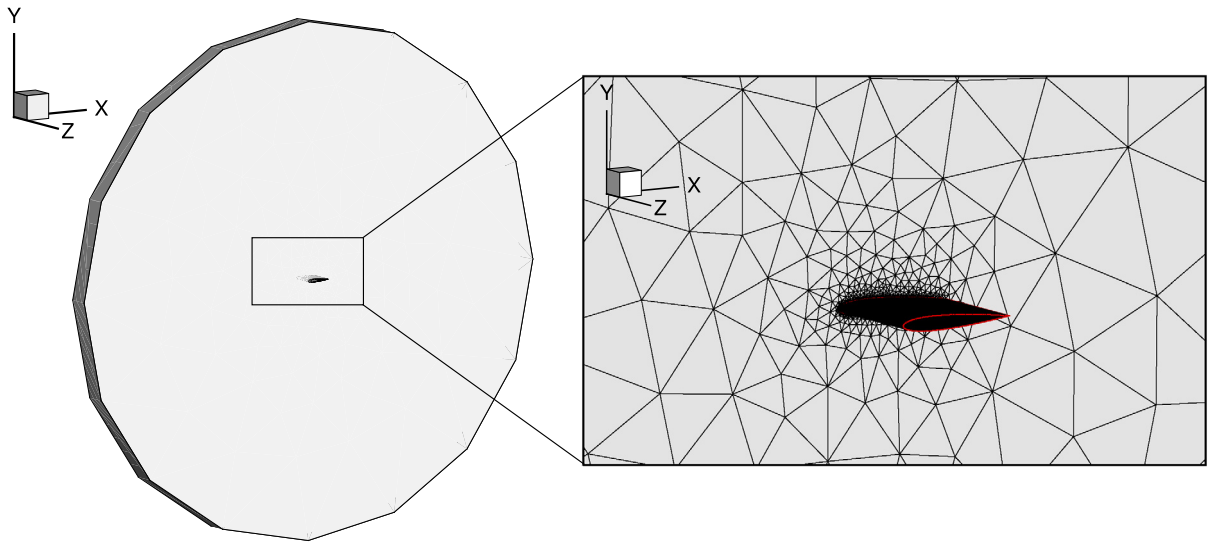

Fig. 8. At left: geometry of the whole computational domain for the infinite-span NACA 0012 wing tests. At right: enlargement of the rectangular portion of the domain highlighted at left, the surface meshes corresponding to the wall at $z = 0$ and to the wing are shown and the wing edges are drawn in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

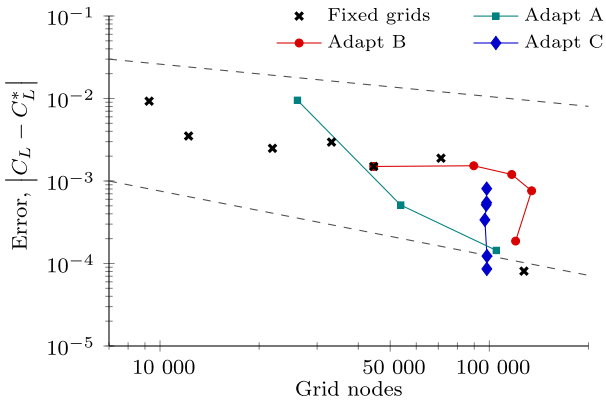

Fig. 9. Grid convergence studies for the transonic flow around the steady infinite-span wing test case. The absolute deviations of the lift coefficient C_L with respect to reference value C_L^* are shown versus the number of grid nodes. In addition to the results on the fixed grids described in Table 2, three series of adaptive steady computations are displayed. To draw the dashed lines corresponding to first- and second-order accuracy, the relation between the number of nodes and the grid spacing h are retrieved from grids with $h_w = 0.05$ and 0.025 in Table 2.

elastic analogy fails. The so-called *Prediction* step consists in the computation of the solution over the mesh that has been already deformed to comply with the boundary motion. Thanks to the prediction phase, mesh adaptation is driven by criteria computed over a solution updated on the new mesh configuration, rather than the one at the previous time step. The inclusion of the *Prediction* step is found to allow for larger time steps while preserving the same accuracy as it will be shown in the following Section 5. Thus, the complete procedure can be summarized as follows

I. Mesh deformation: the grid $\mathcal{K}^n$ is updated to $\mathcal{K}^{n+}$ to comply with the boundary position at t^{n+1}.
II. Prediction phase: the solution at the new time step, over the grid $\mathcal{K}^{n+}$, is computed.
III. Mesh adaptation of the grid $\mathcal{K}^{n+}$ on the basis of the new solution $\mathsf{u}(t^{n+1}, \mathcal{K}^{n+})$ and consistent update of the spatial discretization.
IV. Computation of the solution u^{n+1} over the grid $\mathcal{K}^{n+1}$, using as initial guess the solution $\mathsf{u}(t^{n+1}, \mathcal{K}^{n+})$.

It should be reminded that, thanks to the ALE interpretation of the local grid modifications as a series of continuous deformations, an explicit interpolation of the old solution over the new grid is not required.

Table 2
Fixed grids used in the convergence study for the transonic flow around the steady infinite-span wing test case. A linear variation in the grid spacing is obtained during the mesh generation by imposing the value h_w over the wing surface and the value h_∞ at far-field; to preserve the correct geometry of the wing surface, a smaller value h_{LE} is imposed near the trailing edge. The ratio between these three parameters is kept constant among all grids.

Wing surface h_w	Far-field h_∞	Leading edge h_{LE}	Grid nodes N_v
0.02	0.8	0.002	127 367
0.025	1.0	0.0025	71 315
0.03	1.2	0.003	44 548
0.035	1.4	0.0035	33 172
0.04	1.6	0.004	21 952
0.05	2.0	0.005	12 211
0.06	2.4	0.006	9 267

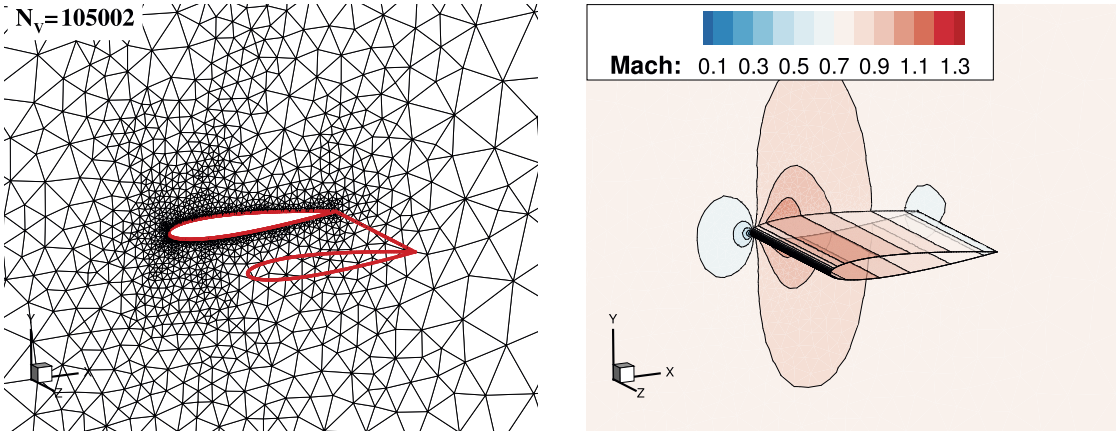

Fig. 10. Grid and Mach contour plots of the transonic flow around the steady infinite-span wing at $M_\infty = 0.755$ and $\alpha_\infty = 0.016°$ over the wall at $z = 0$ and over the wing surface. Since the grid elements over the wing surface are very small, to make the left picture clearer, only the wing edges are drawn, in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3
Quality improvement at different time steps of the test *Enhance* for the translating infinite-span wing with Δt_1, corresponding to a Courant number of 2. The average, the worst and the best value before and after mesh optimization are reported. Remind that the best quality is 1 and corresponds to regular tetrahedra.

Performed displacement	Before adaptation			After adaptation		
	Average	Worst	Best	Average	Worst	Best
4.8 %	0.695289	0.000083	0.999565	0.771129	0.041053	0.999369
10.4 %	0.644737	0.000024	0.998924	0.751961	0.104121	0.999074
16.0 %	0.629724	0.000013	0.997824	0.738972	0.108341	0.997708
25.6 %	0.580335	0.000018	0.998159	0.729795	0.083266	0.999420
33.6 %	0.607000	0.000005	0.998135	0.720783	0.078378	0.998894
41.6 %	0.619678	0.000023	0.998515	0.716963	0.079011	0.998515
44.8 %	0.657054	0.026241	0.998617	0.715037	0.083704	0.998617
56.8 %	0.563106	0.000005	0.998932	0.712779	0.081304	0.998932
64.8 %	0.614716	0.000012	0.998932	0.705929	0.057188	0.998932
76.8 %	0.580232	0.000001	0.998908	0.704456	0.078917	0.998606
86.4 %	0.595999	0.000029	0.998498	0.701020	0.074342	0.998498
94.4 %	0.608164	0.000010	0.998451	0.699856	0.084938	0.998451

5. Results

The three-dimensional conservative adaptive scheme is here assessed through simulations of reference test cases of aeronautical interest. First, Subsection 5.1 presents the steady results about the transonic flow around the infinite-span NACA 0012 wing, for which adaptive solutions are compared to the fixed-grid computations. The same problem is also tackled in the laboratory reference frame, i.e. by performing the unsteady simulation of the wing traveling through the still domain at

