COMPRESSIBLE FLUID-FLOW ALE FORMULATION ON CHANGING TOPOLOGY MESHES FOR AEROELASTIC SIMULATIONS

G. Quaranta , A. Guardone , D. Muffo
Dipartimento di Ingegneria Aerospaziale, Politecnico di Milano,
via La Masa 34, 20158 Milano, Italy

Keywords: *Unsteady Aerodynamics, Fluid-Structure Interactions, Arbitrary Lagrangian Eulerian*

Abstract

The analysis of unsteady fluid flows on moving domains is a very complex task, that may be often tackled using domain remeshing techniques. In the present paper a novel mesh movement strategy is presented. It is based on the blending of simple local edge-swapping with mesh deformation by means of the elastic analogy. To deal with mesh topology changes an extension of the classical Arbitary Lagrangian–Eulerian formulation of the fluid flow equations is developed. In this way the use of interpolations of fluid fields between old and new grid is avoided. Furthermore, this extension allows the easy implementation of high order time integration schemes. Preliminary two dimensional numerical simulations are presented to demonstrate the correctness of the present approach. They shows how this approach guarantees a high quality of the grid without resorting to remeshing, resulting in a very efficient solver useful for the analysis of Fluid–Structure Interaction problems, even for the cases which requires large mesh deformations or changes in the domain topology.

1 Introduction

The interaction between a deformable structure and the fluid flowing around it may lead to instability phenomena (e.g. flutter), which are potentially dangerous for the integrity of the structure itself [5]. In order to predict the behavior of complex aeronautical structures in these conditions, high fidelity CFD models must coupled with the structural models of the aircraft. This branch of continuum mechanics—usually termed Computational Aeroelasticity (CA)—is by definition highly interdisciplinary and since the past two decades it has been a very active and fruitful research area [4, 19, 27]. Two peculiar difficulties arise in the numerical simulation of compressible flows on moving domains. Since the computational domain is continuously changing its shape, a new grid complying with the new domain geometry is to be constructed at each time level. For small displacements it is usually sufficient to slightly reshape or deform the initial grid into the new geometry, without changing the grid connectivity and number of nodes. Very efficient algorithms have been obtained in this case [3, 6, 7]. As a result the computational mesh is moving during the simulation, so the standard finite volume Eulerian formulation of the flow equations must be dropped in favor of the so-called *Arbitrary Lagrangian–Eulerian* (ALE) approach, in which the control volumes are allowed to change their position and shape in time [10].

Indeed, CA is useful for all the cases where the CFD equations are defined in a domain whose shape varies continuously with time due to the movement of the boundaries, like deformable surfaces of the structures. So, equivalent procedures can be employed when dealing with the simulation of rotorcraft, including helicopters

and tilt-rotors, or with problems like store separations. However, in such cases arbitrary large displacements may occur between the different structural parts, and so deformation-based algorithm cannot be used anymore. In fact the mesh obtained by deformation is likely to become too coarse in the direction of movement and the resulting stretched and skewed elements will adversely affect the solution accuracy, or cause the comparison of invalid negative-volume elements.

The current strategy to address these problems is the overset grid technique [23], often known as *Chimera*, which has been successfully applied to rotorcraft simulations [1], of to store separation problems [16,17]. The main drawback of the Chimera approach is that the interpolation between the different grids representing the domain is quite complex and user intervention is often required. As an alternative, the numerical simulation of these kind of flows can be carried out by using remeshing techniques which require the interpolation of the solution over the new grid at each time level [14]. Moreover, multi-step high-order time integration schemes such as the Backward Differencing Formulæ cannot be used due to the different grids used at each time level, unless re-interpolation is used to compute the backward solutions over the current grid. Needless to say, the computational burden of such an approach could be very large.

In the present research, a novel strategy to the simulation of unsteady aerodynamics with moving domain is proposed to circumvent the difficulties of both Chimera and standard re-interpolation techniques. The method is centered around a mesh update technique which guarantees a high quality of the grid even when very large movements are required, without resorting to remeshing, thus resulting in a very efficient computational code. This technique is based on a blend of mesh deformation with local edge-swapping and can be easily applied to two- and three-dimensional problems. The main innovation is obtained noticing that the new grid can be obtained via a suitable deformation of a fictitious grid defined by the current element-node connectivity and the previous node positions. This is possible since the number of grid nodes remains constant as the simulation progresses, so corresponding cells are defined at each time level. By keeping into account correctly the deformation caused by the element that disappears and the new ones created during the edge-swapping, it is possible to compute the solution at the current time level by employing standard ALE techniques, thus avoiding the need to interpolate the flow fields onto a new grid. Moreover, high order time integration schemes, like standard BDF techniques and others, can be implemented very easily.

The present paper reports on preliminary two-dimensional numerical simulations which demonstrate the suitability of the proposed approach. In section 2, the flow solver in the Arbitrary-Langrangian framework is briefly described. The mesh movement technique, including the swapping algorithm, is described in section 3; in section 4, the ALE solver is modified to account for a possibly changing mesh connectivity, which results from the occurrence of edge-swapping. In section 5, numerical results are reported for two-dimensional flows. In section 6, a summary of the present study is given.

2 Edge-based ALE solver for compressible flows

The governing equations for a compressible inviscid fluid in two spatial dimensions are provided by the well-known Euler equations in an Arbitrary Lagrangian Eulerian (ALE) framework [8,9], namely,

$$\frac{d}{dt}\int_{\mathcal{C}(t)} \mathsf{u} + \oint_{\partial\mathcal{C}(t)} \left[\mathbf{f}(\mathsf{u}) - \mathsf{u}\, v\right]\cdot n = 0, \forall \mathcal{C}(t) \subseteq \Omega(t), \tag{1}$$

where u, $\mathsf{u} = (\rho, m, E^{\mathrm{t}})^{\mathrm{T}} \in \mathbb{R}^+ \times \mathbb{R}^3$, is the vector unknown of the density ρ, momentum vector m and total energy per unit volume E^{t}. The solution is sought for in the spatial domain $\Omega \in \mathbb{R}^2$, with boundary $\partial\Omega$ for all times $t \in \mathbb{R}^+$. System (1) is to be made complete by specifying suitable initial and boundary conditions, see e.g. [12]. In Eq. (1), $\mathbf{f} = (\mathsf{f}_x, \mathsf{f}_y)^{\mathrm{T}} \in \mathbb{R}^4 \times \mathbb{R}^2$ is the flux function defined as $\mathbf{f}(\mathsf{u}) = \left(m,\ m \otimes m/\rho + P(\mathsf{u})\ I,\ \left[E^{\mathrm{t}} +\right.\right.$

