Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **5**, No. 2, pp. 194-211

DOI: 10.4208/aamm.11-m1186
April 2013

A Pseudo-Stokes Mesh Motion Algorithm

Luciano Gonçalves Noleto[2,*], Manuel N. D. Barcelos Jr.[2] and
Antonio C. P. Brasil Jr.[1]

[1] *Universidade de Brasília, Faculdade de Tecnologia, Departamento de Engenharia Mecânica,*
Laboratório de Energia e Ambiente. Asa Norte. Brasília-DF, Brasil, 70.910-900
[2] *Universidade de Brasília no Gama. Complexo de Educação, Cultura, Esporte e Lazer.*
Área Especial de Indústria 1, Setor Leste, Faculdade Lote 1. Gama-DF, Brasil,
72.444-210

Received 3 May 2011; Accepted (in revised version) 10 October 2012

Available online 22 February 2013

Abstract. This work presents a moving mesh methodology based on the solution of a pseudo flow problem. The mesh motion is modeled as a pseudo Stokes problem solved by an explicit finite element projection method. The mesh quality requirements are satisfied by employing a null divergent velocity condition. This methodology is applied to triangular unstructured meshes and compared to well known approaches such as the ones based on diffusion and pseudo structural problems. One of the test cases is an airfoil with a fully meshed domain. A specific rotation velocity is imposed as the airfoil boundary condition. The other test is a set of two cylinders that move toward each other. A mesh quality criteria is employed to identify critically distorted elements and to evaluate the performance of each mesh motion approach. The results obtained for each test case show that the pseudo-flow methodology produces satisfactory meshes during the moving process.

AMS subject classifications: 76M10, 65M50, 35R37

Key words: Moving boundaries, mesh motion strategy, finite element projection methods.

1 Introduction

Nowadays for several CFD problems, the fluid domain encloses or is enclosed by moving boundaries. Thus, when the moving boundary is subjected to a large amplitude motion, a moving mesh strategy is necessary. Therefore, a mesh motion algorithm has to be employed throughout a simulation. This algorithm is basically a numerical solver that

[*] Corresponding author.
Email: lucianonoleto@unb.br (L. G. Noleto), manuelbarcelos@unb.br (M. N. D. Barcelos Jr.), brasiljr@unb.br (A. C. P. Brasil Jr.)

computes and updates mesh node displacements and velocities, evaluated at any given time step.

The mesh motion algorithm must fulfill three conditions to be efficient and reliable [10, 11]:

- The algorithm must be compatible with the Geometric Conservation Law;
- The mesh after moving must have good quality;
- The computational effort for mesh motion must be small.

The geometric conservation law states that any change in the element control area or volume between two consecutive time steps must be equal to the area or volume swept by the cell boundary during these time steps [17, 19]. As a consequence, the geometric quantities associated to the moving mesh must be computed in a way that the integration preserves a uniform flow field. Literature shows that the geometric conservation law can be represented by a null mesh velocity divergent constraint [21]. Also, there are cases where the non-fulfillment of this constraint leads to numerical instabilities and as a consequence to wrong flow solutions or spurious results [7, 17].

The first numerical schemes [2, 16] to solve the mesh motion problem modeled the mesh as a fictitious structure where linear springs were placed between the nodes. The stiffness of every linear spring is inversely proportional to the distance between the nodes. The total stiffness of the pseudo structure has the contribution of each linear spring. Due to the model characteristics, the finite element method (FEM) was a natural choice for solving the problem. However, this procedure was limited to problems with small mesh deformations between consecutive time steps because large deformations may lead to element collapsing.

Other mesh motion approaches compute the mesh displacement field by solving a diffusion problem. The displacements are associated to nodal velocity field. The problem formulation is based on the Laplacian of the nodal velocity field. Hence, the Laplace equations are solved, and the mesh displacements are recovered by multiplying the resulting velocity field by a predefined time step. The main drawback of this method is the choice of the right diffusivity coefficient to avoid collapsed elements in the moving mesh process.

Robust mesh motion approaches based on the solution of pseudo structure problems use the superposition of linear and torsional springs [9]. From the beginning, the linear and torsional springs method was developed for two-dimensional triangular elements. Adapting this technique to three-dimensional elements such as tetrahedrons required a repositioning of the torsional spring action plane which avoids the element face and volume collapse [6]. For large mesh deformation, distorted element might still be produced and as a consequence this can lead to numerical stiffness and wrong flow solutions.

In the last few years several works in the literature proposed new mesh motion approaches employing geometrical function interpolation and mapping to tackle large mesh deformation problems. For instance, [28] presented a mesh moving algorithm for dynamic meshes. A Delaunay graph is generated for the domain to represent the moving boundary. The mesh deformation is performed through this graph. The authors present some test cases. The results showed that this methodology is robust and adequate for large deformations and oscillatory movement. Also, [25] presented a moving mesh methodology that employs fixed meshes for different configurations of a moving boundary. In other words, the flow is calculated in a fixed mesh. When the boundary starts to move, the mesh is changed to another configuration. The authors defined this method as a fixed Arbitrary Eulerian-Lagrangean (ALE). The mesh domain is defined by a boundary function that changes the domain during the movement of the boundary. These functions are level-set functions that hold similarities with the immersed boundary method. However, this method is classified as a Cartesian grid method what differentiates it from immersed boundary methods. The authors implemented an algorithm with projection techniques based on a FEM framework. Several test cases were evaluated, one of them is the lost foam casting in non-Newtonian fluids. The results obtained were considered satisfactory.

The work in [29] presented a moving mesh methodology based on unstructured meshes that uses direct and explicit interpolation. The mesh quality is maintained, according to the authors, even for large deformations. A tree-code style technique is used for interpolation optimization. It is also considered as a point-to-point method The tests presented by the authors produced reasonable results. The article [24] investigated the computational cost of a mesh motion algorithm based on the Delaunay graph mapping of the original mesh. This algorithm contemplates large mesh deformations by using a fast-locating technique to optimize node repositioning. Also, the algorithm contemplates advances to determine background regions close to the moving nodes. Triangular meshes were employed in CFD simulations over moving airfoils and moving aircraft parts. The results obtained by the authors showed the computational cost and its decrease by optimizing some aspects of the algorithm.