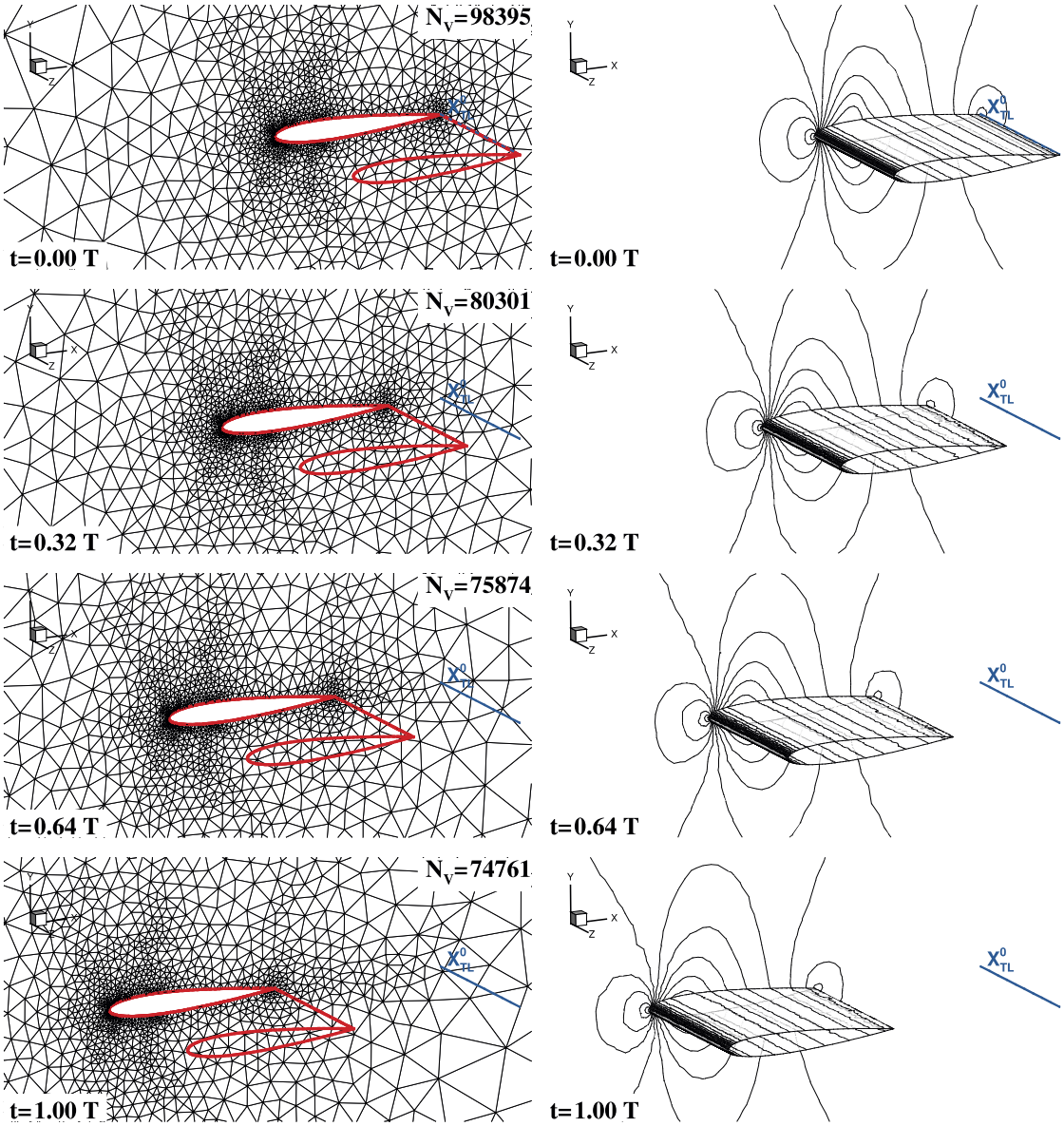

Fig. 11. Grids and pressure contour lines over the wall at $z = 0$ and over the wing surface at the beginning, at 33%, at 66% and at the end of one chord displacement for the infinite-span wing in the test *Enhance*, where mesh adaptation is exploited only when elastic analogy fails. The test is performed with the time step Δt_1 corresponding to a Courant number of 2. For grid plots, the wing is only sketched by drawing in red the edges because grid elements over the wing surface are too small to be clearly represented in the picture. The blue line labeled X^0_{TL} indicates the initial position of the trailing edge. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the flight velocity. The flow field around an oscillating infinite-span NACA 0012 wing is also investigated in Subsection 5.3 and the three-dimensional results are compared with reference ones. Then, steady and unsteady simulations are performed also for a finite-span NACA 0012 wing, a model which is of primary importance in rotor-crafts CFD since it can represent the tip of a rotor blade. Moreover, differently from the infinite-span wing case, this flow field is fully three-dimensional, so it is a valuable test to assess the capability of the proposed scheme.

Finally, in Subsections 5.5 and 5.6, two tests characterized by large boundary displacements are presented, i.e the infinite-span NACA 0012 wing traveling for 20 chords and a piston-induced shock-tube problem in which the domain experiences a reduction of more than the half of the initial size.

In all numerical experiments presented in this section, air is modeled as a constant-specific-heats ideal gas with specific heat ratio $\gamma = 1.4$. The flow equations have been made dimensionless by choosing the following reference values: $P_{\text{ref}} = 1$ atm, $\rho_{\text{ref}} = 1.225$ kg/m^3 and $L_{\text{ref}} = 1$ m.

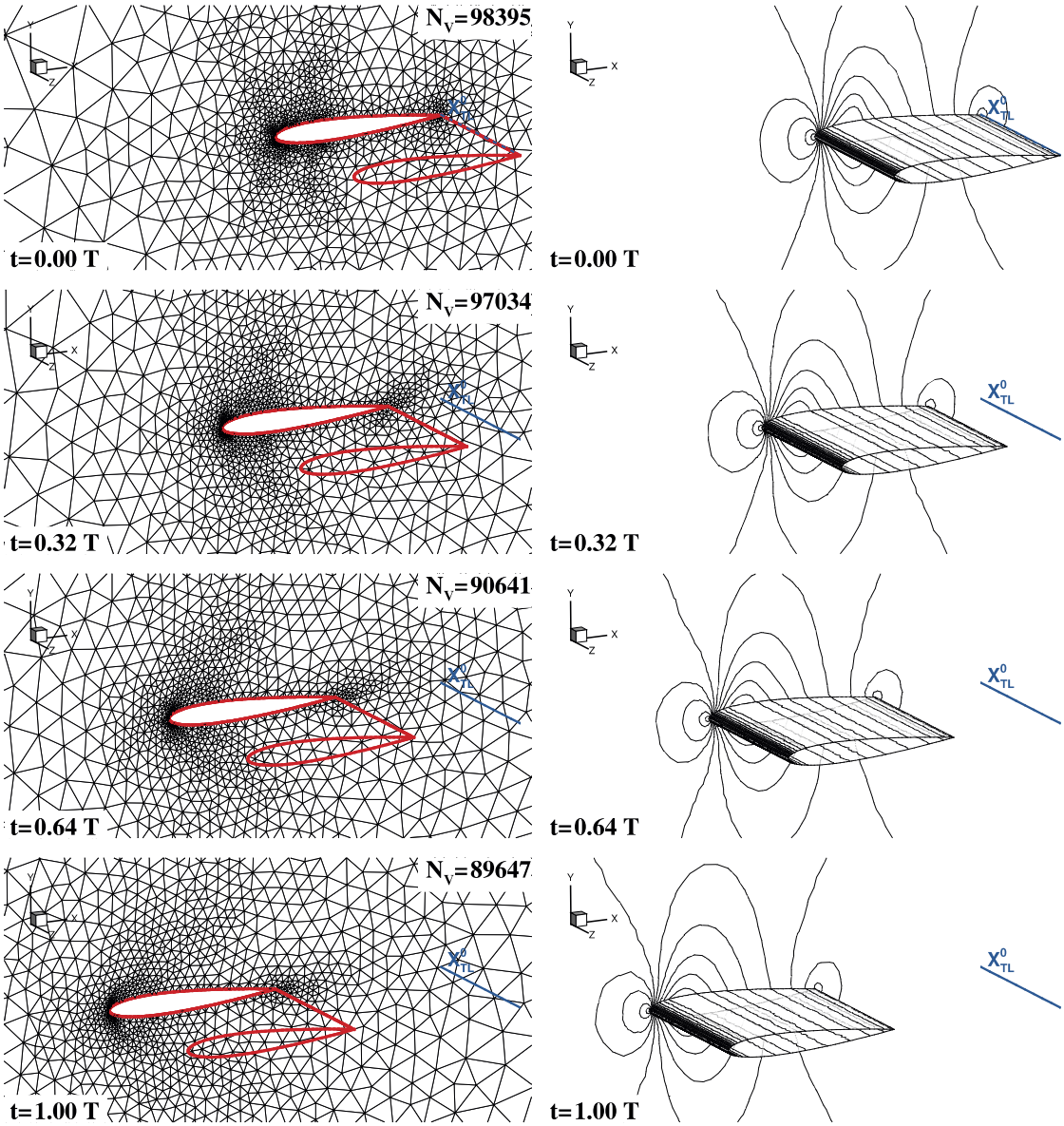

Fig. 12. Grids and pressure contour lines over the wall at $z = 0$ and over the wing surface at the beginning, at 33%, at 66% and at the end of one chord displacement for the infinite-span wing in the test *Adapt*, where mesh adaptation is performed every time step. The test is performed with the time step Δt_1 corresponding to a Courant number of 2. For grid plots, the wing is only sketched by drawing in red the edges because grid elements over the wing surface are too small to be clearly represented in the picture. The blue line labeled X^0_{TL} indicates the initial position of the trailing edge. In Supplementary Material, Movies S1 and S2 show the variation of the grid and of the pressure during this simulation, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.1. Steady simulation of infinite-span NACA 0012 wing

The computational domain built to model the flow field around a so-called infinite-span NACA 0012 wing is shown in Fig. 8. The wing span is equal to the chord c and the far-field consists in a cylindrical surface with a radius of $12c$, centered at the leading edge. For an infinite-span wing, the lateral planes at $z = 0$ and $z = c$ are treated as walls, which, under the considered inviscid approximation, correspond to symmetry boundary conditions. Therefore, the flow is intrinsically bi-dimensional. A free-stream flow at Mach number $M_\infty = 0.755$ and incidence $\alpha_\infty = 0.016°$ is imposed, along with unity dimensionless pressure and density.

A convergence study on fixed grids is performed by comparing the error on the lift coefficient obtained in different simulations. First, a reference value is computed on a fine grid composed by 622 693 nodes, gathered mainly near the wing surface where a grid spacing between 0.001 and 0.01 is enforced. It should be pointed out that some other fine grids have been tested, but no significant modifications in the reference lift coefficient were observed. Then, the fixed grids described

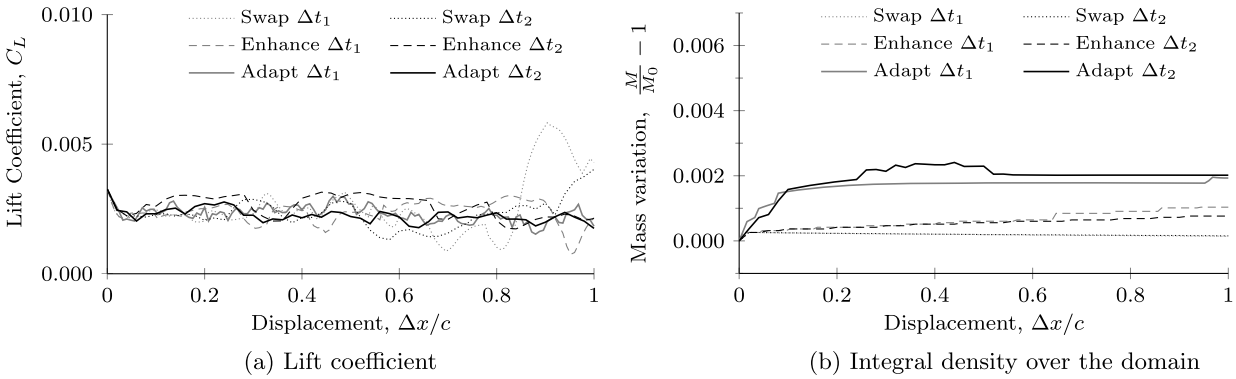

Fig. 13. Lift coefficient and integral value of the density over the domain for the infinite-span NACA 0012 wing in the laboratory frame. The results obtained by exploiting only edge-swapping (*Swap*), mesh adaptation only when elastic analogy fails (*Enhance*) and every time step (*Adapt*) are compared. Time steps Δt_1 and Δt_2 correspond to a Courant number of 2 and 5, respectively. In the left picture, the value of the integral density, labeled M, is scaled with respect to the steady value M_0.

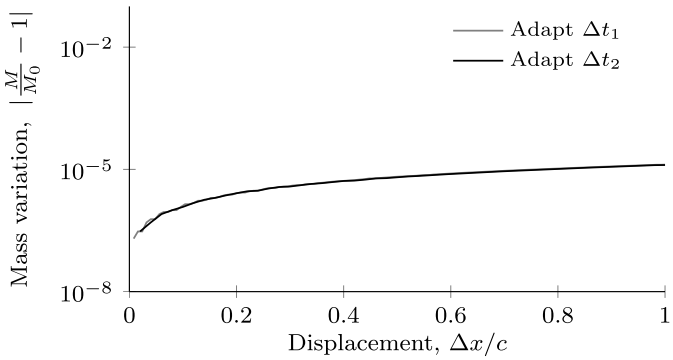

Fig. 14. Absolute variation of the integral value of the density over the domain for the infinite-span NACA 0012 wing in the laboratory frame, in the *Adapt* tests performed by enforcing no coarsening on the far-field boundary to avoid variations in the volume of the domain, which lead to variations in the integral density.