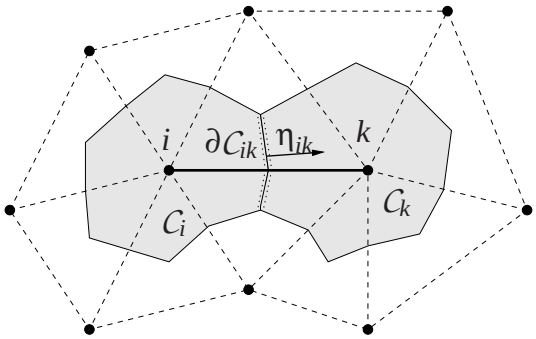

Fig. 1 Edge associated with the finite volume interface $\partial\mathcal{C}_{ik} = \partial\mathcal{C}_i \cap \partial\mathcal{C}_k$ and metric vector η_{ik} (integrated normal) in two spatial dimensions. The two shaded regions are the finite volumes $\mathcal{C}_i$ and $\mathcal{C}_k$; dashed lines indicate the underlying triangulation.

$P(\mathsf{u})\big]\,\rho/m\big)^{\mathrm{T}}$, where I is the 2×2 identity matrix and P is the pressure and where the scalar product in Eq. (1) is computed as $\mathbf{f}(\mathsf{u})\cdot n = \mathsf{f}_x n_x + \mathsf{f}_y n_y$. The vector $n = n(s,t) = (n_x, n_y)^{\mathrm{T}}$ is the outward unit vector normal to the boundary $\partial\mathcal{C}(t)$ of the control volume $\mathcal{C}(t)$ and it is a function of the curvilinear coordinate s along $\partial\mathcal{C}_i$ and of the time as well. The term $\mathsf{u}v = (\rho v, m\otimes v, E^{\mathrm{t}}v)^{\mathrm{T}}$, with $v = v(s,t)$ local velocity of $\partial\mathcal{C}(t)$, accounts for the flux contribution due to the movement of the control volume $\mathcal{C}(t)$.

2.1 Edge-based finite volume solver and Interface Velocity Consistency

The discrete counterpart of the Euler equation (1) is obtained by selecting a finite number of non overlapping volumes $\mathcal{C}_i(t) \subset \Omega(t)$, with boundary $\partial\mathcal{C}_i(t)$, such that $\bigcup_i \mathcal{C}_i(t) \equiv \Omega(t)$. According to the node-centered finite volume approach considered here, each finite volume $\mathcal{C}_i$ surrounds a single node i of the triangulation of Ω, as shown in Figure 1. For each finite volume, equation (1) reads

$$\frac{d[V_i\,\mathsf{u}_i]}{dt} = -\oint_{\partial\mathcal{C}_i}\big[\mathbf{f}(\mathsf{u}) - \mathsf{u}v\big]\cdot n_i, \quad \forall i\in\mathcal{K}, \quad (2)$$

where $V_i = V_i(t)$ is the volume (area in two spatial dimensions) of $\mathcal{C}_i$, $\mathcal{K}$ is the set of all nodes of the triangulation and $n_i = n_i(s,t)$ denotes the outward normal with respect to the volume $\mathcal{C}_i$, see Fig. 1. The unknown u is approximated over $\mathcal{C}_i$ by its average value $\mathsf{u}_i = \mathsf{u}_i(t)$. The right hand side of (2) is now rearranged to put into evidence the boundary contribution, namely,

$$\oint_{\partial\mathcal{C}_i}\big[\mathbf{f}(\mathsf{u}) - \mathsf{u}v\big]\cdot n_i = \sum_{k\in\mathcal{K}_{i,\neq}}\int_{\partial\mathcal{C}_{ik}}\big[\mathbf{f}(\mathsf{u}) - \mathsf{u}v\big]\cdot n_i + \int_{\partial\mathcal{C}_i\cap\partial\Omega}\big[\mathbf{f}(\mathsf{u}) - \mathsf{u}v\big]\cdot n_i, \quad (3)$$

where $\mathcal{K}_{i,\neq} = \{k\in\mathcal{K}, k\neq i | \partial\mathcal{C}_i\cap\partial\mathcal{C}_k \neq \emptyset\}$ is the set of the indexes k of the finite volumes $\mathcal{C}_k$ sharing a portion of their boundary with $\mathcal{C}_i$, $\mathcal{C}_i$ excluded. In the finite volume jargon, the set $\partial\mathcal{C}_{ik} = \partial\mathcal{C}_i\cap\partial\mathcal{C}_k$ is often referred to as the cell interface between the volumes $\mathcal{C}_i$ and $\mathcal{C}_k$ (Figure 1). Each interface $\partial\mathcal{C}_{ik}$ is associated to the corresponding edge *i*-*k* connecting nodes i and k of the triangulation of Ω. A suitable (approximate) integrated numerical flux $\Phi_{ik}\in\mathbb{R}^5$, representing the flux across the cell interface $\partial\mathcal{C}_i\cap\partial\mathcal{C}_k$, is now introduced [15]. Considering a centered approximation of the unknown and of the flux function at the cell interfaces, the domain contributions read

$$\begin{aligned}&\int_{\partial\mathcal{C}_{ik}}\big[\mathbf{f}(\mathsf{u}) - \mathsf{u}v\big]\cdot n_i \simeq \\ &\frac{\mathbf{f}(\mathsf{u}_i)+\mathbf{f}(\mathsf{u}_k)}{2}\cdot\int_{\partial\mathcal{C}_{ik}} n_i - \frac{\mathsf{u}_i+\mathsf{u}_k}{2}\int_{\partial\mathcal{C}_{ik}} v\cdot n_i \\ &= \frac{1}{2}\big[\big(\mathbf{f}(\mathsf{u}_i)+\mathbf{f}(\mathsf{u}_k)\big)\cdot\eta_{ik} - \big(\mathsf{u}_i+\mathsf{u}_k\big)\mathsf{v}_{ik}\big] \\ &= -\Phi_{ik}(\mathsf{u}_i,\mathsf{u}_k,\mathsf{v}_{ik},\eta_{ik}),\end{aligned}$$

where η_{ik} is the integrated outward normal, $\eta_{ik} = \int_{\partial\mathcal{C}_{ik}} n_i$, and where v_{ik} is the *integrated interface velocity*, which satisfies the following *interface velocity consistency* (IVC) condition

