INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN BIOMEDICAL ENGINEERING
Int. J. Numer. Meth. Biomed. Engng. 2010; **26**:1435–1448
Published online 19 January 2009 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cnm.1223
COMMUNICATIONS IN NUMERICAL METHODS IN ENGINEERING

Arbitrary Lagrangian Eulerian approximation with remeshing for Navier–Stokes equations

C. M. Murea*,†

Laboratoire de Mathématiques, Informatique et Applications, Université de Haute-Alsace, 4, rue des Frères Lumière, 68093 MULHOUSE Cedex, France

SUMMARY

An algorithm that allows remeshing in the arbitrary Lagrangian Eulerian (ALE) framework is presented. At every time step, we could triangulate the domain using either uniform size meshes or adapted meshes. We analyze the conditions when two time-advancing algorithms based on the backward Euler scheme provide identical approximations. Numerical results are presented for Navier–Stokes equations on moving domain. For three academic tests presented in this paper, the uniform size mesh technique provides more accurate results than the classical ALE method, in particular when the domain is expanding particularly fast. Copyright © 2009 John Wiley & Sons, Ltd.

Received 14 February 2008; Revised 8 December 2008; Accepted 12 December 2008

KEY WORDS: arbitrary Lagrangian Eulerian; finite element; Navier–Stokes

1. INTRODUCTION

A successful method for solving numerically partial differential equations on moving domain is the arbitrary Lagrangian Eulerian (ALE) framework [1, 2]. Most commonly, at each time step, a mesh of the moving domain is obtained as the image by mapping of a fixed mesh of a reference domain.

Different strategies have been proposed to get a mesh of the moving domain by replacing the interior vertices of a fixed mesh in order to accommodate the displacement of the boundary. In [3], the displacement of an interior vertex is computed iteratively making a mean of the displacements of the neighboring vertices. The relocation of an interior vertex is given by making a weighted mean of the displacements of the boundary vertices in [4, p. 90]. A method based on a relaxation technique to solve an unconstrained optimization problem is presented in [5]. In [6], the vertices are moved by solving an elasticity problem. We emphasize that all generated meshes will have the same number of nodes as the initial mesh and the inter-vertex connections stay unchanged. In [7], a method to relocate the nodes of the reference mesh is proposed. In addition, a strategy to optimize the triangulation by changing the inter-vertex connections is studied, but the number of vertices is still constant.

In the case when the domain is expanding particularly fast, the image by mapping of a fixed mesh will be distortioned and consequently the accuracy will be affected. The fluid domain have to be remeshed, either locally or globally. In [8], the norm of the velocity gradient is used to point out the zone that requires refining or corsering. When remeshing, a transfer procedure of the velocity and pressure from the old mesh onto the new one is necessary. In [9], a dynamic programming

*Correspondence to: C. M. Murea, Laboratoire de Mathématiques, Informatique et Applications, Université de Haute-Alsace, 4, rue des Frères Lumière, 68093 MULHOUSE Cedex, France.
†E-mail: cornel.murea@uha.fr

algorithm is proposed in order to reconnect the nodes of the current mesh, without repositioning of the internal nodes. Consequently, the interpolation error is reduced when the solution is transferred from one mesh to other.

The aim of this paper is to introduce an algorithm that permits to retriangulate the computing domain at each time step. We regenerate a mesh for the current time domain independently of the mesh used previously, so the meshes do not have the same number of vertices during the numerical simulation. We can use this feature in order to get meshes of uniform size or to adapt the meshes to the computed solution. Contrary to the so-called conservative scheme [10], we keep the ALE time derivative inside the integral sign. In this paper we present numerical results for Navier–Stokes equations in domains with fast growth. In the case when all the meshes are obtained through the ALE map from a fixed mesh, the error increases when the time step decreases for a particular application. This is an undesirable phenomena. Contrary, when we employ meshes with similar size, when the time step decreases, the error also decreases. In addition, the uniform size mesh technique provides more accurate results for our numerical tests.