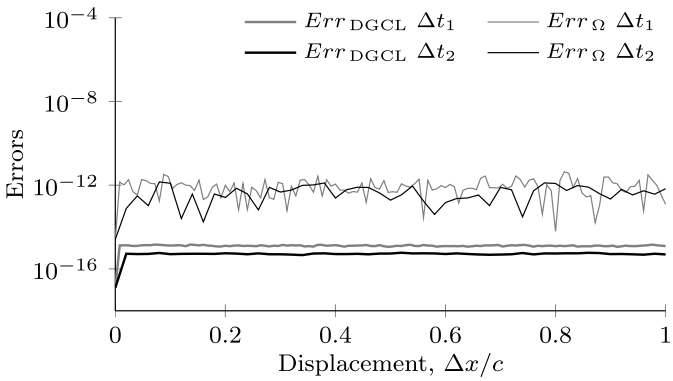

Fig. 15. Error on the fulfillment of the DGCL condition and on the volume closure in the test *Adapt* of the infinite-span NACA 0012 wing in the laboratory frame, with two different time steps. These two quantities are computed according to Definitions (19).

in Table 2 are used to perform steady computations. The errors with respect to the reference lift coefficient C_L^* are displayed in Fig. 9 and an order of convergence between one and two can be observed, as expected.

Fig. 9 reports also the results of three cycles, labeled *A*, *B* and *C*, of steady adaptive computations, consisting in only the last two steps of Fig. 7. For the cycle labeled *A*, starting from a quite coarse grid made of 26 142 nodes, two adaptation steps are sufficient to obtain a good convergence on the solution, i.e. the target grid spacing computed on the basis of the solution on the last grid does not deviate appreciably from the previous one. The final grid is composed by 105 002 nodes and 555 229 tetrahedra and a detail is displayed in Fig. 10, which shows the surface grid over the end-wall and the wing along with the solution. This result is used as initial solution in the following simulations of the pitching wing.

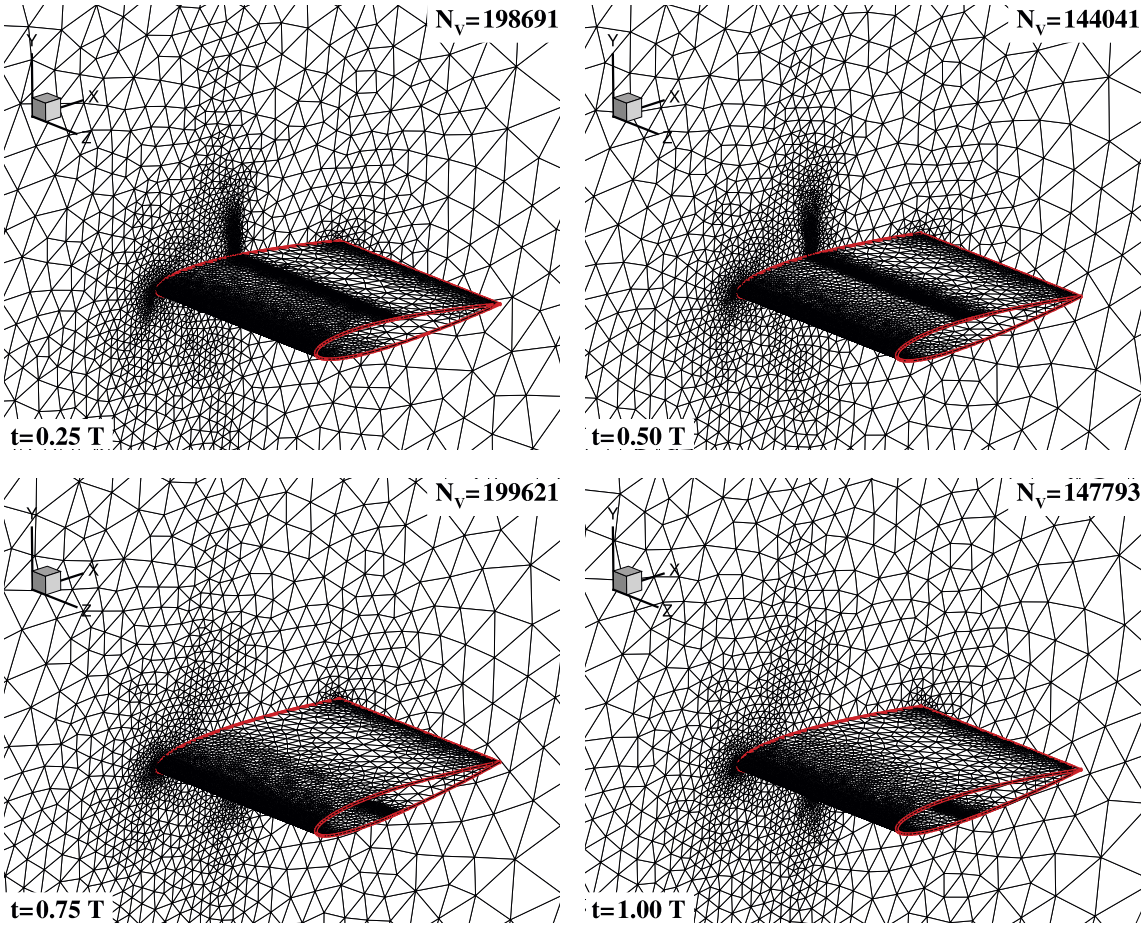

Fig. 16. Details of the computational grid on plane $z = 0$ and over the wing surface at different times of the pitching infinite-span wing. The last period of the simulation with Δt_1 is shown. In Supplementary Material, Movie S3 displays the grid during the last period of this simulation.

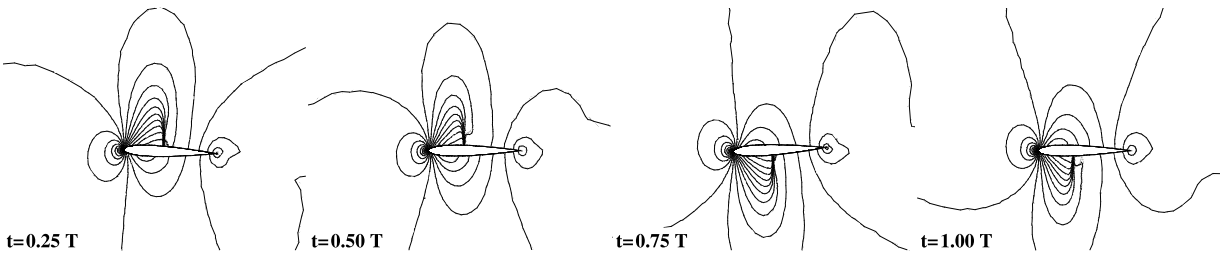

Fig. 17. Mach contour plot on plane $z = c$ at different times of the pitching infinite-span wing. The last period of the simulation with Δt_1 is shown.

In the cycle labeled B, four adaptation steps are performed starting from the grid characterized by $h_w = 0.003$ in Table 2. The error on lift coefficient follows approximately the trend shown by the fixed-grid results until the coarsening process becomes more relevant than the refinement, reducing the number of nodes without undermining the solution accuracy. In the last cycle, labeled C, a quite fine, uniform mesh is used as starting point, to assess the capability of the adaptation process to improve the solution accuracy without increasing excessively the grid size. Indeed, although the number of nodes remains approximately constant, the error with respect to the reference solution decreases during the cycle of steady adaptation. The final grid of this cycle, composed by of 98 113 nodes, is used in the following unsteady simulations of Subsection 5.2. Despite different initial grids, the three adaptation cycle leads approximately to the same number of nodes and lift coefficient values.

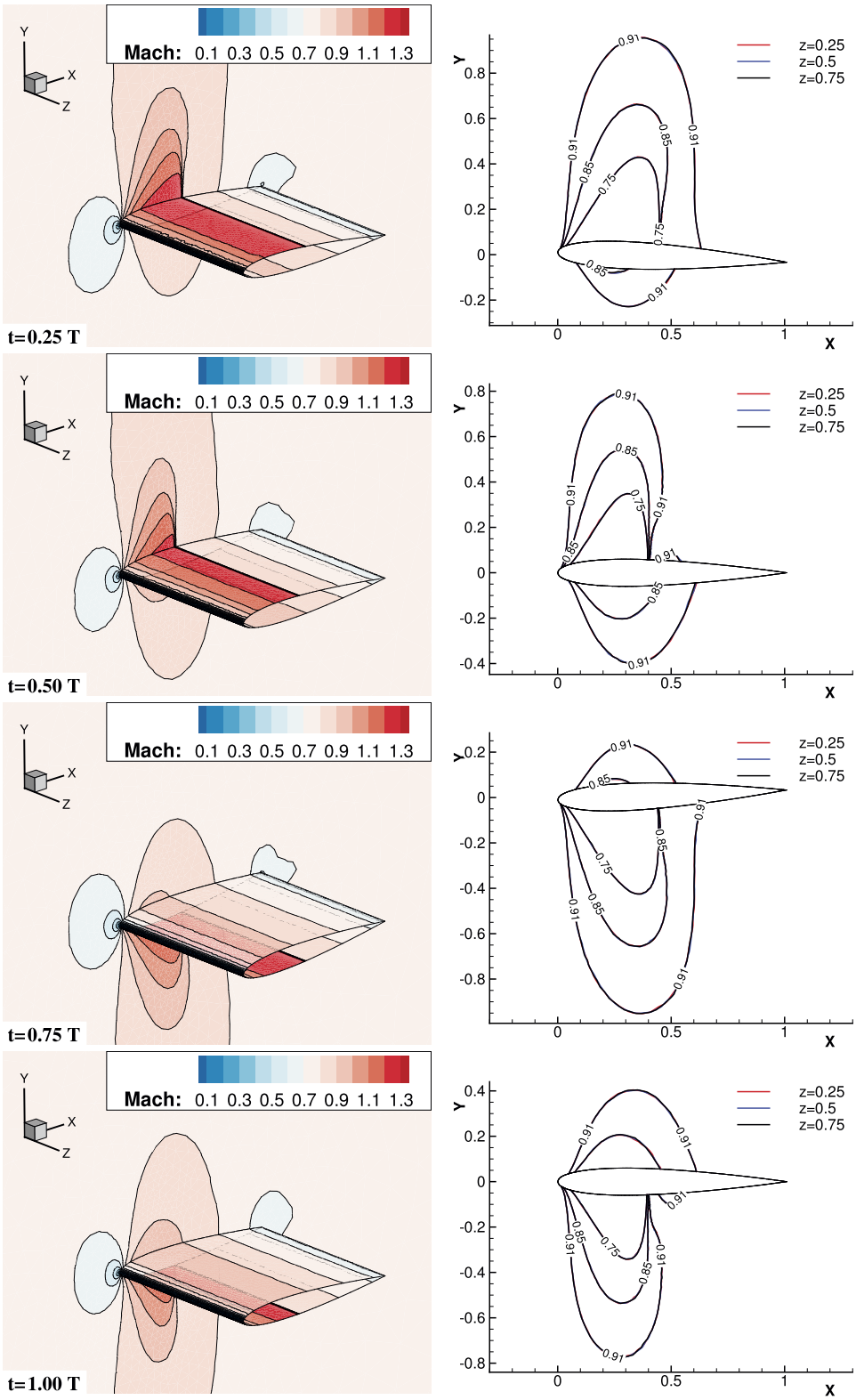

Fig. 18. Mach contour plot and pressure iso-lines every quarter of the last period of the simulation of the pitching infinite-span wing with Δt_1. The iso-lines corresponding to three values of the pressure at three different wing sections are compared. In Supplementary Material, Movie S4 shows the Mach number contour during the last period of this simulation.