$$\mathsf{v}_{ik} = \int_{\partial\mathcal{C}_{ik}} v\cdot n_i. \quad (4)$$

Thanks to the piecewise constant representation of the unknown in the finite volume framework, $\mathsf{u} = \mathsf{u}_i$ over the boundary portion $\partial\mathcal{C}_i\cap\Omega$ and the

boundary contribution in (3) simplifies to

$$\int_{\partial\mathcal{C}_i\cap\partial\Omega}\left[\mathbf{f}(\mathrm{u})-\mathrm{u}\,v\right]\cdot n_i \simeq$$

$$\mathbf{f}(\mathrm{u}_i)\cdot\int_{\partial\mathcal{C}_{ik}} n_i - \mathrm{u}_i\int_{\partial\mathcal{C}_{ik}} v\cdot n_i \tag{5}$$

$$=\mathbf{f}(\mathrm{u}_i)\cdot\xi_i-\mathrm{u}_i\nu_i=-\Phi^{\partial}(\mathrm{u}_i,\nu_i,\xi_i),$$

where ξ_i is the integrated outward normal, $\xi_i = \int_{\partial\mathcal{C}_i\cap\partial\Omega} n_i$, and where ν_i is the integrated interface velocity of the i-th boundary node, defined by the following interface velocity condition

$$\nu_i = \int_{\partial\mathcal{C}_i\cap\partial\Omega} v\cdot n_i\,, \tag{6}$$

It is well known [15] that the use of the second-order integrated numerical flux introduced above may lead to the appearance of spurious oscillations in advection dominated flows and in particular near discontinuities of the flow variables. Following [24], a high-resolution expression for the integrated numerical flux is used in the present work based on the Total Variation Diminishing (TVD) approach. To this purpose, the second order approximation Φ^{II}_{ik} is replaced by the first order Roe [18] flux Φ^{I}_{ik} near flow discontinuities. The switch is controlled by the the limiter of van Leer [24]. Note the the above high-resolution version of the scheme requires the definition of an extended edge data structure that includes also the extension nodes $i^\star$ and $k^\star$, that are needed in the evaluation of the limiter function. As done by [26], the extension nodes belong to the two edges best aligned with i-k. For a general, namely, not centered approximation of the numerical fluxes, one finally obtains

$$\frac{d}{dt}\left[V_i\,\mathrm{u}_i\right] = \sum_{k\in\mathcal{K}_{i,\neq}} \Phi(\mathrm{u}_i,\mathrm{u}_k,\nu_{ik},\eta_{ik}) + \Phi^{\partial}(\mathrm{u}_i,\nu_i,\xi_i), \tag{7}$$

The implementation of the finite volume scheme described above is straightforward and very efficient, see e.g. [20]. All computations are performed only over the edges of the mesh: edges are present in one-, two- and three-dimensional grids and therefore the extension to different spatial dimension requires only few modifications to the code, that are limited to the definition of the vector unknown and to the associated flux function.

2.2 Time integration

The expressions of the average interface and boundary velocity Eq. (4) and Eq. (6) satisfying the IVC conditions are now used to complete system (7) expressing the conservation of mass, momentum and total energy in the ALE framework. This leads to a system of Differential Algebraic Equations (DAE) consisting in $N_{\mathrm{dof}} = N\times(d+2)$ Ordinary Differential Equations (ODE), with N_{dof} the total number of degrees of freedom, N the total number of grid points and $d = 1,2,3$ the number of spatial dimensions, and $N_{ik}+N_{i,\partial}$ algebraic relations, with N_{ik} and $N_{i,\partial}$ number of grid edges and of boundary nodes, respectively. Note that the algebraic equations for the interface velocities ν_{ik} and ν_i are totally uncoupled from the ODE fluid dynamics subsystem, if the coordinates of the grid nodes are known functions of the time. As a consequence, in this case the complete system (7)–(4)–(6) can be reduced to simple ODEs by substitution. This is not the case for example in fully coupled fluid-structure interaction problems in which the node velocity is obtained by solving the grid movement problem which in turn depends from the structural problem solved with the loads coming from the flow field, thus resulting in a coupled system composed by structural, fluid dynamics and mesh deformation problem as well. Moreover, the coupled form of the system is preferred here to stress the existence of a consistency constraint on the interface velocities which leads to an additional system of algebraic (or ODE, cf. system (8)) to be solved together with the ODE system describing the flow dynamics.

The integrated interface velocity represents the derivative of the volume swept by the interface during the mesh movement. So, the system of Differential Algebraic Equations (DAE) can be formally recasted as a system of $N_{\mathrm{dof}}+N_{ik}+$

$N_{i,\partial}$ ODEs as

$$\begin{cases} \dfrac{d}{dt}[V_i\,\mathsf{u}_i] = \displaystyle\sum_{k\in\mathcal{K}_{i,\neq}} \Phi(\mathsf{u}_i,\mathsf{u}_k,\nu_{ik},\eta_{ik}) + \Phi^{\partial}(\mathsf{u}_i,\nu_i,\xi_i), \\ \dfrac{dV_{i,ik}}{dt} = \nu_{ik}\,, \\ \dfrac{dV_{i,\partial}}{dt} = \nu_i\,, \end{cases} \tag{8}$$

where the following definitions have been introduced

$$\frac{dV_{i,ik}}{dt} = \int_{\partial C_{ik}} v\cdot n_i \qquad \text{and} \qquad \frac{dV_{i,\partial}}{dt} = \int_{\partial C_i\cap\partial\Omega} v\cdot n_i. \tag{9}$$

The above ODE system can be solved by means of standard integration techniques, as detailed in the following. In this respect, it is to be noted that the derivatives dV_i/dt, $dV_{i,ik}/dt$ and $dV_{i,\partial}/dt$ are related by the following scalar identity

$$\frac{dV_i}{dt} = \oint_{\partial C_i} v\cdot n_i = \sum_{k\in\mathcal{K}_{i,\neq}} \int_{\partial C_{ik}} v\cdot n_i + \int_{\partial C_i\cap\partial\Omega} v\cdot n_i = \sum_{k\in\mathcal{K}_{i,\neq}} \frac{dV_{i,ik}}{dt} + \frac{dV_{i,\partial}}{dt}. \tag{10}$$

The adoption of IVC conditions (4) and (6) for the computation of the integrated interface velocities implies the automatic fulfillment of Eq. (10) which represents an additional conservation law that must considered when moving and deforming domains are considered: the *volume conservation law*.