Reference [26] proposed solutions for some issues concerning mesh motion techniques. One of these issues is how to deal with large mesh deformation. This issue limits the algorithm to small boundary movements, and therefore reduces its applicability. The authors proposed and discussed a mesh motion algorithm suited for large deformations. Some cases were studied: a propeller, two falling spheres and a three-dimensional flow with fluid-structure interaction. The algorithm was able to deal with large deformations of different mesh sizes and refinement. The work in [27] proposed a mesh motion technique base on a reduced domain strategy. The mesh is divided into active and inactive zones, in order to reduce computational costs. The mesh is partitioned and only the nodes close to a moving boundary are moved. Tests with unstructured meshes were performed, yielding good results with low computational cost.

The text in [31] presented a mesh motion algorithm that uses base functions and data reduction. The idea is to allow deformations inside a given tolerance for unstructured meshes. Within that tolerance, fewer nodes are used and interpolation errors are reduced. The method is tailored for smooth deformations that are frequently present in aeroelasticity problems. Good results are obtained with low computational cost. Reference [32] is a sequence of the previous work. At this time, structured meshes are employed and tested by the mesh motion algorithm. Three error functions are employed by the algorithm to optimize the mesh deformation. The results showed that the unitary error function produced the best outcomes. The work in [34] made comparisons among several mesh motion techniques for ALE methodologies. Three techniques are highlighted: a biharmonic partial differential equation, a linear-elastic technique and a harmonic partial differential equation (PDE). All equations are discretized by the Galerkin FEM. This work employed hexahedral meshes, and tests showed that the biharmonic technique yielded precise results, but with higher computational cost.

The work in [30] reviewed mesh motion techniques based on the theory of linear elasticity. These techniques do not contemplate rigid body rotation. The authors modified the linear elasticity technique to be able to perform rigid body rotation. Therefore, the algorithm became more robust, and large deformations could be achieved. A Galerkin FEM is employed to discretize the mesh motion equations. Test cases based on Navier-Stokes CFD solutions were employed. The cases yielded good results, but a real case is needed to prove the efficiency of this technique. Reference [33] presented a node velocity field mesh motion technique. Algorithms based on FEM are employed to discretize the mesh equations. The geometric conservation law is used as a restraint to guarantee good quality mesh after deformation. The tests yielded reasonable results.

The idea of using a nodal velocity approach to solve mesh motion problems is not new [20]. One could understand it as a particular case of a pseudo flow problem. The robustness of its solution can be improved by using a projection FEM as solver. The advantage of this approach is the possibility of obtaining a final mesh configuration with better element quality. Projection methods have the characteristic of project the velocity into a null divergent field [13]. Therefore, this strategy will fulfill the geometric conservation law by default. As a consequence the null divergent constraint may guarantee mesh quality by avoiding collapsed elements and the flow solution smoothness would improve. The main difficulty of this methodology is still in the choice of reasonable diffusion coefficient.

The goal of this paper is to present a mesh motion strategy based on a pseudo Stokes flow problem, discretized by a projection FEM. The main advantages of this technique are the null divergent constraint which guarantees the mesh quality after the motion, the use of existing well known and tested numerical methodologies and algorithms now adapted to solve a mesh motion problem and the possibility to apply to different types of mesh. The results of the pseudo Stokes mesh motion approach are compared to four other traditional strategies: Laplacian Operator based on diffusion problems, Linear Spring Analogy, Torsional Spring Analogy and Linear-Torsional Spring Analogy. Two-dimensional moving mesh test cases are devised: a rotating rigid airfoil with a prescribed constant rotation velocity and two rigid cylinders moving toward each other with constant velocity. Mesh quality is assessed by a proper parameter and by observation of the null divergent constraint.

The structure of this paper is divided in five sections, including introduction as Section 1. Section 2 presents the mesh motion strategies, along its respective numerical methods. In Section 3, a parameter is introduced to evaluate the mesh quality. The numerical results are presented in Section 4. Finally, Section 5 presents the main conclusions.

2 Mesh motion algorithms

In a context of fluid flow problems with moving boundaries, parts of the domain will move, inducing a geometric modification. Considering two-dimensional triangular finite element meshes, the node position is modified without modifying the element structure. This geometric modification is defined by the prescribed motion of the moving boundary. This motion is mathematically performed by a mapping function χ_t, inside a context of continuum mechanics theory [18]. Therefore, the mathematical problem can be expressed by the following statement:

Find $\mathbf{x}(\mathbf{X},t)$ such that

$$\mathcal{L}(\mathbf{x})=0 \tag{2.1}$$

with boundary conditions prescribed on $\Gamma_{w,t}$ (moving boundary). $\mathbf{x}=\mathbf{x}_w$. $\mathcal{L}(\cdot)$ is a mathematical operator related to the mapping χ_t.

The strategies to solve the motion problem lies on two main approaches:

- Node displacement field; $(\mathbf{x}(t))$;
- Node velocity field; $(\mathbf{w}(t))$;

2.1 Node displacement field

The movement of a elastic pseudo-structural continuum domain is governed by the elastodynamic equations [9]:

$$\rho_{ps}\frac{\partial^2\mathbf{x}}{\partial t^2}-\nabla\cdot\mathbf{T}=f_{ps} \tag{2.2}$$

with boundary condition set as the prescribed displacement at the moving boundary:

$$\mathbf{x}(t)=\mathbf{x}_0(t) \quad \text{on } \Gamma_{w,t}. \tag{2.3}$$

Here, $\mathbf{x}$ is the displacement field, ρ_{ps} is the pseudo-structural density, $\mathbf{T}$ is the Cauchy stress tensor and f_{ps} is the reaction force vector acting on the domain. The displacement field must describe a elastic domain who will endure small deformations in order to satisfy (2.2).

The Cauchy stress tensor relation is given as follows [3,18]:

$$\mathbf{T}=\lambda(\nabla\cdot\mathbf{x})\mathbf{I}+2K\mathbf{E}, \tag{2.4}$$

where K is the diffusivity coefficient, λ is a material constant, $\mathbf{I}$ and $\mathbf{E}$ are the identity and strain tensors respectively. Replacing on (2.2):

$$\rho_{ps}\frac{\partial^2\mathbf{x}}{\partial t^2}-\lambda\nabla(\nabla\cdot\mathbf{x})-2K\nabla\cdot\mathbf{E}=f_{ps}. \tag{2.5}$$

The strain tensor can be written as follows [18, 23]:

$$\mathbf{E}=0.5(\nabla\mathbf{x}+(\nabla\mathbf{x})^T). \tag{2.6}$$

Inserting (2.6) in (2.5):

$$\rho_{ps}\frac{\partial^2\mathbf{x}}{\partial t^2}-(\lambda+K)\nabla(\nabla\cdot\mathbf{x})-K\nabla^2\mathbf{x}=f_{ps}. \tag{2.7}$$

Neglecting reaction forces and considering no volumetric variation of the domain (i.e., $\nabla\cdot\mathbf{x}=0$):

$$\rho_{ps}\frac{\partial^2\mathbf{x}}{\partial t^2}-K\nabla^2\mathbf{x}=0. \tag{2.8}$$

The role of the diffusivity coefficient will be discussed in the next section.