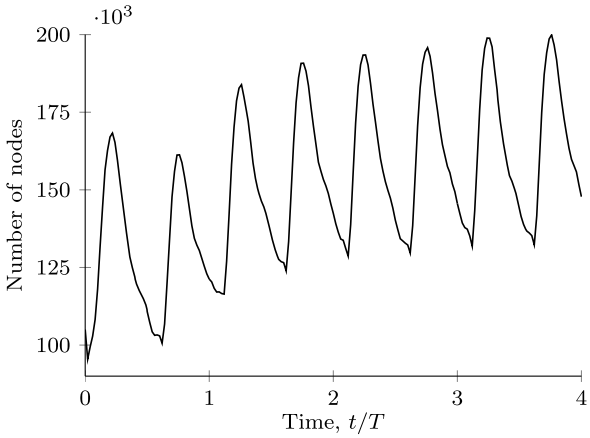

Fig. 19. Variation of the number of grid nodes during the whole simulation of the pitching infinite-span wing with Δt_1. Two peaks can be observed in each period, due to the two maximum intensities of the shock on the upper and the lower wing surface.

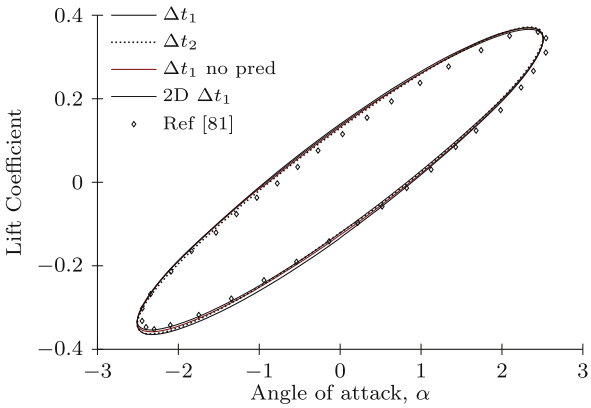

Fig. 20. Lift coefficient versus angle of attack for the infinite-span pitching wing. The results obtained in the last period of the simulations with two time steps are shown along with the ones obtained without the prediction step. Moreover the 2d results and the ones provided by [81] are shown.

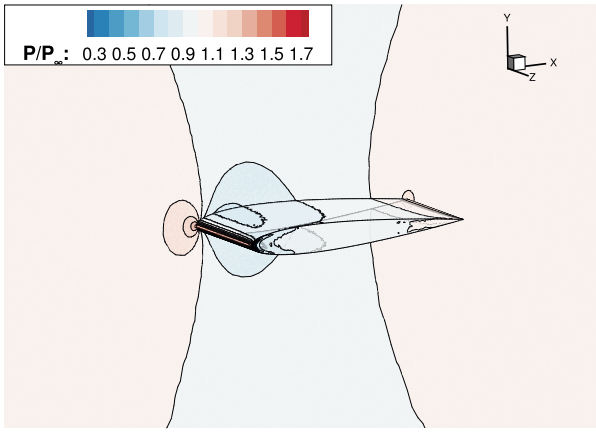

Fig. 21. Pressure contour plot on plane $z = 0$ and over the wing for the steady simulation of the finite-span NACA 0012 wing.

5.2. Unsteady simulation of infinite-span NACA 0012 wing in the laboratory frame

The numerical experiment presented in the previous subsection is now repeated in the laboratory frame; namely, the wing moves through the domain at the flight velocity and air conditions are enforced at the far-field quiescent.

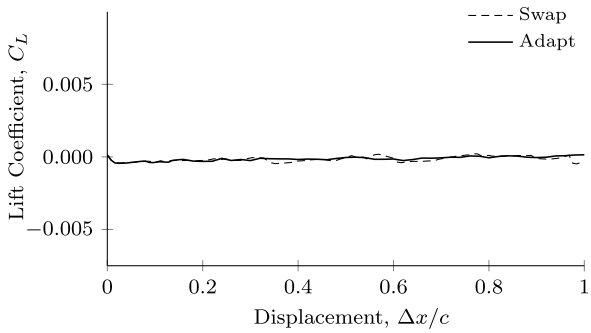

Fig. 22. Lift coefficient for the finite-span NACA 0012 wing in the laboratory frame. The results obtained by exploiting only edge-swapping (*Swap*) and adaptation at each time step (*Adapt*) are compared.

The initial solution is obtained from the steady one by subtracting the free-stream velocity. A displacement of one chord in the negative x-direction at $M_\infty = 0.755$ is imposed to the wing during the simulation. The whole simulation time is divided into 125 time steps Δt_1, resulting in a Courant number of 2 on the minimum grid edge, that is $h_{\min} = 0.004$.

Before introducing the results of the adaptive ALE scheme, it should be considered that it was not possible to solve this problem, i.e. to accomplish the whole displacement, on a fixed connectivity grid, even with a time step ten times smaller than Δt_1.

In the first test, labeled as *Enhance*, mesh optimization is exploited only during those time steps in which elastic analogy fails in following the motion of the wing. Fig. 11 shows the grids and the pressure contour lines at the beginning of the simulation and when one third, two thirds and the whole displacement are carried out. The scheme proves to able to deal with the large movement of the wing but leads to a coarsening of the grid and to a slight deterioration of the solution. The only aim of the optimization in this test is the enhancement of mesh quality according to Definition (18). Indeed, as Table 3 shows, the average, the worst and the best element quality are improved thanks to the re-meshing. However, because of mesh deformation at fixed connectivity, smaller elements, initially gathered in the crucial regions of the grid, are prone to become badly shaped and consequently to collapse. This degradation occurs mainly in the region adjacent to the wing leading and trailing edge, where the average size of the grid elements at the end of the displacement is larger than the one on the initial grid. A similar test, labeled in the following *Swap*, is performed in which mesh quality is restored only by means of edge swapping, thus no node insertions or deletions are exploited.

Then, the complete adaptive computational procedure (*Adapt* test) is adopted, using an error estimator base on the Hessian of the pressure and of the vorticity. As it can be noticed from Fig. 12, the adaptive process based also on the solution leads to a refinement near the trailing and leading edge, and the adaptive solutions show less oscillations in the pressure field with respect to the test *Enhance*. The differences in the solution at diverse time levels are negligible and the steady nature of the problem is well preserved despite the unsteady simulation.

Finally, Fig. 13 shows a quantitative comparison of the results obtained in the unsteady simulations *Enhance*, *Adapt* and *Swap*, with two different values of time step: the already introduced Δt_1 and Δt_2, which corresponds to a Courant number of 5. Sub-Fig. 13(a) displays the variations of the lift coefficient, which remain limited—approximately within ± 0.0015 with respect to the steady computed value of 0.00327—in the *Enhance* and even less in the *Adapt* simulations, where thanks to the mesh adaptation the grid spacing is modified to improve solution accuracy. Conversely, in the *Swap* tests the results degenerate in the second half because the edge-swapping technique alone is not able to adapt the grid in a proper way. No significant differences between different time steps are observed, especially in the adaptive tests.

Sub-Fig. 13(b) displays the variation in time of the integral value of the density over the domain, computed by a trapezoidal rule on each finite volume. Also in this case, the variations around the initial value are very limited, below the 0.25%. The initial increase that can be observed in the *Adapt* tests is due to the adaptation that occurs on far-field, which is represented in the initial grid as a circle. Since an accurate representation of this boundary is not required during this simulation, the adaptation parameter for the corresponding Hausdorff criterion is relaxed and, according to the smooth behavior of the solution in this region, the deletion of some nodes on this boundary lead to a slightly modification of the domain volume. This de-refinement occurs in this test and not in the steady one, since in unsteady tests more importance has to be placed on the coarsening process to prevent an excessively increase in the grid nodes [79]. However, to assess more clearly the conservativeness of the proposed approach, both *Adapt* simulations are performed also not allowing coarsening on the far-field. In this way the integral value of the density over the domain is not affected by the variation in the volume domain and remains approximately constant (variation less than 0.001%) during the simulation, as it can be evinced in Fig. 14.

Finally, to assess the fulfillment of the DGCL condition, the error committed at each time step in satisfying the discrete counterpart of Equation (6) is reported in Fig. 15, along with the error on the closure of the domain. As expected, thanks to the computation of the interface velocities described in Subsection 2.4, these errors can be considered null, assessing the capability of the scheme to automatically fulfilled the DGCL. For the sake of completeness, the error quantities shown in Fig. 15 are computed at each time step at the end of the adaptation process according to the following definitions:

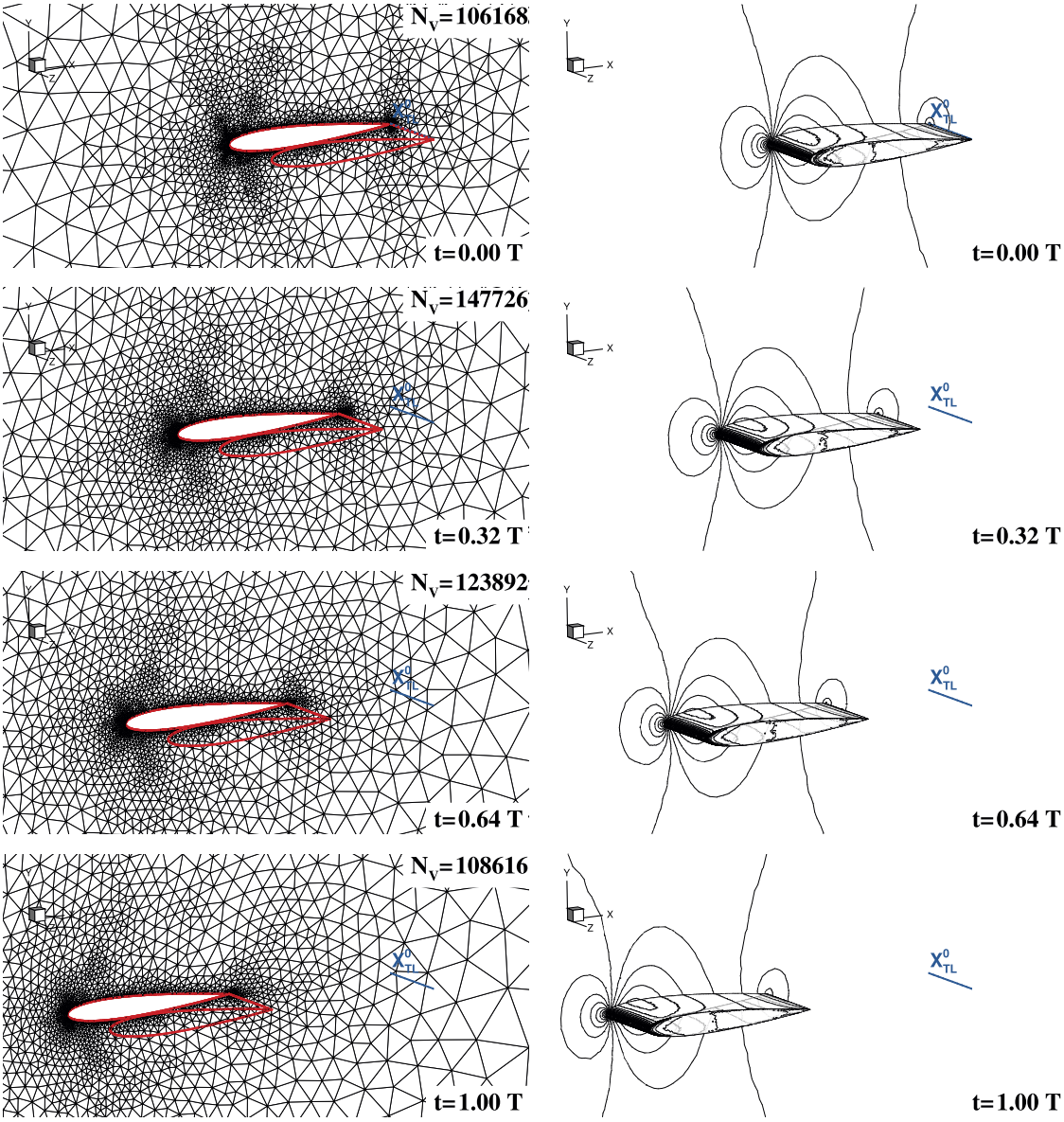

Fig. 23. Grids and pressure contour lines over the wall at $z = 0$ and over the wing surface at the beginning, at 33%, at 66% and at the end of one chord displacement for the finite-span wing. For grid plots, the wing is only sketched by drawing in red the edges. The blue line labeled X^0_{TL} indicates the initial position of the trailing edge. In Supplementary Material, Movies S5 and S6 show the variation of the grid and of the pressure during this simulation, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$Err_{\mathrm{DGCL}} = \sqrt{\sum_{i\in\mathcal{K}^{n+1}} \left(-\frac{V_i^{n+1} - V_i^n}{\Delta t} + \sum_{k\in\mathcal{K}_{i,\neq}} \nu_{ik}^{n+1} + \nu_i^{n+1} \right)^2} \Big/ \Omega^{n+1}, \tag{19a}$$

$$Err_{\Omega} = \sqrt{\left(-\frac{\Omega^{n+1} - \Omega^n}{\Delta t} + \sum_{i\in\mathcal{K}_{\partial}^{n+1}} \nu_i^{n+1} \right)^2} \Big/ \Omega^{n+1}, \tag{19b}$$

where Ω represents the volume of the domain.