The Backward Euler (BE) time discrete counterpart of system (8) is easily obtained as

$$\begin{cases} V_i^{n+1}\,\mathsf{u}_i^{n+1} - V_i^n\,\mathsf{u}_i^n = \\ \quad \left[\displaystyle\sum_{k\in\mathcal{K}_{i,\neq}} \Phi(\mathsf{u}_i,\mathsf{u}_k,\nu_{ik},\eta_{ik})^{n+1} + \Phi^{\partial}(\mathsf{u}_i,\nu_i,\xi_i)^{n+1}\right]\Delta t^n \\ V_{i,ik}^{n+1} - V_{i,ik}^n = \nu_{ik}^{n+1}\,\Delta t^n \\ V_{i,\partial}^{n+1} - V_{i,\partial}^n = \nu_i^{n+1}\,\Delta t^n \end{cases} \tag{11}$$

where all quantities are assumed to be known at time level n and the grid-dependent quantities $V_i^{n+1}, \eta_{ik}^{n+1}$ and ξ_i^{n+1} are computed from the (known) positions of the grid nodes at time level $n+1$. The average interface velocities therefore reads

$$\begin{aligned} \nu_{ik}^{n+1} &= \frac{V_{i,ik}^{n+1} - V_{i,ik}^n}{\Delta t^n} \\ \nu_i^{n+1} &= \frac{V_{i,\partial}^{n+1} - V_{i,\partial}^n}{\Delta t^n} \end{aligned} \tag{12}$$

Note that the following relation, which is the discrete counterparts of (10), holds

$$V_i^{n+1} - V_i^n = \Delta V_i^n = \sum_{k\in\mathcal{K}_{i,\neq}} \Delta V_{i,ik}^n + \Delta V_{i,\partial}^n.$$

The nonlinear system for the fluid variables u at time level $n+1$ is solved here by means of a modified Newton method, in which the Jacobian of the integrated flux function is approximated by that of the first-order scheme, and by resorting to a dual time-stepping technique [25], to improve the conditioning number of the Jacobian matrix.

The expressions of the integrated interface velocities ν_{ik} and ν_i satisfying the IVC condition for a Backward Differences Formulæ (BDF) scheme are now derived. For a nonlinear ODE $dy/dt = f(y,t)$, the second-order BDF scheme with variable time step reads $\alpha_{-1}\Delta y^n + \alpha_0\Delta y^{n-1} = f(y^{n+1},t^{n+1})\,\Delta t^n$, where $\Delta y^{n+1} = y^{n+1} - y^n$ and $\alpha_{-1} = (1+2\beta^n)/(1+\beta^n)$, $\alpha_0 = -(\beta^n)^2/(1+\beta^n)$ with $\beta^n = \Delta t^n/\Delta t^{n-1}$. Therefore, the interface velocities satisfying the IVC conditions in a two-step BDF scheme read

$$\begin{aligned} \nu_{ik}^{n+1} &= \frac{\alpha_{-1}\Delta V_{i,ik}^n - \alpha_0\Delta V_{i,ik}^{n-1}}{\Delta t^n} \\ \nu_i^{n+1} &= \frac{\alpha_{-1}\Delta V_{i,\partial}^n - \alpha_0\Delta V_{i,\partial}^{n-1}}{\Delta t^n}. \end{aligned} \tag{13}$$

Using this approach any generic high order Multi-Step (MS) time integration scheme can be easily implemented. In fact, the above definitions of the interface velocities allows for the IVC condition to be identically satisfied, i.e. the conservation of volume, and, differently from other approaches [21], do not require to modify the coefficients α of the BDF scheme, thus preserving the time accuracy.

3 Mesh movement technique

At each time step the structural elements move either rigidly or elastically depending on the phenomenon under investigation. In any case, the boundary of the aerodynamic domain must be moved following this prescribed motion and the elements on the inner mesh of the domain must be modified accordingly. All these operations must be performed preserving the overall quality of the mesh to reduce numerical errors. The displacement of internal nodes can be obtained using different strategies but for non-structured meshes the more widespread by far are those based on some form of elastic analogy. Batina [3] introduced the elastic analogy by representing each side of the grid as a spring with a nonlinear stiffness proportional to the edge length. To avoid the occurrence of invalid elements with negative volumes, Degand and Farhat [7] introduced additional torsional springs at each vertex.

The grid deformation algorithm presented here extends to idea of the elastic analogy by representing each element as a deformable continuum. Differently from the spring analogy, such a choice make the method less prone to *element breakdown*: the crossing of an element node through a face. To reduce the computational burden, a simple linear constitutive law is implemented, i.e.

$$\sigma = D\varepsilon.$$

When two dimensional grids are investigated, the analogy with the plane strain elastic model is exploited; the stress and strain vectors are $\sigma = \{\sigma_{x_1x_1}, \sigma_{x_2x_2}, \sigma_{x_1x_2}\}^T$, and $\varepsilon = \{\varepsilon_{x_1x_1}, \varepsilon_{x_2x_2}, \varepsilon_{x_1x_2}\}^T$, and the D matrix equal to

$$D = \frac{E}{(1+\nu)(1-2\nu)} \begin{pmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & 1-2\nu \end{pmatrix}.$$

A robust method for grid deformation is achieved adopting a local Young modulus proportional to the minimal dimension of each element, as stated by the following simple law

$$E_{\text{el}} = \frac{1}{\min\limits_{j,k \in \text{el}} \|x_j - x_k\|^\beta}. \qquad (14)$$

In this way small elements close to wall boundaries are stiffer, so they tend to move rigidly with walls, leaving the burden to absorb the required deformations on the larger elements, usually located far from the boundaries. The coefficient β can be used to control the mesh deformation behavior, increasing the stiffness ratio between small and large elements. A Poisson coefficient $\nu \in [0; 0.35]$ is chosen in order to avoid bad numerical conditioning of the problem. Additional details can be found in [6].

When large boundary movements are faced with simple mesh deformations distorted and tangled elements may appear, leading to large numerical errors. For this reason it becomes necessary to adopt a technique to improve the mesh quality without going through global remeshing. In order to avoid any interpolation of the fluid fields between different grids, as explained in the following Section 4, it is essential to keep constant the total number of vertices. A very simple, but extremely effective, technique to improve the quality of triangular meshes, without inserting new vertices, is the edge swapping. The basic idea is to flip the edge of an element, changing the topological structure by local reconnection, without the addition or removal of vertices (see Figure 2). The capability of the swapping to improve the quality of triangular or tetrahedral meshes is well assessed [11]. Edge swapping has been used in connection with deforming meshes [2] where two vertices that move in the opposite direction are allowed to disconnect to avoid excessive stretching of the element. As a result the elements seems to *flow* in the domain allowing bodies to move freely into the computational grid (see Figure 2).