2.2 Node velocity field

The governing equations for a laminar incompressible time-dependent pseudo-flow in a continuum domain are given by:

$$\nabla\cdot\mathbf{w}=\mathbf{0}, \tag{2.9a}$$

$$\frac{\partial\mathbf{w}}{\partial t}+\mathbf{w}\cdot\nabla\mathbf{w}=-\frac{1}{\rho_{pf}}\nabla p+\nu_{pf}\nabla^2\mathbf{w}+f_{pf}. \tag{2.9b}$$

Here, $\mathbf{w}$, e, p are the velocity and pressure fields, ρ_{pf} e ν_{pf} are the density and kinematic viscosity and f_{pf} is the body force term. The boundary conditions are defined at $\Gamma_{pf}=\Gamma_d\bigcup\Gamma_{w,t}\bigcup\Gamma_o$, such as:

$$w(x,t)=w \quad \text{on } \Gamma_{w,t}, \qquad p(x,t)=p_{ref} \quad \text{on } \Gamma_o. \tag{2.10}$$

$\Gamma_{w,t}$ represents the moving boundary. Γ_o is the boundary where a reference pressure p_{ref} is prescribed.

The Stokes problem, also known as creeping flow problem, is defined by predominance of viscous forces over convective forces [1, 12]. As a consequence, the velocity $\mathbf{w}$ and the length scales related to the flow are small. Under these circumstances and admitting absence of body forces, (2.9b) becomes:

$$\frac{\partial\mathbf{w}}{\partial t}-\nu_{pf}\nabla^2\mathbf{w}=-\frac{1}{\rho_{pf}}\nabla p. \tag{2.11}$$

The diffusivity coefficient for displacement and velocity fields (K and ν_{pf}) have the utmost importance for this strategy, since its value yields adequate element deformation [20]. If one sets them as 1, a true Laplacian problem is recovered. The choice the diffusivity coefficient determines the final mesh state. A direct Laplacian smoothing tends to distort small elements, while variable diffusivity coefficients tend to diminish this effect. These coefficients can be based on associating larger mesh velocity gradients to larger elements, or on the distance to the moving boundary. Adequate coefficient values, as well as strategies of variable diffusivity coefficient are well known at the literature for edge-based and element-based algorithms [20].

The diffusivity can be varied based on the distance between the node and the moving boundary. In general, small distances lead to large diffusivity values and large distances result in small diffusivity values [20]. Its model is formulated as a function of the distance l to the nearest moving boundary [15]. The present work employs a variable diffusivity for the node velocity field and an imposed diffusivity for the node displacement field. Node velocity calculation is stabilized with the proper choice of a viscosity value compatible with the time step for the projection method, such as:

$$\nu_{pf}=\frac{h^2}{2\Delta t}, \tag{2.12}$$

where h is the element dimension, given as twice the element area. The advantage of using a variable diffusivity is to avoid local quality degeneration. This diffusivity coefficient will enclose the deformation at the internal part of the mesh.

3 Discretization

3.1 Pseudo structure solution

The node displacement strategy treats the mesh as a dynamic system, with fictitious mass, damping, and stiffness matrices. Its mechanical behavior is, after discretization, governed by [2,6]:

$$\mathcal{L}(\mathbf{x})=\mathbf{M}\ddot{x}+\mathbf{C}\dot{x}+\mathbf{K}x-\mathbf{R}, \tag{3.1}$$

where **M**, **C** and **K** are the mass, damping and stiffness matrices. The vector **R** is the reaction force. For a *quasi*-static model, mass and damping matrices can be neglected, resulting in:

$$\mathcal{L}(\mathbf{x})=\mathbf{K}x-\mathbf{R}. \tag{3.2}$$

The construction of these matrices is made by association with a discrete mechanical system by [9]:

- Lumping the mass matrix at the vertex;
- Lumping the damping matrix between two vertices;
- Adding pseudo-springs between two vertices or at the vertex.

A fictitious linear spring is set on each edge connecting two given vertices (Fig. 1) and a fictitious torsional spring is attached to each vertex. The *quasi*-static formulation is used because the motion equations obey the kinematic compatibility between flow

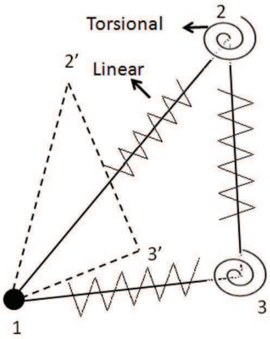


Figure 1: Fictitious Springs at the nodes.

and structure. Therefore, it is suited for flow-structure problems. This compatibility will determine how the mesh position x at the moving boundary will relate with the mesh velocity in any given time step Δt, such as:

$$x_{\Gamma_{w,t}}=\Delta t\ \mathbf{w}_{\Gamma_{w,t}}. \tag{3.3}$$

For one of the test cases of the present work the boundary nodes have a uniform rotation movement of radius $\mathbf{r}$ with a constant angular velocity ω:

$$\mathbf{w}_{\Gamma_{w,t}}=\omega\mathbf{r}. \tag{3.4}$$

The stiffness matrix and the reaction force vector can be divided in parts related to the internal (I) and external (E) mesh degrees of freedom:

$$\begin{bmatrix} \mathbf{K}_{EE} & \mathbf{K}_{EI} \\ \mathbf{K}_{IE} & \mathbf{K}_{II} \end{bmatrix}\begin{bmatrix} \mathbf{x}_E \\ \mathbf{x}_I \end{bmatrix}-\begin{bmatrix} \mathbf{R}_E \\ \mathbf{R}_I \end{bmatrix}=\mathcal{L}(\mathbf{x}). \tag{3.5}$$

Eq. (3.5) is solved with a null force condition at the internal mesh nodes $(\mathbf{R}_E=0)$. Besides, a *quasi*-static model is used accordingly with Eq. (3.2). The stiffness matrix is built using the improved spring analogy method from [8]. Therefore, based on the pseudo-structural methodology, three methods are tested [9]:

- Linear Springs;
- Torsional Springs;
- Linear-Torsional Springs.