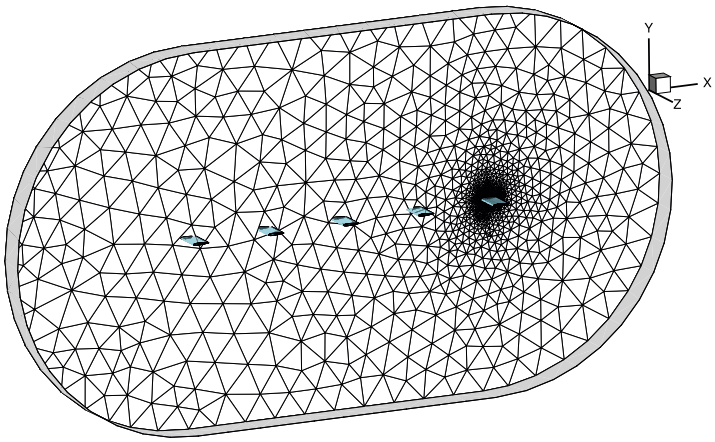

Fig. 24. Initial grid over the wall at $z = 0$ for the 20-chord displacement of the infinite-span wing. Also the wing surface (filled with light blue) at the beginning and after displacements of 5, 10, 15 and 20 chords is shown. To highlight the importance of mesh adaptation, it can be noticed that the grid spacing of the initial grid in the region near $x = -20c$ is approximately equal to one chord. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.3. Transonic flow around the oscillating infinite-span NACA 0012

The results of compressible inviscid simulations around an oscillating NACA 0012 wing are now presented. The pitching motion is prescribed by the sinusoidal function $\alpha(t) = \alpha_\infty + \alpha_0 \sin(\omega t)$, with an initial angle of attach $\alpha_\infty = 0.016°$ and an oscillation amplitude $\alpha_0 = 2.51°$ around the quarter-chord. The reduced frequency is $k = \omega c/u_\infty = 0.1628$, where u_∞ is the free-stream velocity (corresponding to $M_\infty = 0.755$), as in [80].

The steady transonic simulation presented in Subsection 5.1 is used to start the unsteady adaptive simulation, in which four complete oscillation periods are computed with the fully adaptive procedure described in Fig. 7. Two different time-steps are tested, Δt_1 and Δt_2, obtained by subdividing the period time into 100 and 50 steps, respectively. Because of the large time step, the solution changes significantly between two consecutive time iteration and therefore the adaptation cycle is performed at every step, with adaptation criteria based on the Hessian of the pressure and of the vorticity.

Figs. 16 and 17 show the computational grids and the Mach contour lines at different times for the last period of the simulation with Δt_1. Thanks to mesh adaptation the shock that oscillates between the upper and the lower wing surface is well captured. Furthermore, the bi-dimensional character of the flow is correctly reproduced despite the three-dimensional unstructured grid. Fig. 18 highlights this result by showing the Mach contour over the wing surface and some pressure iso-lines at three different sections, namely at $z/b = 0.25$, at $z/b = 0.5$ and at $z/b = 0.75$, for different times. The different wing sections exhibit exactly the same Mach contour and pressure iso-lines. Finally, the variation of the number of grid nodes during the whole simulation is displayed in Fig. 19. The oscillation follows the pitching of the wing and, after the first period, it reaches approximately the same maximum and minimum values. During each period, the two positive peaks correspond to the strongest shock waves on the upper and on the lower surface.

The effect of variation of the time step is finally investigated in terms of the lift coefficient. In Fig. 20, the curves C_L–α obtained with the two different time steps are compared to the results obtained with the same software but over a bi-dimensional grid and to the Euler ones provided by Anderson and Batina [81]. As expected, the curve C_L–α shows the hysteresis due to the phase lag between the variation of the angle of attack and the lift in time [82]. No relevant differences can be observed between the results obtained with different time steps and a good agreement with the bi-dimensional and the inviscid numerical results is achieved.

In addition to the tests previously described, in which a prediction step was always performed, an additional simulation is carried out skipping the prediction phase. The resulting curve C_L–α is also shown in Fig. 20 and it matches closely to the one obtained with the prediction step. However, although only one solution computation is performed at each time step, instead of two as in the simulation with the prediction step, a comparable overall computational time was required. This quite surprising result is possibly due to the fact that, thanks to the prediction step, the computation of the solution on the adapted grid starts from an initial guess closer to the final one, and thus the convergence of the second computation is much faster. Moreover, the importance of the prediction step is confirmed by the impossibility to successfully complete the simulation with the larger time step Δt_2 without this phase. Probably, due to high unsteadiness of the problem, if the larger time step is used, the adaptation is carried out on a mesh that is too different from the actual one and therefore the convergence is extremely difficult to be achieved in the next solution computation.

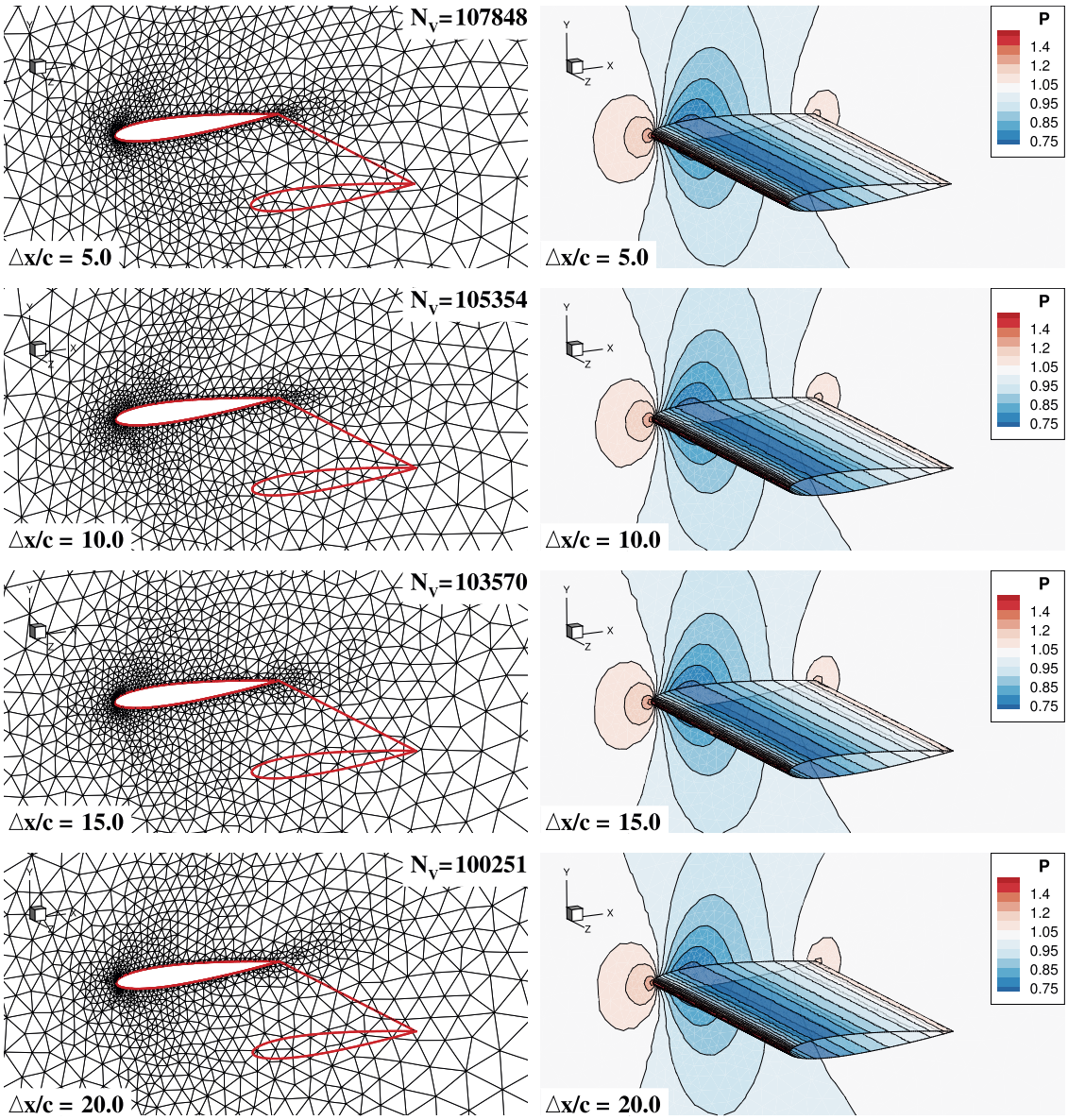

Fig. 25. Grids and pressure contour plots over the wall at $z = 0$ and over the wing surface for the 20-chord displacement of the infinite-span wing. Results are shown at displacements of 5, 10, 15 and 20 chords and the grids can be compared with the initial one shown in Fig. 24. For grid plots, the wing is only sketched by drawing in red the edges. In Supplementary Material, Movies S7 and S8 show the variation of the grid and of the pressure during this simulation, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.4. Unsteady simulation of finite-span NACA 0012 wing in the laboratory frame

Numerical results concerning the flow field around a finite-span wing are now presented. The tests presented in this subsection refer to a NACA 0012 wing with a span of half chord. The geometry is similar to the one presented in Fig. 8, except that the wall opposite to the wing root is at $z = 2c$.

A steady simulation is first performed at Mach $M_\infty = 0.755$ and incidence $\alpha = 0.016°$, by performing five adaptive steps. The final grid contains 106 168 nodes and 588 284 elements. With respect to the infinite-span wing test, a more marked refinement is performed, especially near the trailing edge and the wing tip. Fig. 21 displays the resulting pressure contour plot of the flow field, which is fully three-dimensional because of the circulatory motion that originates near the wing tips on account of the pressure difference between the upper and the lower wing surface. Indeed, also the lift coefficient, see Fig. 22, differs significantly from the one obtained for infinite-span wing at the same flight conditions, mainly because of the induced drag due to the down-wash that occurs near the wing tips.