To decide if an edge must be swapped it is necessary to fix a quality measure of elements. Following the analysis presented by Shewchuk [22], what it is most important to improve is the condition number of the stiffness matrix associated with the fluid problem. The element size distribution is controlled by the need to minimize the error bound and it does not have to be fixed by the swapping procedure. So, a scale-invariant quality measure Q has been

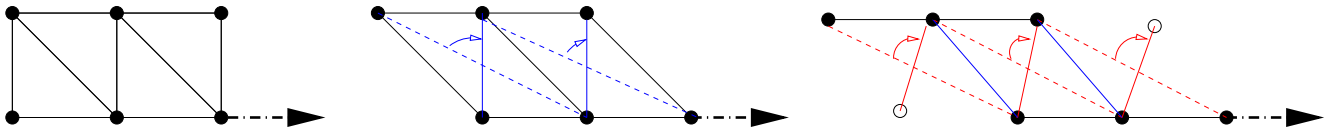

Fig. 2 Edge swapping applied to a deforming grid. A driver force cause the translation of the lower row of vertices; the dashed edge are flipped transforming stretched elements in more regular ones. Blue edges are those created/deleted during the first swapping step, while red lines represent edges created/deleted during the second step. At the end of the process the first vertex on the left of lower row is connected by a new edge with the last edge on right of the upper row.

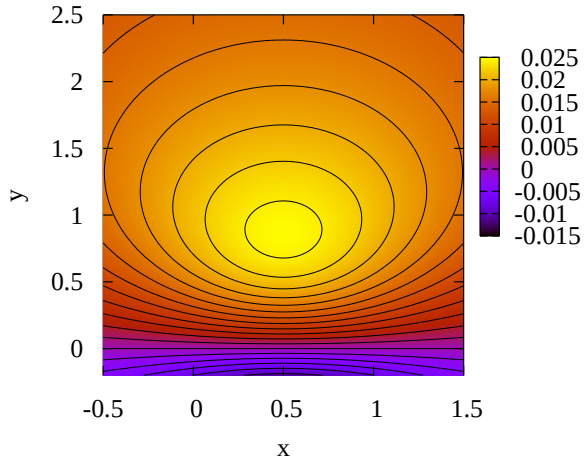

Fig. 3 Element quality measure Q for a triangle of vertices $(0,0)$, $(1,0)$, (x,y). The highest value is associated with the equilateral triangle with the third vertex in $(0.5, \sqrt{(3)}/2)$.

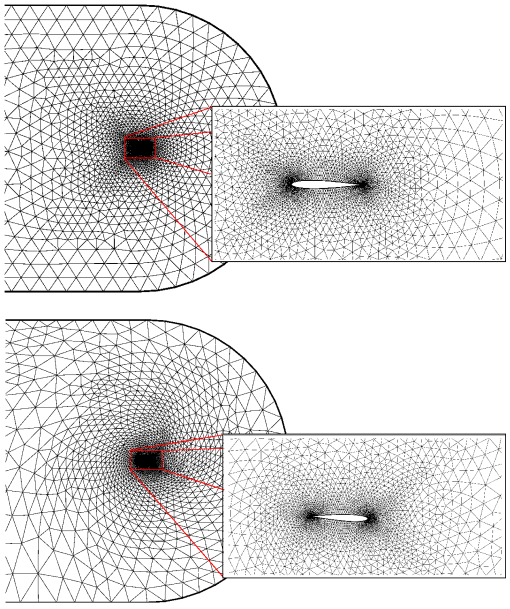

Fig. 4 Mesh movement applied to the 180 deg. rotation of a NACA 0012 profile. The external boundary nodes are kept fixed. Up original configuration 0 deg. ; bottom grid after 180 deg. rotation.

chosen among those presented in [22] for two-dimensional grids, defined as

$$Q = \frac{A}{\sum \ell_i^2 + \sqrt{(\sum {\ell_i}^2)^2 - 48A^2}}, \qquad (15)$$

where A is the element are and ℓ_i the i-th edge length. Figure 3 illustrates the value of quality measure Q for different triangles. Measure Q is signed, which means that is negative if the area is negative, so it is possible to recognize the occurrence of inverted elements.

The complete mesh movement procedure is organized as described in the following pseudo-code:

Grid deformation(displacement of boundary node)
for each edge **do**
  find the associated elements
  Q(current elements)
  Edge swap
  Q(new elements)
  if *min*(Q) increases **then**
    Apply swap
  end
end

The cycle on grid edges can be applied a fixed number of times (one or more), or it can be re-

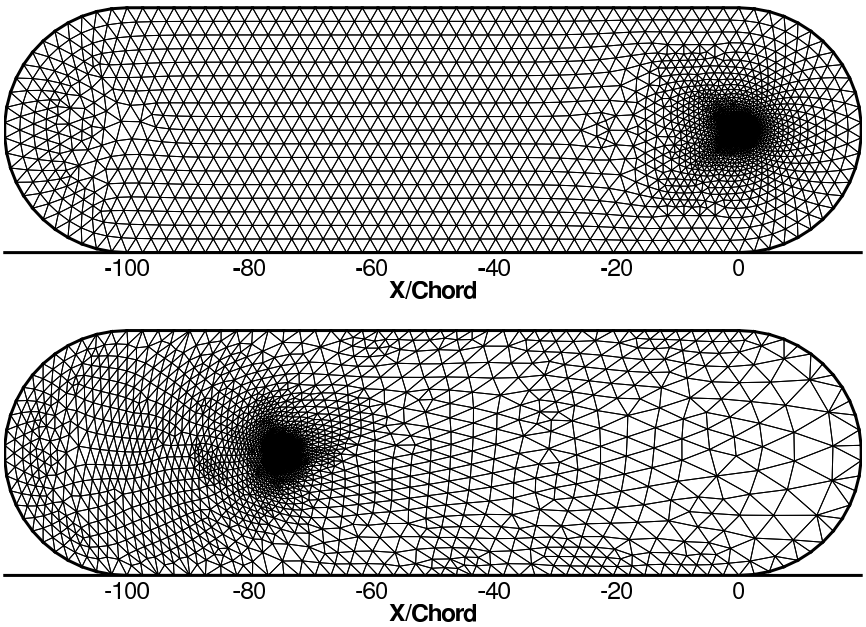

Fig. 5 Mesh movement strategy applied to an 80-chords long forward translation of NACA 0012 profile.

peated up to the case where no additional edge swap is performed.