Stiffness for the linear and torsional methods is calculated by geometric parameters [9]. For moving boundary problems of small and medium scale, the solution strategy involves direct methods for linear system resolution, due to its simplicity and robustness. Mesh position x^{n+1} is updated by the sum of the mesh displacement vector to the previous configuration x^n, such as [9]:

$$x^{n+1}=x^n+\begin{bmatrix} \mathbf{x}_E \\ \mathbf{x}_I \end{bmatrix}. \tag{3.6}$$

3.2 Pseudo-Stokes projection algorithm

An explicit methodology is employed to solve Eqs. (2.9a) and (2.11). One can consider a set of known variables at time t, given by $(\mathbf{w}^n, p^n, \mathbf{x}^n)$. The solution on time $t+\Delta t$, given by $(\mathbf{w}^{n+1}, p^{n+1}, \mathbf{x}^{n+1})$ is obtained by segregation of velocity and pressure calculations:

$$\frac{1}{\Delta t}(\mathbf{w}^*-\mathbf{w}^n)=-\frac{1}{\rho_{pf}}\nabla p^n+\nu^n_{pf}\nabla^2\mathbf{w}^n, \tag{3.7a}$$

$$\frac{1}{\Delta t}(\mathbf{w}^{n+1}-\mathbf{w}^*)=-\frac{1}{\rho_{pf}}\nabla(p^{n+1}-p^n), \tag{3.7b}$$

$$\nabla\cdot\mathbf{w}^{n+1}=0. \tag{3.7c}$$

The fractional step algorithm introduces a predicted velocity $\mathbf{w}^*$, that will be corrected at the end of each step. One can obtain a Poisson equation for the pressure and its respective boundary condition:

$$\nabla^2(p^{n+1}-p^n)=\frac{\rho_{pf}}{\Delta t}\nabla\cdot\mathbf{w}^*, \tag{3.8a}$$

$$\nabla(p^{n+1}-p^n)\cdot\mathbf{n}=\frac{\rho_{pf}}{\Delta t}\mathbf{w}^*\cdot\mathbf{n} \quad \text{on } \Gamma. \tag{3.8b}$$

The projection method described here is called Incremental Projection Scheme. This method is a modification of the version proposed by [4], which improves convergence as reported in the literature [5, 14]. By considering the finite element space dimension equals to N and defining base functions as:

$$\mathbf{N}_i\colon i=1,\cdots,\mathbf{N}, \qquad \mathbf{N}_j\colon j=1,\cdots,\mathbf{N}. \tag{3.9}$$

The matrix form of the discrete finite element problem is:

1. Predict velocity by Momentum equation:
$$\mathbf{M}\cdot\Delta\mathbf{w}^*=\mathbf{F}^*_w(\mathbf{w}^n,\mathbf{s}^n,p^n); \tag{3.10}$$
2. Pressure-Poisson problem:
$$\mathbf{A}\cdot\Delta p=\mathbf{F}_p(\mathbf{w}^*); \tag{3.11}$$
3. Velocity correction:
$$\mathbf{M}\cdot\Delta\mathbf{w}^{n+1}=\mathbf{F}_w(\Delta p); \tag{3.12}$$
4. Mesh Update:
$$\mathbf{x^{n+1}}=\mathbf{x^n}+\mathbf{w^{n+1}\Delta t}, \tag{3.13}$$

where $\mathbf{s}$ is the weight function, used in a FEM framework. For Eqs. (3.10)-(3.13), the matrices $\mathbf{M}$ and $\mathbf{A}$ are the mass and Laplacian matrices, respectively, and given as follows:

$$\mathbf{M}_{ij}=\frac{1}{\Delta t}(\mathbf{N}_i,\mathbf{N}_j); \qquad \mathbf{A}_{i,j}=(\nabla\mathbf{N}_i,\nabla\mathbf{N}_j). \tag{3.14}$$

The vectors $\mathbf{F}_w^*$, $\mathbf{F}_p$ and $\mathbf{F}_w$ are related to the right hand side discretization through Steps 1-3. The boundary integral terms concerning the boundary conditions are included on these vectors.

An advantage of this numerical scheme resides on the mass matrix. This matrix is lumped in a diagonal form, and it is mounted only when the mesh position is updated. (3.11) is solved by a preconditioned conjugate gradient, where the preconditioner is the Cholesky partial factorization.

4 Mesh quality

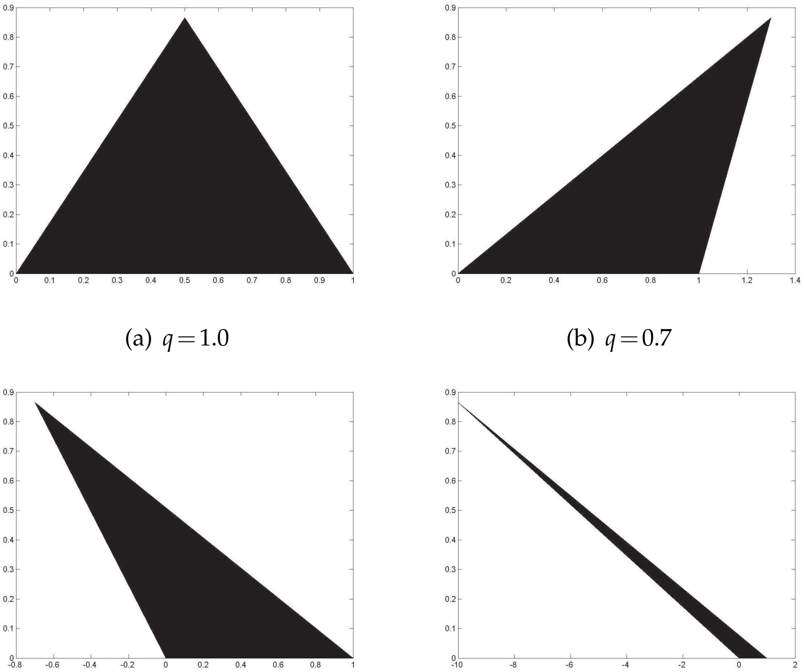
Meshing process is defined as the division of any given domain into smaller elements. This division aims to allow numerical solution of a differential equation at this domain by replacing it into a set of algebraic equations. Mesh generation is an important part of preprocessing procedure. A high quality meshing process is necessary when one performs a simulation with a moving mesh. Therefore, an evaluation method must be implemented to assess the quality of every element. The present paper will use the work of [22], which consists in evaluating the element quality with respect to the equilateral simplex, as the reference of a element that is not distorted or collapsed. For a 2D triangular element, the quality is expressed as:

$$q=f\frac{A}{a^2+b^2+c^2}, \tag{4.1}$$

where A is the area, a, b and c are the lengths of the element sides and $f=4/\sqrt{3}$ is a normalizing factor which levels the quality of an equilateral triangle to 1. This parameter is calculated for every element. It is important to indicate which values of q are adequate to ensure an accurate calculation.