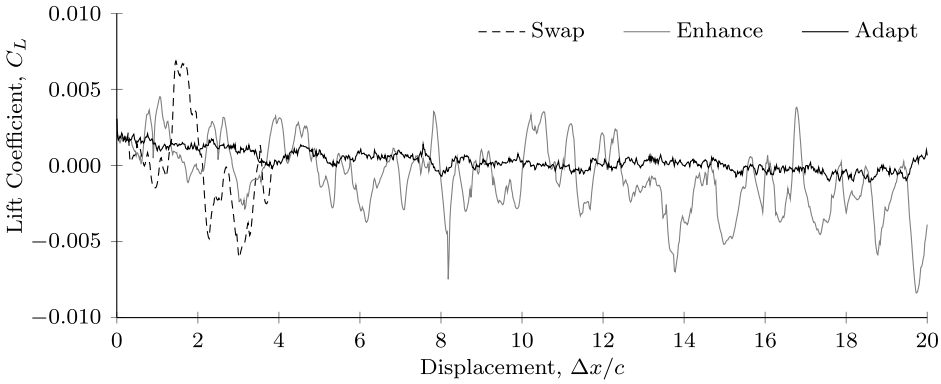

Fig. 26. Lift coefficient for the 20-chord displacement of the infinite-span wing. The results obtained by exploiting only edge-swapping (*Swap*), mesh adaptation only when elastic analogy fails (*Enhance*) and every time step (*Adapt*) are compared. The test *Swap* is not complete, since edge-swapping technique alone is not able to handle such a large displacement.

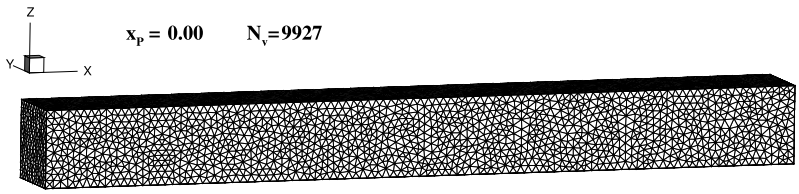

Fig. 27. Initial grid for the piston-induced shock-tube problem. The piston is represented by the left wall and its initial position is $x_P = 0.0$. The domain length along x direction is $L = 1$, while along y and z it is $L_2 = 0.1$.

Then, the same flow field is simulated in the laboratory reference frame by imposing to the wing a displacement of one chord, as described for the infinite-span wing in Subsection 5.2. The time required by the whole movement is divided into 125 time steps and mesh adaptation is performed at each time step. Fig. 23 shows the grid and the solution at different times. The variations over the flow field are limited and only a slight degradation in the solution near the trailing edge can be observed. Fig. 22 displays the variation of the lift coefficient for the described adaptive simulation and also for the one in which only swap is exploited. In both cases, the oscillation amplitude is restricted. Therefore, the capability of the proposed interpolation-free adaptive ALE scheme to deal with large displacement is assessed also by the present test.

5.5. 20-chords displacement of the infinite-span NACA 0012 wing in the laboratory frame

A test similar to the one described in Subsection 5.2 is now performed to assess the capability of the proposed approach to deal with arbitrarily large displacements. More specifically, the infinite-span NACA 0012 wing experiences now a displacement of 20 chords. The initial conditions of the flow field are the same presented in the previous test characterized by 1-chord displacement ($M_\infty = 0.755$ and $\alpha_\infty = 0.016°$), but the computational domain has been enlarged by 20 chords in front of the wing leading edge. The initial grid is shown in Fig. 24, which includes also the positions of the wing after 5, 10, 15 and 20 chords movements.

According to the results of the 1-chord test, only the large time step (Δt_2 in Subsection 5.2) is used, i.e. the whole simulation time is divided in 1000 time steps. With respect to the imposed smallest edge size $h_{\min} = 0.0025$, this choice corresponds to a Courant number of 8. Fig. 25 shows the grid and the solution at different times obtained performing mesh adaptation at each time step. Comparing the grids near the wing surface to the initial grid shown in Fig. 24, it can be observed the important action of mesh adaptation thanks to which the refinement near the wing surface is preserved during the motion, although the grid spacing in that region in the initial grid was much larger. This leads also to negligible differences in the pressure field during the simulations.

The significant improvement obtained thanks to mesh adaptation can be evinced more clearly from Fig. 26 that displays the variation of the lift coefficient during the described adaptive simulation and during the test *Enhance*, in which mesh adaptation is exploited to increase mesh quality only when elastic analogy fails. As expected, adapting the mesh according to both the movement and the solution leads to less oscillations in the results, as shown by the variations in the lift coefficient that remain below 0.003. On the contrary, larger variations (up to 0.1) are obtained in the test *Enhance*. Fig. 26 reports also the partial results of the test *Swap*, in which no nodes are inserted or deleted, but mesh quality is re-stored only through edge-swapping. However, this simulation does not reach the end of the test because the edge-swapping technique alone is not able to provide a sufficient quality improvement when the mesh has been considerably deformed.

This test demonstrates the capability of the proposed adaptive ALE scheme to deal with large displacements, even if a larger time step is used.

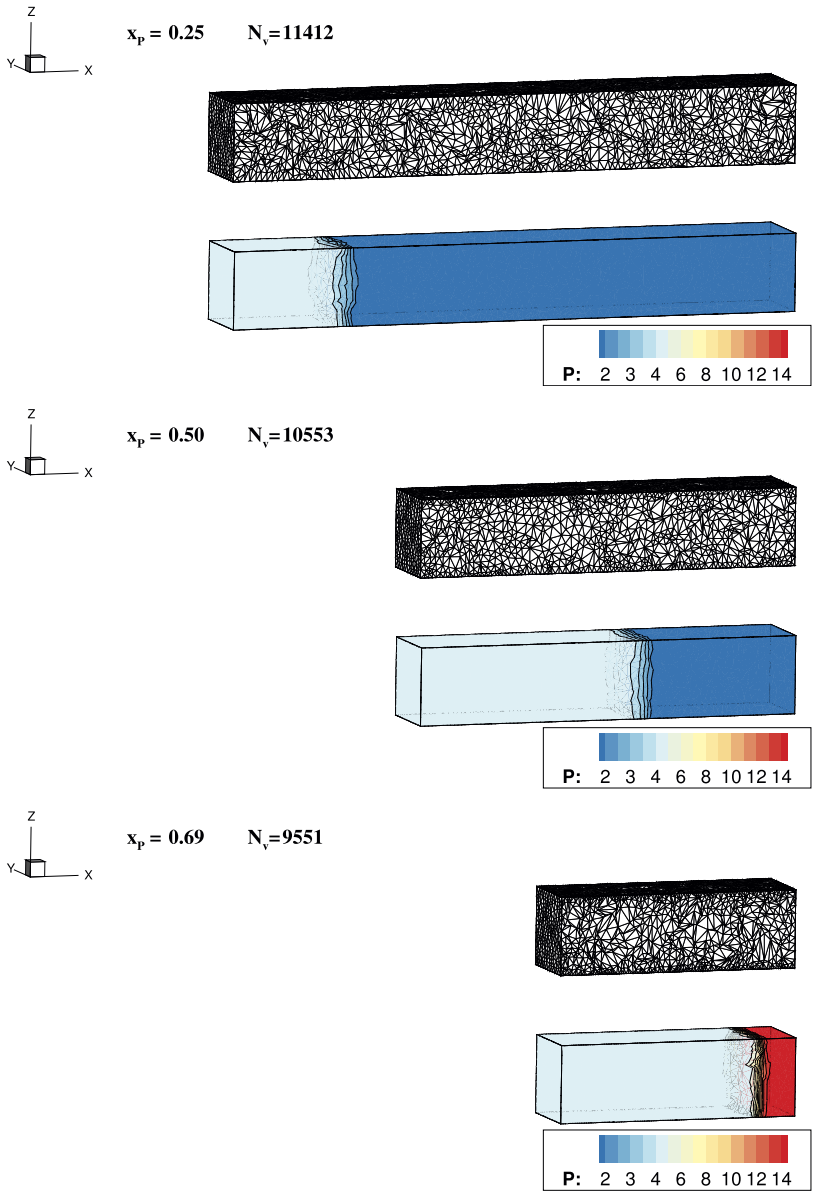

Fig. 28. Grids and pressure contour plots for the piston-induced shock-tube problem for the test *Enhance*. The results corresponding to $x_P = 0.25$, $x_P = 0.5$, $x_P = 0.6875$ are shown. Piston is impulsively started with a constant non-dimensional velocity $V_P = 1.25\sqrt{\gamma}$, the consequent shock travels at Mach $M_s = 2.0$.

5.6. Piston-induced shock-tube problem

This subsection presents the numerical results for a piston-induced shock-tube flow, in which a piston is impulsively started with the non-dimensional velocity $V_P = 1.25\sqrt{\gamma}$. As a consequence, a shock forms at the initial time and travels through the domain with a speed corresponding to a Mach number $M_s = 2.0$, until it reaches the opposite wall ($x = 1$) where it is reflected back with a different intensity and velocity. This test represents a valuable benchmark for the mesh adaptation strategy, because grid refinement is required to correctly capture the shock front, but mesh adaptation is of primary importance also to allow the significant domain deformation. Indeed, the length of the domain at the end of the simulation is less than an half of the initial one, since the simulation is stopped when the position of the piston is $x_P = 0.6875$, after the shock reflection.

Two different numerical tests are carried out: the *Enhance* test where mesh adaptation is used only when elastic analogy fails, and the *Adapt* test in which mesh adaptation according to the solution is exploited at each time step. The whole simulation time is subdivided in 220 time steps, so that the Courant number is 5, considering that a smallest edge size of $h_{\min} = 0.001$ is imposed. The initial grid is shown in Fig. 27.

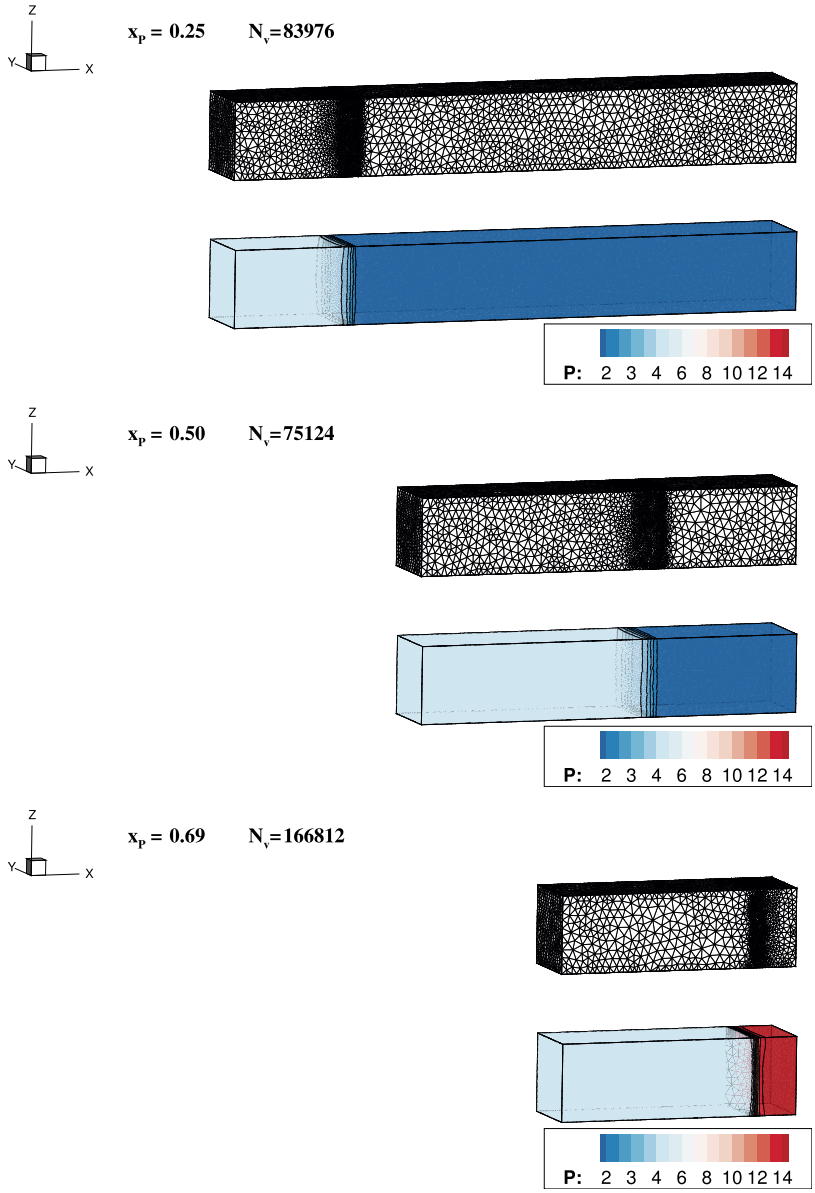

Fig. 29. Grids and pressure contour plots for the piston-induced shock-tube problem for the test *Adapt*. The results corresponding to $x_P = 0.25$, $x_P = 0.5$, $x_P = 0.6875$ are shown. Piston is impulsively started with a constant non-dimensional velocity $V_P = 1.25\sqrt{\gamma}$, the consequent shock travels at Mach $M_s = 2.0$. In Supplementary Material, Movies S9 and S10 show the grid and of the pressure contour plot during this simulation.