2. NAVIER–STOKES EQUATIONS IN ALE FRAMEWORK

Let Ω_t be a two-dimensional domain depending on time parameter $t\in[0,T]$. We assume that the evolution in time of the domain is known. We denote by Σ_t^D a part of the boundary $\partial\Omega_t$ and we set $\Sigma_t^N=\partial\Omega_t\setminus\Sigma_t^D$.

The Navier–Stokes equations in moving domain are: find the velocity $\mathbf{v}(\cdot,t):\Omega_t\to\mathbb{R}^2$ and the pressure $p(\cdot,t):\Omega_t\to\mathbb{R}$, such that

$$\rho\left(\frac{\partial\mathbf{v}}{\partial t}+(\mathbf{v}\cdot\nabla)\mathbf{v}\right)-\nabla\cdot\boldsymbol{\sigma}=\mathbf{f}\quad\forall t\in(0,T)\quad\forall\mathbf{x}\in\Omega_t \tag{1}$$

$$\nabla\cdot\mathbf{v}=0\quad\forall t\in(0,T)\quad\forall\mathbf{x}\in\Omega_t \tag{2}$$

$$\mathbf{v}=\mathbf{g}\quad\forall t\in(0,T)\quad\forall\mathbf{x}\in\Sigma_t^D \tag{3}$$

$$\boldsymbol{\sigma}\mathbf{n}=\mathbf{h}\quad\forall t\in(0,T)\quad\forall\mathbf{x}\in\Sigma_t^N \tag{4}$$

$$\mathbf{v}(\mathbf{x},0)=\mathbf{v}^0(\mathbf{x})\quad\text{in }\Omega_0 \tag{5}$$

where $\rho>0$ is the mass density, $\mu>0$ is the dynamic viscosity, $\mathbf{f}=(f_1,f_2)$ are the applied volume forces, $\mathbf{g}$ is the prescribed boundary velocity, $\mathbf{h}$ is the prescribed boundary stress, $\boldsymbol{\sigma}=-p\mathbb{I}_2+2\mu\boldsymbol{\varepsilon}(\mathbf{v})$ is the stress tensor, $\boldsymbol{\varepsilon}(\mathbf{v})=\frac{1}{2}(\nabla\mathbf{v}+(\nabla\mathbf{v})^{\mathrm{T}})$ is the strain rate tensor, $\mathbf{v}^0$ is the initial fluid velocity.

We denote by $\widehat{\Omega}$ a fixed reference domain. Let $\mathscr{A}_t$, $t\in[0,T]$ be a family of transformations such that $\mathscr{A}_t(\widehat{\Omega})=\Omega_t$, where $\widehat{\mathbf{x}}=(\widehat{x}_1,\widehat{x}_2)\in\widehat{\Omega}$ represents the ALE coordinates and $\mathbf{x}=(x_1,x_2)=\mathscr{A}_t(\widehat{\mathbf{x}})$ the Eulerian coordinates. We assume that both $\mathscr{A}_t$ and its inverse $\mathscr{A}_t^{-1}$ are of class $\mathscr{C}^1$. In addition, we suppose that the derivative $(\partial\mathscr{A}_t/\partial t)(\widehat{\mathbf{x}})$ exists for all $\mathbf{x}\in\widehat{\Omega}$. In the sequel, $\widehat{\mathbf{J}}_{\mathscr{A}_t}$ stands for the jacobian matrix of the ALE map $\widehat{\mathbf{x}}\to\mathscr{A}_t(\widehat{\mathbf{x}})$, defined by

$$\widehat{\mathbf{J}}_{\mathscr{A}_t}(\widehat{\mathbf{x}})=\frac{\partial\mathscr{A}_t(\widehat{\mathbf{x}})}{\partial\widehat{\mathbf{x}}}$$

and $\widehat{J}_{\mathscr{A}_t}(\widehat{\mathbf{x}})=\det(\widehat{\mathbf{J}}_{\mathscr{A}_t}(\widehat{\mathbf{x}}))$ denotes its determinant.