Fig. 2(a) shows an equilateral triangle that has a quality equal to one. On Figs. 2(b) and 2(c), deformed triangles are shown, with quality values equal to 0.7 and 0.5. One can note that the last triangle (Fig. 2(d)) is highly deformed, almost collapsed. Such an exaggerated deformation can lead to inaccurate calculations. Based on that result, an important consideration can be stated: if the quality drops to very small values after deformation, then the flow solution can be compromised by the element shape. However, this quality evaluation method is only employed to show how the element shapes are modified after the mesh motion process. Further conclusions about the flow solutions related to the different levels of mesh quality, here introduced, are not possible, and more investigation is still necessary.

The geometric conservation law represents another way to evaluate mesh quality. As stated before, the computation of geometric quantities associated with a moving mesh must be computed in a way that preserves a uniform flow field. The conservation law starts by application of the integral form of the mass conservation equation on the element. Hence, one can conclude after some mathematical manipulation that the geometric



(a) $q=1.0$

(b) $q=0.7$

(c) $q=0.5$

(d) $q=0.01$

Figure 2: Mesh elements with different quality values.

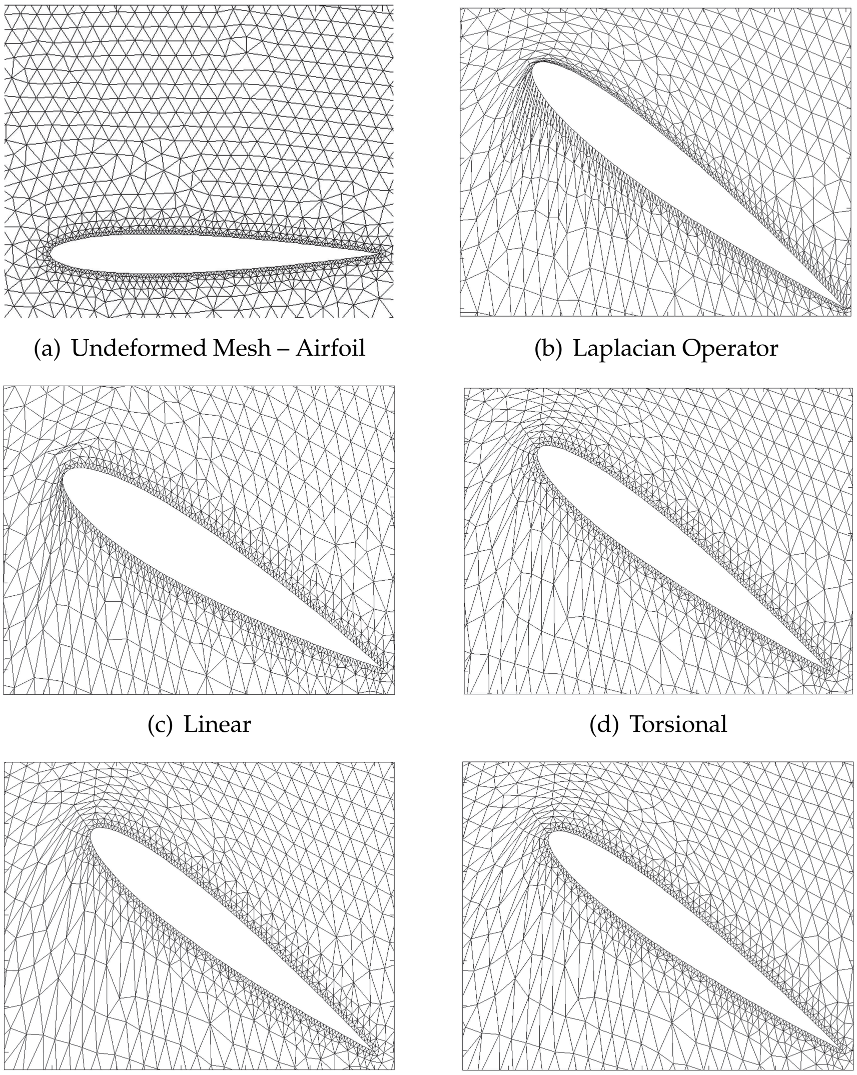
conservation law can be represented by the following condition [21]:

$$\nabla\cdot\mathbf{w}=0. \tag{4.2}$$

The main conclusion of Eq. (4.2) is the maintenance of a mesh velocity null divergent field. Therefore, it represents a necessary condition to recover uniform flow after mesh movement.

5 Results

Fig. 3 shows the deformed meshes of the airfoil for each methodology. Every node on the airfoil was set to move to the final angle of attack of 30 degrees, and the rotation center is on the trailing edge of the airfoil. One can note that some methodologies presented different shapes of deformed elements at the surroundings of the airfoil. The Laplacian and Linear Springs methodologies presented a small number of flattened elements above the airfoil. At the leading edge, the Linear Spring methodology presented elements with invalid triangulation [8]. This result caused the interruption of the simulation. One can note