The grids and the contour plot of the pressure in both tests are displayed in Figs. 28 and 29. Comparing them, the improvement due to mesh adaptation is significant and it can be observed better in Fig. 30, where the numerical results are compared to the analytical solution. A good agreement, especially for the reflected shock, is achieved when mesh adaptation is exploited.

6. Conclusions

A novel unsteady ALE scheme was proposed to solve the finite volume formulation of the three-dimensional Euler equations in adaptive simulations without explicitly interpolating the solution between different grids. Edge splitting, edge collapse, edge swapping, element splitting and node insertion by Delaunay triangulation are used to locally adapt the grid to better represent local features of the flow field and to comply with the boundary displacement while preserving the quality of the initial grid. These local grid modifications are interpreted as a series of fictitious continuous deformations (expansions and collapses) plus a step in which local grid connectivity is modified when all involved interfaces have null area. As a result, the volume change due to local grid modifications can be included in the standard ALE scheme by additional

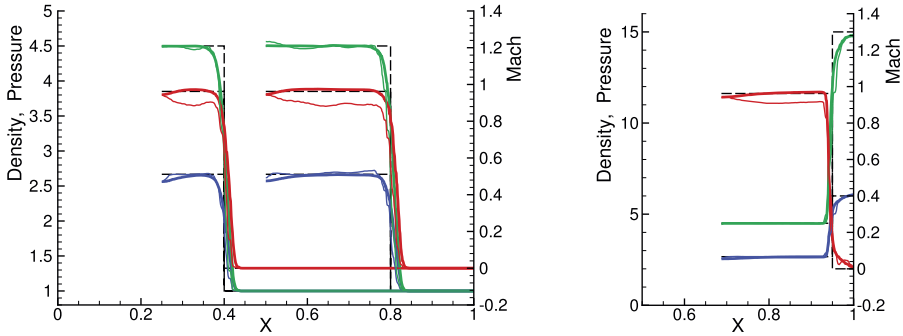

Fig. 30. Solution of the piston-induced shock-tube problem: Mach number (red —), non-dimensional density (blue —) and non-dimensional pressure (green —) along the center-line, i.e. $y = z = 0.05$, at different times. The numerical solutions of the test *Adapt* (thick lines —) and of the test *Enhance* (thin lines —) are compared with the exact one (- -); Left: solution when the piston position is $x_P = 0.25$ and $x_P = 0.5$. Right: the solution after the shock reflection, at $x_P = 0.6875$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

fictitious fluxes and the interface velocities can be computed so that the Discrete Geometric Conservation Law is fulfilled even if the grid topology varies.

Thanks to the proposed interpretation of grid modifications, the conservation of flow variables is inherently guaranteed by scheme construction and the implementation of high-order multi-step time schemes poses no problems because the solutions at previous time levels can be retrieved easily, without resorting to interpolation among grids at different time levels. However, particular care has to be taken to include in the governing equations the contribution of the removed finite volumes, since the non-null volumes swept by their interfaces are required for the computation of the solution on the interacting nodes during the next time step.

The validity of the proposed interpolation-free adaptive ALE scheme was assessed by some preliminary simulations of moving-boundary problems, during which moderate and large boundary deformations were successfully tackled and the grid spacing was properly modified to obtain a good solution accuracy. The automatic fulfillment of the DGCL is numerically proven, as also the mass conservation.

Furthermore, the proposed approach has a general character, since it is based on a standard edge-structured which disregards any information about elements during the solution of the flow equations. Therefore, no additional difficulties are expected to extend the approach to different kinds of grid elements and to hybrid grids. Three-dimensional mesh adaptation for unstructured non-tetrahedral grid is left for future studies. In a similar way, the proposed approach is not limited to error estimators based only on first and second order derivatives of flow variables, which have been exploited in this work for their simplicity but may be not suitable to control the error on the quantities of interest in some applications. For instance, if an output-based error estimator can be obtained from the solution of the adjoint problem, appropriate local mesh adaptation techniques can be adopted to reduce the estimated error, once this information has been localized to the grid elements. Moreover, according to the described key points, the three-steps procedure can be possibly extended to additional local mesh adaptation techniques not considered in the present paper.

As a final remark, it should be pointed out that the current stage of development of the numerical code does not allow us to perform simulations of complex three-dimensional unsteady problems within an acceptable computational time. To this end, an optimization of the whole numerical strategy is under way, including the parallelization of both the solver and the grid adaptation step. Once this issue is solved, the proposed approach can be exploited also in viscous flow simulations and in other fields, where standard ALE approaches are widely used. Indeed, once the local modification has been properly described within the three-steps procedure, the solution on the new grid can be recovered by means of standard ALE schemes.

Funding sources

This research is partially supported by the European Research Council (ERC) [Consolidator Grant N. 617603, Project NSHOCK, funded under the FP7-IDEAS-ERC scheme].

Appendix A. Supplementary material

Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.jcp.2017.03.034.

References

[1] C. Hirt, A.A. Amsden, J. Cook, An arbitrary Lagrangian–Eulerian computing method for all flow speeds, J. Comput. Phys. 14 (3) (1974) 227–253.
[2] J. Donea, S. Giuliani, J. Halleux, An arbitrary Lagrangian–Eulerian finite element method for transient dynamic fluid–structure interactions, Comput. Methods Appl. Mech. Eng. 33 (1) (1982) 689–723, http://dx.doi.org/10.1016/0045-7825(82)90128-1.
[3] M. Lesoinne, C. Farhat, Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations, Comput. Methods Appl. Mech. Eng. 134 (1996) 71–90.

[4] C. Farhat, M. Lesoinne, P. LeTallec, Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: momentum and energy conservation, optimal discretization and application to aeroelasticity, Comput. Methods Appl. Mech. Eng. 157 (1998) 95–114.
[5] C. Farhat, P. Geuzaine, C. Grandmont, The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids, J. Comput. Phys. 174 (2001) 669–694, http://dx.doi.org/10.1006/jcph.2001.6932.
[6] K. Stein, T. Tezduyar, R. Benney, Mesh moving techniques for fluid–structure interactions with large displacements, ASME J. Appl. Mech. 70 (1) (2003) 58–63, http://dx.doi.org/10.1115/1.1530635.
[7] A. Masud, M. Bhanabhagvanwala, R.A. Khurram, An adaptive mesh rezoning scheme for moving boundary flows and fluid–structure interaction, Comput. Fluids 36 (2007) 77–91, http://dx.doi.org/10.1016/j.compfluid.2005.07.01.
[8] S. Basting, A. Quaini, S. Čanić, R. Glowinski, Extended ALE method for fluid–structure interaction problems with large structural displacements, J. Comput. Phys. 331 (2017) 312–336.
[9] J.S. Peery, D.E. Carroll, Multi-material ALE methods in unstructured grids, Comput. Methods Appl. Mech. Eng. 187 (2000) 591–619.
[10] A.J. Barlow, A compatible finite element multi-material ALE hydrodynamics algorithm, Int. J. Numer. Methods Fluids 56 (2008) 953–964, http://dx.doi.org/10.1002/fld.1593.
[11] P.-H. Maire, J. Breil, S. Galera, A cell-centred arbitrary Lagrangian–Eulerian (ALE) method, Int. J. Numer. Methods Fluids 56 (8) (2008) 1161–1166, http://dx.doi.org/10.1002/fld.1557.
[12] M. Berndt, J. Breil, S. Galera, M. Kucharik, P.-H. Maire, M. Shashkov, Two-step hybrid conservative remapping for multimaterial arbitrary Lagrangian–Eulerian methods, J. Comput. Phys. 230 (2011) 6664–6687.
[13] M. Kucharik, M. Shashkov, Conservative multi-material remap for staggered multi-material Arbitrary Lagrangian–Eulerian methods, J. Comput. Phys. 258 (2014) 268–304, http://dx.doi.org/10.1016/j.jcp.2013.10.050.
[14] A.L. Ortega, G. Scovazzi, A geometrically-conservative, synchronized, flux-corrected remap for arbitrary Lagrangian–Eulerian computations with nodal finite elements, J. Comput. Phys. 230 (2011) 6709–6741.
[15] J. Waltz, N. Morgan, T. Canfield, M. Charest, L. Risinger, J. Wohlbier, A three-dimensional finite element arbitrary Lagrangian–Eulerian method for shock hydrodynamics on unstructured grids, Comput. Fluids 92 (2014) 172–187, http://dx.doi.org/10.1016/j.compfluid.2013.12.021.
[16] N.R. Morgan, J.I. Waltz, D.E. Burton, M.R. Charest, T.R. Canfield, J.G. Wohlbier, A point-centered arbitrary Lagrangian Eulerian hydrodynamic approach for tetrahedral meshes, J. Comput. Phys. 290 (2015) 239–273, http://dx.doi.org/10.1016/j.jcp.2015.02.024.
[17] P. Knupp, L.G. Margolin, M. Shashkov, Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods, J. Comput. Phys. 176 (2002) 93–128, http://dx.doi.org/10.1006/jcph.2001.6969.
[18] M. Kucharik, M. Shashkov, B. Wendroff, An efficient linearity-and-bound-preserving remapping method, J. Comput. Phys. 188 (2) (2003) 462–471.
[19] L. Margolin, M. Shashkov, Second-order sign-preserving conservative interpolation (remapping) on general grids, J. Comput. Phys. 184 (1) (2003) 266–298, http://dx.doi.org/10.1016/S0021-9991(02)00033-5.
[20] R. Garimella, M. Kucharik, M. Shashkov, An efficient linearity and bound preserving conservative interpolation (remapping) on polyhedral meshes, Comput. Fluids 36 (2007) 224–237, http://dx.doi.org/10.1016/j.compfluid.2006.01.014.
[21] R. Liska, M. Shashkov, P. Váchal, B. Wendroff, Optimization-based synchronized flux-corrected conservative interpolation (remapping) of mass and momentum for arbitrary Lagrangian–Eulerian methods, J. Comput. Phys. 229 (2010) 1467–1497, http://dx.doi.org/10.1016/j.jcp.2009.10.039.
[22] X. Zeng, G. Scovazzi, A frame-invariant vector limiter for flux corrected nodal remap in arbitrary Lagrangian–Eulerian flow computations, J. Comput. Phys. 270 (2014) 753–783, http://dx.doi.org/10.1016/j.jcp.2014.03.054.
[23] P. Geuzaine, C. Grandmont, C.H. Farhat, Design and analysis of ALE schemes with provable second-order time-accuracy for inviscid and viscous flow simulations, J. Comput. Phys. 191 (1) (2003) 206–227, http://dx.doi.org/10.1016/S0021-9991(03)00311-5.
[24] D.J. Mavriplis, Z. Yang, Construction of the discrete geometric conservation law for high-order time-accurate simulations on dynamic meshes, J. Comput. Phys. 213 (2006) 557–573.
[25] G. Carré, S.D. Pino, B. Després, E. Labourasse, A cell-centered Lagrangian hydrodynamics scheme on general unstructured meshes in arbitrary dimension, J. Comput. Phys. 228 (2009) 5160–5183.
[26] D. Burton, N. Morgan, T. Carney, M. Kenamond, Reduction of dissipation in Lagrange cell-centered hydrodynamics (CCH) through corner gradient reconstruction (CGR), J. Comput. Phys. 299 (2015) 229–280.
[27] J. Cheng, C.-W. Shu, A high order ENO conservative Lagrangian type scheme for the compressible Euler equations, J. Comput. Phys. 227 (2) (2007) 1567–1596.
[28] W. Boscheri, M. Dumbser, A direct arbitrary-Lagrangian–Eulerian ADER-WENO finite volume scheme on unstructured tetrahedral meshes for conservative and non-conservative hyperbolic systems in 3D, J. Comput. Phys. 275 (2014) 484–523.
[29] H. Guillard, C. Farhat, On the significance of the geometric conservation law for flow computations on moving meshes, Comput. Methods Appl. Mech. Eng. 190 (2000) 1467–1482.
[30] L. Formaggia, F. Nobile, Stability analysis of second-order time accurate schemes for ALE–FEM, Comput. Methods Appl. Mech. Eng. 193 (39–41) (2004) 4097–4116, http://dx.doi.org/10.1016/j.cma.2003.09.028.
[31] S. Étienne, A. Garon, D. Pelletier, Perspective on the geometric conservation law and finite element methods for ALE simulations of incompressible flow, J. Comput. Phys. 228 (7) (2009) 2313–2333, http://dx.doi.org/10.1016/j.jcp.2008.11.032.
[32] R. Loubère, P.-H. Maire, M. Shashkov, J. Breil, S. Galera, ReALE: a reconnection-based arbitrary-Lagrangian–Eulerian method, J. Comput. Phys. 229 (2010) 4724–4761.
[33] R. Anderson, N. Elliott, R. Pember, An arbitrary Lagrangian–Eulerian method with adaptive mesh refinement for the solution of the Euler equations, J. Comput. Phys. 199 (2004) 598–617, http://dx.doi.org/10.1016/j.jcp.2004.02.021.
[34] J. Morrell, P. Sweby, A. Barlow, A cell by cell anisotropic adaptive mesh ALE method, Int. J. Numer. Methods Fluids 56 (2008) 1441–1447, http://dx.doi.org/10.1002/fld.1599.
[35] V. Springel, E pur si muove: galilean-invariant cosmological hydrodynamical simulations on a moving mesh, Mon. Not. R. Astron. Soc. 401 (2010) 791–851.
[36] R. Löhner, Mesh adaptation in fluid mechanics, Eng. Fract. Mech. 50 (5/6) (1995) 819–847.
[37] T.J. Baker, Mesh adaptation strategies for problems in fluid dynamics, Finite Elem. Anal. Des. 25 (3–4) (1997) 243–273, http://dx.doi.org/10.1016/S0168-874X(96)00032-7.
[38] W.G. Habashi, J. Dompierre, Y. Bourgault, D. Ait-Ali-Yahia, M. Fortin, M.-G. Vallet, Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. Part I: General principles, Int. J. Numer. Methods Fluids 32 (2000) 725–744.
[39] K.J. Fidkowski, D.L. Darmofal, Review of output-based error estimation and mesh adaptation in computational fluid dynamics, AIAA J. 49 (4) (2011) 673–694.
[40] O. Hassan, E. Probert, K. Morgan, N. Weatherill, Unsteady flow simulation using unstructured meshes, Comput. Methods Appl. Mech. Eng. 189 (2000) 1247–1275.
[41] O. Hassan, K. Morgan, N. Weatherill, Unstructured mesh methods for the solution of the unsteady compressible flow equations with moving boundary components, Phil. Trans. R. Soc. A 365 (1859) (2007) 2531–2552.
[42] C. Dobrzynski, P. Frey, Anisotropic delaunay mesh adaptation for unsteady simulations, in: Proceedings of the 17th International Meshing Roundtable, Springer, 2008, pp. 177–194.