Figure 4 show the application of this mesh movement scheme to the rotation of a NACA0012 airfoil. The airfoil is rotated while keeping the nodes on the external boundary fixed. Looking at the picture for different rotation angles, it is clear how the inner elements, smaller and so more rigid, follow the movement of the airfoil, keeping the mesh fine where it is still necessary. The edge swapping allows the inner rigid core of elements to slide inside the external grid. Figure 5 shows the same technique applied to the forward translation of the NACA0012 profile by 80 chords, obtained using a sequence of deformation plus swapping steps.

4 ALE scheme for variable grid connectivity

In principle, due to the modification of the grid connectivity resulting from the edge-swapping procedure detailed in the previous section, the solution at time t^{n+1} is to be re-interpolated over the new (modified) grid. In the present section, the idea of interpreting the change in the grid connectivity as a special case of grid deformation is investigated. Two advantages are expected: elimination of possible errors introduced by the flow field interpolation; possibility to use high order schemes for time integration. For simplicity, the Backward Euler scheme is considered first.

In figure 6, the change of the shape of the finite volumes $\mathcal{C}_i$ and $\mathcal{C}_j$ due to the swap of edge *j*-*l* into edge *i*-*k* is depicted. With reference to Figure 6(a), the insertion of edge *i*-*k* can be interpreted as the expansion of a single point over $\partial\mathcal{C}_i$ at time t^n into the interface $\partial\mathcal{C}_{jl}^{n+1}$, which has a non zero measure at time t^{n+1}. The numerical flux associated with edge *i*-*k* at time t^{n+1} reads

$$\Phi_{ik}^{n+1} = \Phi(\mathrm{u}_k^{n+1}, \mathrm{u}_i^{n+1}, \mathrm{v}_{ik}^{n+1}, \mathrm{n}_{ik}^{n+1})$$

where the interface velocity v_{ik}^{n+1} is equal to $\Delta V_{ik}^n/\Delta t^n$, with ΔV_{ik}^n being the shaded area spanned by the interface $\partial\mathcal{C}_{ik}$ during the time step. Similarly, in figure 6(b), the removal of edge *j*-*l* results in the interface $\partial\mathcal{C}_{jl}^n$ collapsing into a single point over $\partial\mathcal{C}_j$ at time t^{n+1}. The numerical flux associated with edge *j*-*l* at time t^{n+1} therefore reads

$$\Phi_{jl}^{n+1} = \Phi(\mathrm{u}_j^{n+1}, \mathrm{u}_l^{n+1}, \mathrm{v}_{jl}^{n+1}, 0)$$

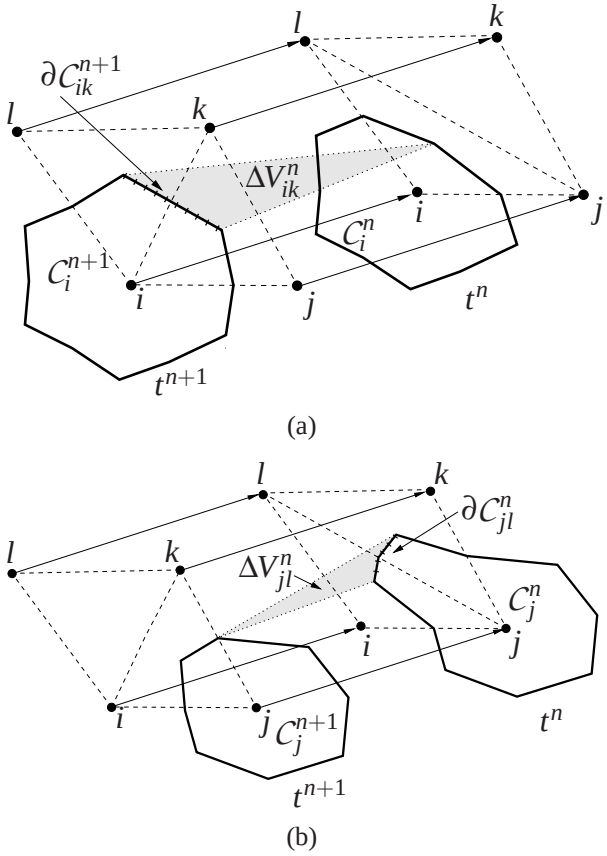

Fig. 6 Computation of the interface velocities in case of edge-swapping. a) Area spanned by edge *i-k*, which is created at time t^{n+1}. b) Area spanned by edge *l-j*, which is removed (swapped) at time t^{n+1}.

where $\eta_{jl}^{n+1} \equiv 0$ since the interface collapses and therefore only the ALE fluxes due to the movement of the interface contribute to the numerical flux across interface ∂C_{jl}^{n+1}. In the expression above, $v_{jl}^{n+1} = \Delta V_{jl}^{n} / \Delta t^n \neq 0$, so there is a finite interface velocity even thought the integrated normal η_{jl}^{n+1} is null. So, it is possible to conclude that edge swapping eventually results in the occurrence of additional fluxes, associated with the grid edges that has been eliminated. The Backward Euler scheme corresponding to (11) therefore reads

$$
\begin{aligned}
V_i^{n+1}\, u_i^{n+1} - V_i^{n}\, u_i^{n} = \\
\Big[\sum_{k \in \mathcal{K}_{i,\neq}^{n+1}} \Phi(u_i, u_k, v_{ik}, \eta_{ik})^{n+1} \\
+ \sum_{k \in (\mathcal{K}_{i,\neq}^{n} - \mathcal{K}_{i,\neq}^{n+1})} \Phi(u_i, u_k, v_{ik}, 0)^{n+1} \\
+ \Phi^{\partial}(u_i, v_i, \xi_i)^{n+1} \Big] \Delta t^n
\end{aligned}
$$

where the set $\mathcal{K}_{i,\neq}^{n} - \mathcal{K}_{i,\neq}^{n+1}$ is the set of the nodes connected to i at time t^n that no longer share a common edge with i at time t^{n+1} as a consequence of edge swapping. In other words, the edges *i-k*, with $k \in (\mathcal{K}_{i,\neq}^{n+1} - \mathcal{K}_{i,\neq}^{n})$ are the new edges, whereas edges *i-k*, with $k \in (\mathcal{K}_{i,\neq}^{n} - \mathcal{K}_{i,\neq}^{n+1})$ are those that were present at time t^n and have been deleted during the edge swapping. A similar procedure allows to extend standard BDF and multistep schemes to the case of moving mesh including edge-swapping — using the same approach shown in Section 2.2 —, without the need of introducing any interpolation of the solution.