(a) Undeformed Mesh – Airfoil

(b) Laplacian Operator

(c) Linear

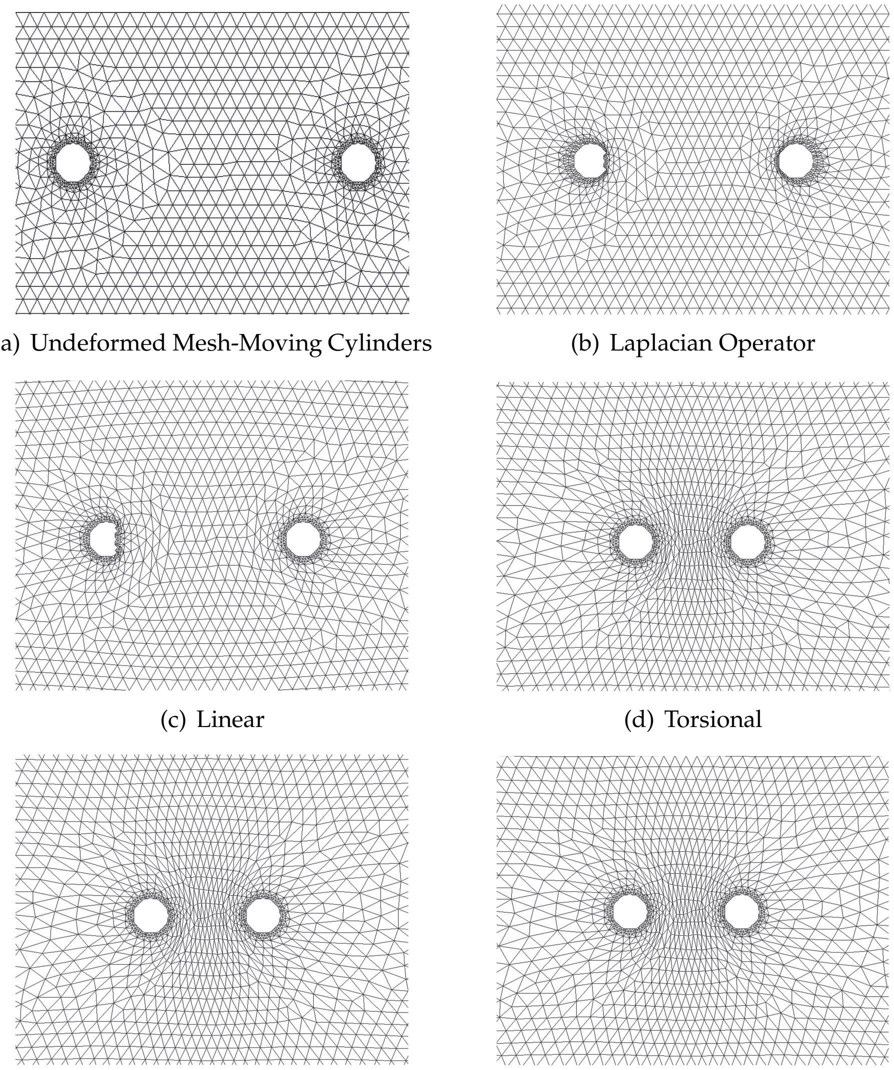
(d) Torsional

(e) Linear-Torsional

(f) Pseudo-Stokes

Figure 3: Deformed meshes – Airfoil.

that wall nodes squeezed the elements at the Torsional and Linear-Torsional methodologies. This is a proof that these motion strategies are not moving the mesh with invalid triangulations. Below the airfoil, elements of the previously mentioned methodologies seem less stretched than the remaining ones. Base nodes of elements below the airfoil remain static, while nodes closer to the lower camber of the airfoil move with it. Torsional, Linear-Torsional and Pseudo-Stokes methodologies show higher number of flattened elements when compared to previous methodologies, with slightly more stretched elements below the airfoil.



(a) Undeformed Mesh-Moving Cylinders

(b) Laplacian Operator

(c) Linear

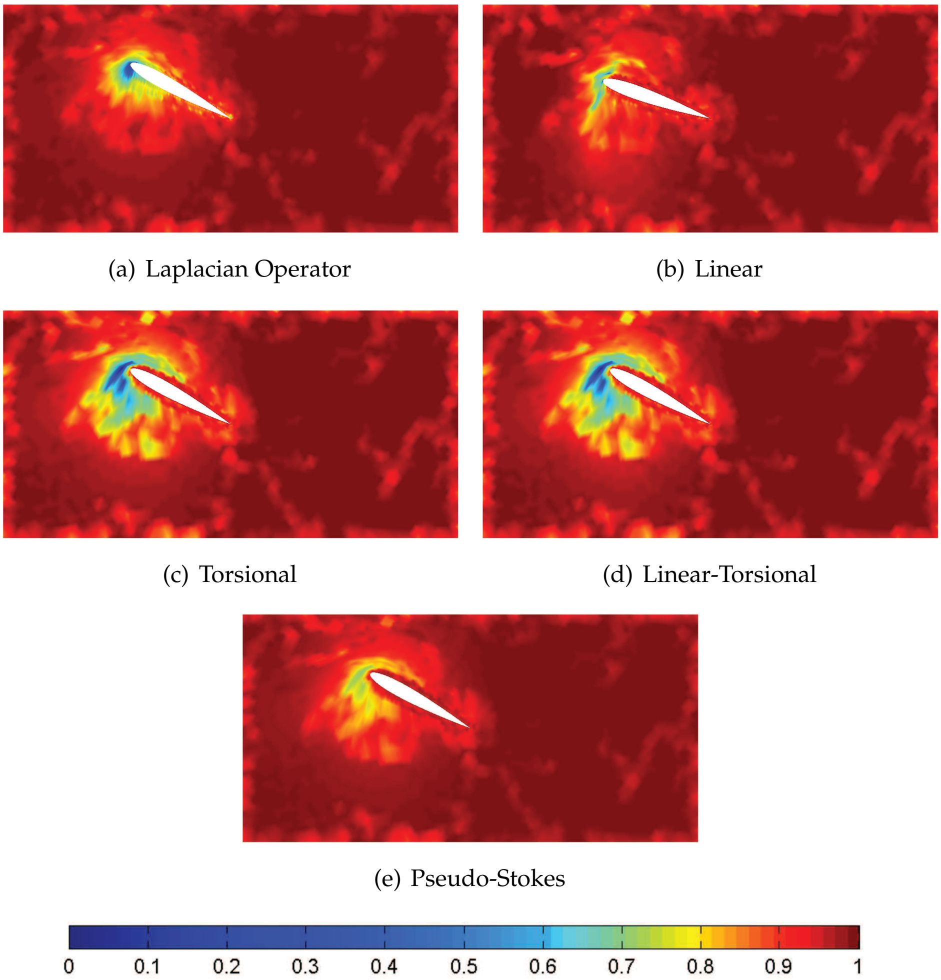
(d) Torsional

(e) Linear-Torsional

(f) Pseudo-Stokes

Figure 4: Deformed meshes – Moving Cylinders.

Fig. 4 shows deformed meshes of cylinders for each methodology. The cylinders were set to move toward each other to the limit before happening negative area elements. Laplacian and Linear Springs have less flattened elements between the cylinders. This result is a consequence of invalid triangulation, causing the simulation to stop again. One can observe the Torsional, Linear-Torsional and Pseudo-Stokes methodologies show less stretched elements behind the cylinders. For these methodologies, there are squeezed elements between the cylinders, but with no sign of invalid triangulation. One can note that cylinders for Torsional, Linear-Torsional and Pseudo-Stokes methodologies were able to