[43] G. Compère, J.J. Jean-François Remacle, J. Hoffman, A mesh adaptation framework for dealing with large deforming meshes, Int. J. Numer. Methods Eng. 82 (2010) 843–867.
[44] R. Wang, P. Keast, P. Muir, A comparison of adaptive software for 1D parabolic PDEs, J. Comput. Appl. Math. 169 (1) (2004) 127–150.
[45] R. Loubère, P.-H. Maire, M. Shashkov, ReALE: a reconnection arbitrary-Lagrangian–Eulerian method in cylindrical geometry, Comput. Fluids 46 (1) (2011) 59–69, http://dx.doi.org/10.1016/j.compfluid.2010.08.024.
[46] B. Perot, R. Nallapati, A moving unstructured staggered mesh method for the simulation of incompressible free-surface flows, J. Comput. Phys. 184 (2003) 192–214.
[47] J. Liu, A second-order changing-connectivity ALE scheme and its application to FSI with large convection of fluids and near contact of structures, J. Comput. Phys. 304 (2016) 308–423.
[48] R. Löhner, Three-dimensional fluid–structure interaction using a finite element solver and adaptive remeshing, Comput. Syst. Eng. 1 (2–4) (1990) 257.
[49] R. Löhner, J. Baum, Three-dimensional store separation using a finite element solver and adaptive remeshing, in: 29th Aerospace Sciences Meeting, vol. 0602, Reno, Nevada, 1991, pp. 1–18.
[50] O. Hassan, K. Sørensen, K. Morgan, N. Weatherill, A method for time accurate turbulent compressible fluid flow simulation with moving boundary components employing local remeshing, Int. J. Numer. Methods Fluids 53 (2007) 1243–1266.
[51] F. Alauzet, A changing-topology moving mesh technique for large displacements, Eng. Comput. 30 (2014) 175–200.
[52] L. Wang, P. Persson, A high-order discontinuous Galerkin method with unstructured space–time meshes for two-dimensional compressible flows on domains with large deformations, Comput. Fluids 118 (2015) 53–68.
[53] G. Olivier, F. Alauzet, A new changing-topology ALE scheme for moving mesh unsteady simulation, in: AIAA Aerospace Sciences Meeting, AIAA 2011-474, Orlando, United States, 2011.
[54] N. Barral, G. Olivier, F. Alauzet, Time-accurate anisotropic mesh adaptation for three-dimensional time-dependent problems with body-fitted moving geometries, J. Comput. Phys. 331 (2017) 157–187.
[55] F. Alauzet, A parallel matrix-free conservative solution interpolation on unstructured tetrahedral meshes, Comput. Methods Appl. Mech. Eng. 299 (2016) 116–142.
[56] D. Isola, A. Guardone, G. Quaranta, Finite-volume solution of two-dimensional compressible flows over dynamic adaptive grids, J. Comput. Phys. 285 (2015) 1–23.
[57] A. Guardone, D. Isola, G. Quaranta, Arbitrary Lagrangian Eulerian formulation for two-dimensional flows using dynamic meshes with edge swapping, J. Comput. Phys. 230 (20) (2011) 7706–7722, http://dx.doi.org/10.1016/j.jcp.2011.06.026.
[58] P. Frey, P.-L. George, Mesh Generation: Application to Finite Element, vol. 32, Wiley, 2010.
[59] C. Dobrzynski, C. Dapogny, P. Frey, A. Froehly, Mmg Platform, /www.mmgtools.org.
[60] R.J. LeVeque, Numerical Methods for Conservation Laws, Birkhäuser, Basel, 1992.
[61] V. Selmin, The node-centred finite volume approach: bridge between finite differences and finite elements, Comput. Methods Appl. Mech. Eng. 102 (1993) 107–138.
[62] V. Selmin, L. Formaggia, Unified construction of finite element and finite volume discretisations for compressible flows, Int. J. Numer. Methods Eng. 39 (1996) 1–32.
[63] A. Harten, High resolution schemes for hyperbolic conservation laws, J. Comput. Phys. 135 (1997) 260–278.
[64] B. van Leer, Towards the ultimate conservative difference scheme II. Monotonicity and conservation combined in a second order scheme, J. Comput. Phys. 14 (1974) 361–370.
[65] P. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys. 135 (2) (1997) 250–258.
[66] D. Muffo, G. Quaranta, A. Guardone, P. Mantegazza, Interface Velocity Consistency in Time-Accurate Flow Simulations on Dynamic Meshes, Tech. Rep. Scientific Report DIA-SR 07-01, Politecnico di Milano, Italy, 2007.
[67] D. Whitaker, Three-Dimensional Unstructured Grid Euler Computations Using a Fully-Implicit, Upwind Method, AIAA Paper 93-3337, 1993, pp. 448–461.
[68] T.J. Barth, D. Jespersen, The design and application of upwind schemes on unstructured meshes, in: 27th Aerospace Sciences Meetings, 1989.
[69] A. Haselbacher, J.J. McGuirk, G.J. Page, Finite volume discretization aspects for viscous flows on mixed unstructured grids, AIAA J. 37 (2) (1999) 177–184, http://dx.doi.org/10.2514/2.711.
[70] A. Jameson, Time Dependent Calculations Using Multigrid, with Applications to Unsteady Flows Past Airfoils and Wings, AIAA paper 1596, 1991, pp. 1–13.
[71] B. Koren, Defect correction and multigrid for an efficient and accurate computation of airfoil flows, J. Comput. Phys. 77 (1988) 183–206.
[72] B. Re, An Adaptive Interpolation-Free Conservative Scheme for the Three-Dimensional Euler Equations on Dynamic Meshes for Aeronautical Applications, Ph.D. thesis, Politecnico di Milano, Department of Aerospace Science and Technology, 2016.
[73] B. Re, A. Guardone, C. Dobrzynski, An adaptive conservative ALE approach to deal with large boundary displacements in three-dimensional inviscid simulations, in: 55th AIAA Aerospace Sciences Meeting, 2017 AIAA SciTech Forum, Grapevine, Texas, 2017.
[74] J.T. Batina, Unsteady Euler airfoil solutions using unstructured dynamic meshes, AIAA J. 28 (8) (1990) 1381–1388.
[75] T. Belytschko, W.K. Liu, B. Moran, Nonlinear Finite Element for Continua and Structures, John Wiley & Sons, 2000.
[76] H. Borouchaki, F. Hecht, P. Frey, Mesh gradation control, Int. J. Numer. Methods Eng. 43 (1998) 1143–1165.
[77] H. Borouchaki, P. Frey, Simplification of surface mesh using Hausdorff envelope, Comput. Methods Appl. Mech. Eng. 194 (2005) 4864–4884.
[78] 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.
[79] D.J. Mavriplis, Unstructured grid techniques, Annu. Rev. Fluid Mech. 29 (1) (1997) 473–514, http://dx.doi.org/10.1146/annurev.fluid.29.1.473.
[80] R. Landon, NACA 0012 Oscillatory and Transient Pitching, Compendium of Unsteady Aerodynamic Measurements, Tech. Rep. R-702, AGARD Report, 1982.
[81] W.K. Anderson, J.T. Batina, Accurate solutions, parameter studies, and comparison for the Euler and potential flow equations, NASA, September 1988, Technical Memorandum 100664.
[82] W. McCroskey, Unsteady airfoils, Annu. Rev. Fluid Mech. 14 (1982) 285–311.