It is important to show how the presented approach is in perfect agreement with the volume conservation law expressed by Eq.(10). Figure 7 highlights the changes of cell volumes due to edge swapping. Moving from step 0 to step 1 the edge *i-l* is swapped with edge *m-k*. As a consequence the area of cell $\mathcal{C}_m$ is increased by two factors: the creation of $\partial \mathcal{C}_{mk}$, which is responsible for the volume marked in green; the volumes swept by the movement of the interfaces $\partial \mathcal{C}_{ml}$ and $\partial \mathcal{C}_{mi}$ connected with the newly created interface $\partial \mathcal{C}_{mk}$, marked in red. For the cells $\mathcal{C}_l$ and $\mathcal{C}_i$ the

green volume is subtracted — actually half of it for each cell — since it represents the volume spanned by edge that collapse, and the same is done for the red volumes. Indeed, the devised approach allows to read the effects of edge swapping simply as a deformation of the cells associated with each vertex; in this way the dynamic problem under investigation is kept within the framework of ALE formulation, and so all comes naturally and so no special "tricks" are required to implement high order integration schemes.

Exactly the same procedure applies to the cases where multiple swapping are applied to the same cell. Figure 7 shows the effect of a sequence of two edge flip: *i*-*l* with *m*-*k* followed by *k*-*i* with *m*-*j*. The bottom part of Figure 7 shows how the total variation of the area associated with C_m can be computed by a simple summation of the contributions coming from each swap operation. The computation of contributions associated with each swap step supplies a simple way to compute the total cell volume variation and the associated fluxes from the old grid to the new one.

5 Numerical results

As an example, in Figure 8, the results obtained for an oscillating airfoil flying at Mach number 0.75 are shown. Standard ALE results refer to computations performed on an airfoil-centered reference, with the airfoil slightly varying its pitch in time, see [13]. Numerical simulations obtained with the present method are instead performed on a fixed grid (laboratory reference) through which the airfoil is translating (and oscillating) at Mach number 0.75. The airfoil displacement is as large as 100 airfoil chords. The present results agree fairly well with the available reference ones thus demonstrating the validity of the proposed approach.

6 Conclusions

A novel approach for analysis of unsteady flow in moving domains has been presented. Combining the mesh deformation with the edge swapping it has been possible to deal with cases where large displacement of the boundaries are required. By keeping the number of vertex constant it has been possible to show how all the element topology modification can be interpreted as a change in the cell volume. As a consequence, the usual ALE formulation for moving and deforming grid can be applied without the application of interpolation schemes while moving from one grid to the following configuration. Preliminary results on a simple 2D problem have been shown. Future applications of this technique to more interesting 2D cases (store separation) is under way. The extension of the proposed approach to three-dimensional cases is not expected to show additional problems, since the most complex part, i.e. the edge and face swapping technique for tetrahedrons, is well assessed in the current literature [11].

References

[1] Ahmad J and Duque E. P. N. Helicopter rotor blade computation in unsteady flows using moving overset grids. *Journal of Aircraft*, Vol. 33, No 1, pp 54–60, 1996. AIAA Paper 94-1922.

[2] Baker T. J. Mesh deformation and modification for time dependent problems. *International Journal for Numerical Methods in Fluids*, Vol. 43, No 6-7, pp 747–768, 2003.

[3] Batina J. Unsteady Euler airfoil solution using unstructured dynamic meshes. *AIAA Journal*, Vol. 28, pp 1381–1388, 1990.

[4] Bennet R. M and Edwards J. W. An overview of recent developments in computational aeroelasticity. *Proc Proceedings of the* 29th *AIAA Fluid Dynamics Conference*, Albuquerque, NM, June 15-18 1998.

[5] Bisplinghoff R. L and Ashley H. *Principles of Aeroelasticity*. Wiley & sons, New York, 1962.

[6] Cavagna L, Quaranta G, and Mantegazza P. Application of Navier-Stokes simulations for aeroelastic assessment in transonic regime. *Computers & Structures*, Vol. 85, No 11-14, pp 818–832, 2007. Fourth MIT Conference on Computational Fluid and Solid Mechanics.

[7] Degand C and Farhat C. A three-dimensional torsional spring analogy method for unstruc-

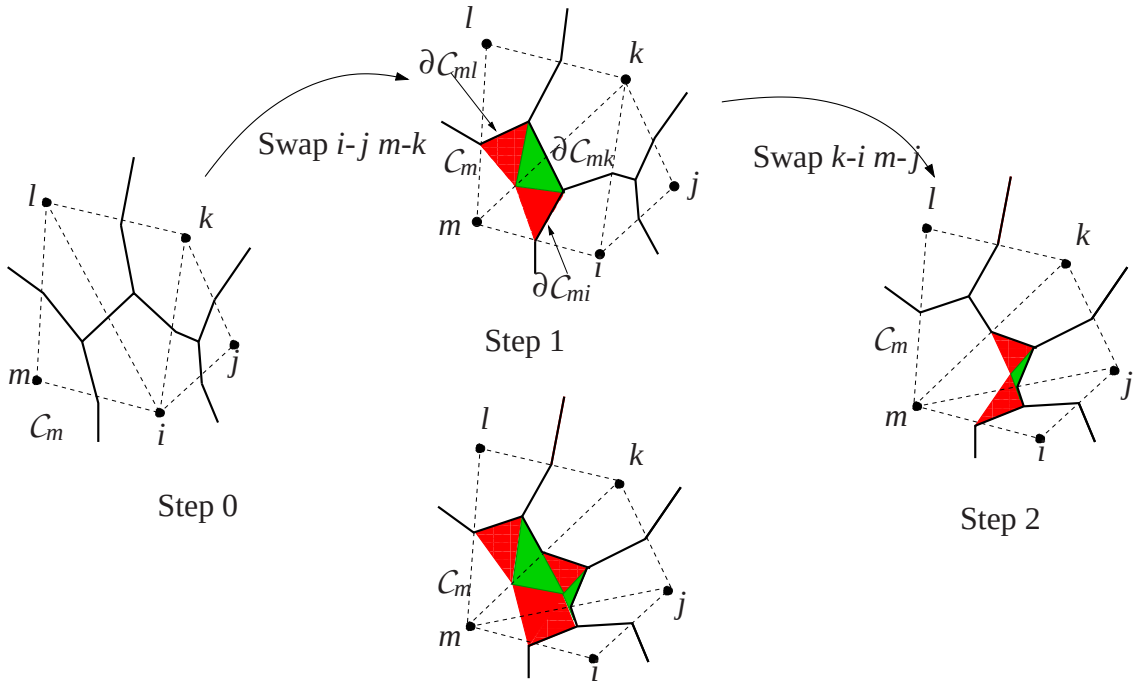

Fig. 7 Effect of edge swapping on the change of cell volume. The green portions represent those due to new interface insertion, while the red portions are those swept by the movement of existing edges. Summing the volumes created during multiple swapping steps it is possible to obtain the total cell area variation shown on the bottom part of the figure.