(a) Laplacian Operator

(b) Linear

(c) Torsional

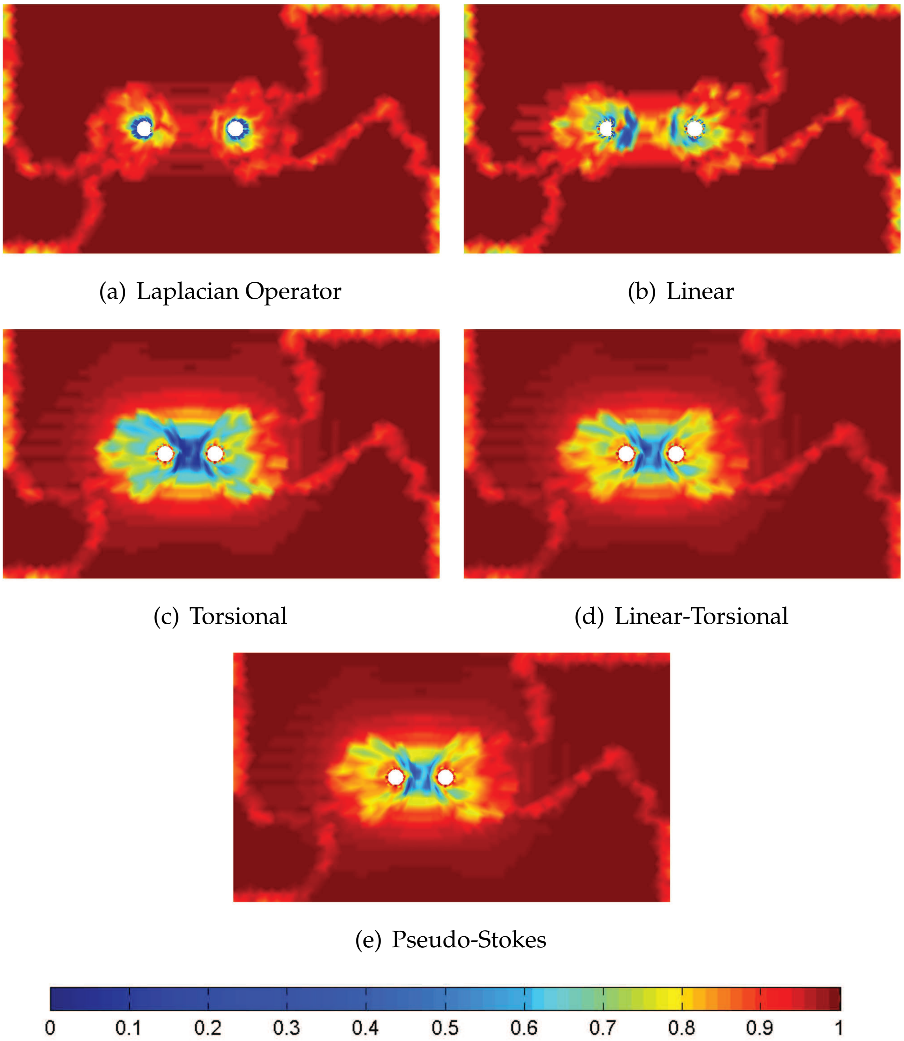
(d) Linear-Torsional

(e) Pseudo-Stokes

Figure 5: Quality Parameter – Airfoil.

come closer. From the computational point of view, one can observe that the Pseudo-Stokes employs extra time to accomplish the full mesh motion test, when compared to the Torsional and the Linear-Torsional methodologies that are also able to fulfill the mesh motion task but with greater number of distorted elements. The extra time spent by the Pseudo Stokes is necessary to ensure the node velocity field with a zero divergent value, and also justified because the numerical scheme used is an explicit one. The performance of the Pseudo Stokes method in time can be improved by using an implicit numerical scheme.

Figs. 5 and 6 show the mesh quality parameter plotted for each methodology applied to the airfoil and the cylinders after movement. Tables 1 and 2 show its minimum values. Laplacian and Linear methods show elements with quality below 0.5 at the trailing



(a) Laplacian Operator

(b) Linear

(c) Torsional

(d) Linear-Torsional

(e) Pseudo-Stokes

Figure 6: Quality Parameter – Moving Cylinders.

edge and between the cylinders. Torsional, Linear-Torsional and Pseudo-Stokes methods presented elements with better quality at the same locations. As seen previously, quality values that will compromise the flow solution are far below 1. Further studies must be executed to determine minimum quality values that ensure good flow solution. Poor quality mesh results computed with the Laplacian and the Linear methods were evident by the invalid triangulation obtained. An important point to be highlighted here is that the mesh quality is also related to the physical problem to be solved. Fluid and structure problems demand different types of meshes. For example, fluid problems, specially for turbulent flows near walls, the mesh quality is very important. Therefore, other mesh quality evaluation methods still need to be considered and tested.

Table 1: Minimum q and maximum mesh velocity divergence – rotating airfoil.

	Min q	Max $\nabla\cdot\mathbf{w}$
Laplacian Operator	2,53E-02	9,08E-06
Linear	2,53E-04	9,76E-06
Torsional	0,25	9,78E-06
Linear Torsional	0,28	7,80E-06
Pseudo-Stokes	0,48	9,80E-06

Table 2: Minimum q and maximum mesh velocity divergence – moving cylinders.

	Min q	Max $\nabla\cdot\mathbf{w}$
Laplacian Operator	5,32E-03	8,00E-06
Linear	1,73E-02	7,08E-06
Torsional	0,2	8,90E-06
Linear Torsional	0,27	9,00E-06
Pseudo-Stokes	0,4	8,00E-06

Tables 1 and 2 also show the maximum mesh velocity divergent calculated for each methodology along the simulation for both cases. One can note that mesh velocity divergent values are very low, never going above 10^{-5}. The null divergent constraint can be understood as a quality parameter showing that these kinds of elements fulfill the geometric conservation law. Torsional, Linear-Torsional and Pseudo-Stokes methodologies show values closer to zero when compared to the remaining ones. The Pseudo-Stokes methodology has as one of its advantage the null velocity field divergent constraint. In order to satisfy this constraint, an adequate value for the mesh viscosity must be computed and a small time step has to be employed to guarantee stability of the explicit projection FEM. Nevertheless, further studies need to be conducted to enhance stability of the projection FEM scheme.

6 Conclusions

In this work a pseudo-Stokes methodology is employed to solve two mesh motion test cases and compared to four other methodologies: Laplacian Operator, Linear Springs, Torsional Springs and Linear-Torsional Springs.

Results for both test cases showed that the majority of methodologies were able to move meshes reasonably. Visual mesh results obtained for these methodologies show that the airfoil has moved to a position of 30 degrees with apparently satisfactory resulting final meshes. It was noted that the Linear methodology provided invalid triangulations for the upper part of the airfoil, and the full rotation movement was not accomplished. For the cylinders, invalid triangulations happened for the Laplacian and the Linear methodologies. The remaining methodologies produced good quality elements.