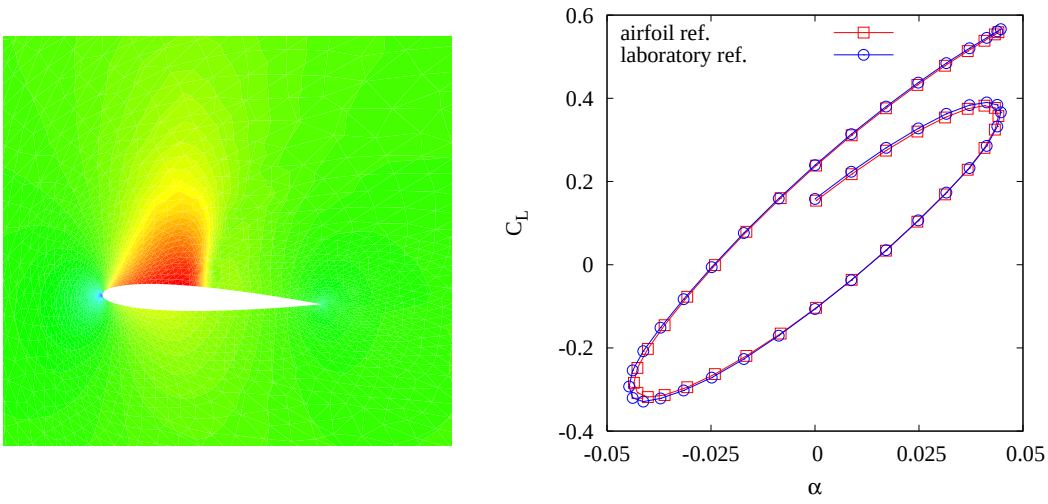

Fig. 8 Left: Mach number M contours in the flowfield at the end of the computation for an oscillating airfoil. Right: C_L-α history computed in the laboratory reference (large grid deformations, airfoil is moving at $M = 0.75$ in a still grid) and in the airfoil reference (the grid is moving at M=0.75).

tured dynamic meshes. *Computers and Structures*, Vol. 80, pp 305–316, 2002.

[8] Donea J. An arbitrary Lagrangian-Eulerian finite element method for transient fluid–structure interactions. *Comput. Methods Appl. Mech. Engrg.*, Vol. 33, pp 689–723, 1982.

[9] Donea J, Huerta A, Ponthot J.-P, and Rodríguez-Ferran A. Arbitrary Lagrangian-Eulerian methods. In Stein R, de Borst E, and Hughes T, editors, *The Encyclopedia of Computational Mechanics*, Vol. 1, pp 413–437. Wiley, 2004.

[10] Donea J. Arbitrary Lagrangian–Eulerian finite element methods. In Belytschko T and Hughes T. J, editors, *Computational Methods for Transient Analysis*, pp 474–516, Amsterdam, The Netherlands, 1983. Elsevier Science Publisher.

[11] Freitag L. A and Ollivier-Gooch C. Tetrahedral mesh improvement using swapping and smoothing. *International Journal for Numerical Methods in Engineering*, Vol. 40, No 21, pp 3979–4002, 1997.

[12] Godlewski E and Raviart P. A. *Numerical approximation of hyperbolic systems of conservation laws*. Springer-Verlag, New York, 1994.

[13] Guardone A, Muffo D, and G.Quaranta. Interface Velocity Consistency for flow simulations on dynamic meshes. *J. Comput. Phys.*, 2008. Sumbitted.

[14] Hassan O, Sørensen K. A, Morgan K, and Weatherill N. P. A method for time accurate turbulent compressible fluid flow simulation with moving boundary components employing local remeshing. *International Journal for Numerical Methods in Fluids*, Vol. 53, No 8, pp 1243–1266, 2007.

[15] LeVeque R. J. *Finite volume methods for conservation laws and hyperbolic systems*. Cambridge University Press, 2002.

[16] Liever P, Habchi S, Engelund W, and Martin J. Stage separation analysis of the X-43A research vehicle. *Proc AIAA* 22nd *Applied Aerodynamics Conference and Exhibit*, Providence, Rhode Island, August 16–19 2004.

[17] Lijewski L and Suhs N. Time-accurate computational fluid dynamics approach to transonic store separation trajectory prediction. *Journal of Aircraft*, Vol. 31, No 4, pp 886–891, 1994.

[18] Roe P. L. Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.*, Vol. 43, pp 357–372, 1981.

[19] Schuster D. M, Liu D. D, and Huttsell L. J. Computational aeroelasticity: Success, progress, challenge. *Journal of Aircraft*, Vol. 40, No 5, pp 843–856, 2003.

[20] Selmin V. The node-centred finite volume approach: bridge between finite differences and finite elements. *Comp. Meths. Appl. Mech. Eng.*, Vol. 102, pp 107–138, 1993.

[21] Selmin V. Coupled fluid/structure system. In Haase W, editor, *Progress in Computational Flow-Structure Interaction*, Vol. 81. Springer-Verlag, Berlin, 2002.

[22] Shewchuk J. What is a good linear element? Interpolation, conditioning, and quality measures. *Proc* 11th *International Meshing Roundtable*, pp 115–126, Ithaca, NY, September 15–18 2002.

[23] Steger J, Dougherty F, and Benek J. A Chimera grid scheme. *Proc Advances in Grid Generation, ASME FED-5*, pp 59–69, 1983.

[24] van Leer B. Towards the ultimate conservative difference scheme II. Monotonicity and conservation combined in a second order scheme. *J. Comput. Phys.*, Vol. 14, pp 361–370, 1974.

[25] Venkatakrishnan V and Mavriplis D. J. Implicit method for the computation of unsteady flows on unstructured grids. *J. Comput. Phys.*, Vol. 127, pp 380–397, 1996.

[26] Weatherill N, Hassan O, Marchant M, and Marcum D. Adaptive inviscid solutions for aerospace geometries on efficiently generated unstructured tetrahedral meshes. AIAA Paper 93-3390, 1993.

[27] Yurkovich R. Status of unsteady aerodynamic prediction for flutter of high-performance aircraft. *Journal of Aircraft*, Vol. 40, No 5, pp 832–842, 2003.

Copyright Statement