Despite demanding adequate mesh viscosity the Pseudo-Stokes strategy completed every test case with the best mesh quality indicator. The Pseudo Stokes method satisfy the geometric conservation law because the restriction of node velocity field zero divergent. One can also state that the main advantages of the Pseudo-Stokes methodology are the mesh velocity zero divergent, the use of existing numerical algorithms and the possibility of application to different types of mesh.

Based on these results, the Pseudo-Stokes strategy showed great potential. However, future studies must be done in order to improve the methodology and solve few shortcoming.

References

[1] G. K. BATCHELOR, An Introduction to Fluid Dynamics, Cambridge University Press, 1983.

[2] J. T. BATINA, *Unsteady euler algorithm with unstructured dynamic mesh for complex-aircraft aerodynamic analysis*, AIAA J., 29(3) (1991), pp. 327–333.

[3] P. CHADWICK, Continuum Mechanics-Concise Theory and Problems, 1999.

[4] A. J. CHORIN, *Numerical solution of the Navier-Stokes equations*, Math. Comput., 22 (1968), pp. 745–762.

[5] R. CODINA AND J. BLASCO, *Stabilized finite element method for the transient Navier-Stokes equations based on a pressure gradient projection*, Computer Methods Appl. Mech. Eng., 182(3-4) (2000), pp. 277–300.

[6] C. DEGAND AND C. FARHAT, *A three-dimensional torsional spring analogy method for unstructured dynamic meshes*, Comput. Struct., 80(3-4) (2002), pp. 305–316.

[7] C. FARHAT AND P. GEUZAINE, *Design and analysis of robust ALE time-integrators for the solution of unsteady flow problems on moving grids*, Comput. Methods Appl. Mech. Eng., 193(39-41) (2004), pp. 4073–4095.

[8] C. Farhat, C. Degand, B. Koobus and M. Lesoinne, Improved method of spring analogy for dynamic unstructured fluid meshes, AIAA J., 4 (1998), pp. 3082–3091.

[9] C. FARHAT, C. DEGAND, B. KOOBUS AND M. LESOINNE, *Torsional springs for two-dimensional dynamic unstructured fluid meshes*, Comput. Methods Appl. Mech. Eng., 163(1-4) (1988), pp. 231–245.

[10] C. FARHAT, P. GEUZAINE AND 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(2) (2001), pp. 669–694.

[11] C. FARHAT, K. G. VAN DER ZEE AND P. GEUZAINE, *Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity*, Comput. Methods Appl Mech. Eng., 195(17-18) (2006), pp. 1973–2001.

[12] W. P. GRAEBEL, Advanced Fluid Mechanics, Academic Press, 2007.

[13] J. GUERMOND AND L. QUARTAPELLE, *Calculation of incompressible viscous flows by an unconditionally stable projection FEM*, J. Comput. Phys., 132(1) (1997), pp. 12–33.

[14] J. GUERMOND, P. MINEV AND J. SHEN, *An overview of projection methods for incompressible flows*, Comput. Methods Appl. Mech. Eng., 195(44-47) (2006), pp. 6011–6045.

[15] H. JASAK AND C. TUKOVIC, *Automatic mesh motion for the unstructured finite volume method*, Trans. Famena, 30(2) (2006), pp. 1–20.

[16] A. Johnson and T. E. Tezduyar, *Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces*, Comput. Methods Appl. Mech. Eng., 119(1-2) (1994), pp. 73–94.

[17] R. Kamakoti and W. Shyy, *Evaluation of geometric conservation law using pressure-based fluid solver and moving grid technique*, Int. J. Numer. Methods Heat Fluid Flow, 14(7) (2004), pp. 849–863.

[18] M. Lai, E. Krempl and D. Ruben, Introduction to Continuum Mechanics, 4th edn, 2009.

[19] M. Lesoinne and 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(1-2) (1996), pp. 71–90.

[20] R. Lohner and C. Yang, *Improved ALE mesh velocities for moving bodies*, Commun. Numer. Methods Eng., 12(10) (1996), pp. 599–608.

[21] A. Masud, *Effects of mesh motion on the stability and convergence of ALE based formulations for moving boundary flows*, Comput. Mech., 38(4-5) (2006), pp. 430–439.

[22] D. Rypl, Sequential and Parallel Generation of Unstructured 3d Meshes, PhD thesis, Czech Technical University, 1998.

[23] A. J. M. Spencer, Continuum Mechanics, 1980.

[24] S. Sun, B. Chen, J. Liu and M. Yuan, *An efficient implementation scheme for the moving grid method based on Delaunay graph mapping*, 2nd, International Symposium, American Institute of Physics, 2010.

[25] R. Codina, G. Houzeaux, H. Coppola-Owen and J. Baiges, *The fixed-mesh ALE approach for the numerical approximation of flows in moving domains*, J. Comput. Phys., 228 (2008), pp. 1591–1611.

[26] G. Compere, J. F. Remacle, J. Jansson and J. Hoffman, *A mesh adaptation framework for dealing with large deforming meshes*, Int. J. Numer. Methods Eng., 82 (2010), pp. 843–867.

[27] M. Darbandi and N. Fouladi, *A reduced domain strategy for local mesh movement application in unstructured grids*, Appl. Numer. Math., 61 (2011), pp. 1001–1016.

[28] X. Liu, N. Qin and H. Xia, *Fast dynamic grid deformation based on Delaunay graph mapping*, J. Comput. Phys., 211 (2006), pp. 405–423.

[29] E. Luke, E. Collins and E. Blades, *A fast mesh deformation method using explicit interpolation*, J. Comput. Phys., in press, 2011.

[30] R. P. Dwight, *Robust mesh deformation using the linear elasticity equations*, Computational Fluid Dynamics 2006 Proceedings of the Fourth International Conference on Computational Fluid Dynamics, ICCFD4, Ghent, Belgium.

[31] T. C. S. Rendall and C. B. Allen, *Efficient mesh motion using radial basis functions with data reduction algorithms*, J. Comput. Phys., 228 (2009), pp. 6231–6249.

[32] T. C. S. Rendall and C. B. Allen, *Reduced surface point selection options for efficient mesh deformation using radial basis functions*, J. Comput. Phys., 229 (2010), pp. 2810–2820.

[33] M. J. Baines, M. E.Hubbard and P. K. Jimack, *Velocity-based moving mesh methods for nonlinear partial differential equations*, Commun. Comput. Phys., 10(3) (2011), pp. 509–576.

[34] T. Wick, *Fluid-structure interactions using different mesh motion techniques*, Comput. Struct., 89 (2011), pp. 1456–1467.