Let $\mathbf{v}$ be the velocity of the fluid in the Eulerian coordinates. The corresponding function in the ALE framework $\widehat{\mathbf{v}}:\widehat{\Omega}\times[0,T]\to\mathbb{R}^2$ is defined by $\widehat{\mathbf{v}}(\widehat{\mathbf{x}},t)=\mathbf{v}(\mathscr{A}_t(\widehat{\mathbf{x}}),t)=\mathbf{v}(\mathbf{x},t)$. We denote the ALE time derivative by

$$\left.\frac{\partial\mathbf{v}}{\partial t}\right|_{\widehat{\mathbf{x}}}(\mathbf{x},t)=\frac{\partial\widehat{\mathbf{v}}}{\partial t}(\widehat{\mathbf{x}},t)$$

DOI: 10.1002/cnm

and the domain velocity by

$$\mathfrak{V}(\mathbf{x},t)=\frac{\partial \mathscr{A}_t}{\partial t}(\widehat{\mathbf{x}})$$

The Navier–Stokes equations in the ALE framework (see [11]) give: find the fluid velocity $\mathbf{v}$ and the fluid pressure p such that

$$\rho\left(\left.\frac{\partial \mathbf{v}}{\partial t}\right|_{\widehat{\mathbf{x}}}+((\mathbf{v}-\mathfrak{V})\cdot\nabla)\,\mathbf{v}\right)-2\mu\nabla\cdot\boldsymbol{\varepsilon}(\mathbf{v})+\nabla p=\mathbf{f}\quad\forall t\in(0,T)\quad\forall\mathbf{x}\in\Omega_t$$

$$\nabla\cdot\mathbf{v}=0\quad\forall t\in(0,T)\quad\forall\mathbf{x}\in\Omega_t$$

The weak form of the Navier–Stokes equations is: find the velocity $\mathbf{v}=\mathbf{g}$ on Σ_t^D and the pressure p such that

$$\int_{\Omega_t}\rho\left.\frac{\partial \mathbf{v}}{\partial t}\right|_{\widehat{\mathbf{x}}}\cdot\mathbf{w}\,\mathrm{d}\mathbf{x}+\int_{\Omega_t}\rho(((\mathbf{v}-\mathfrak{V})\cdot\nabla)\mathbf{v})\cdot\mathbf{w}\,\mathrm{d}\mathbf{x}+a(\mathbf{v},\mathbf{w})+b(\mathbf{w},p)=\ell(\mathbf{w}) \tag{6}$$

$$b(\mathbf{v},q)=0 \tag{7}$$

for all $\mathbf{w}=0$ on Σ_t^D and for all q, where the following notation is used:

$$a(\mathbf{v},\mathbf{w})=\int_{\Omega_t}2\mu\boldsymbol{\varepsilon}(\mathbf{v}):\boldsymbol{\varepsilon}(\mathbf{w})\,\mathrm{d}\mathbf{x}$$

$$b(\mathbf{w},q)=-\int_{\Omega_t}(\nabla\cdot\mathbf{w})q\,\mathrm{d}\mathbf{x}$$

$$\ell(\mathbf{w})=\int_{\Omega_t}\mathbf{f}\cdot\mathbf{w}\,\mathrm{d}\mathbf{x}+\int_{\Sigma_t^N}\mathbf{h}\cdot\mathbf{w}\,\mathrm{d}s$$

Remark 1
Following [12], the system (6)–(7) is called the non-conservative weak formulation.

Proposition 1
If $\partial\mathbf{w}/\partial t|_{\widehat{\mathbf{x}}}=0$, then

$$\int_{\Omega_t}\left.\frac{\partial \mathbf{v}}{\partial t}\right|_{\widehat{\mathbf{x}}}\cdot\mathbf{w}\,\mathrm{d}\mathbf{x}=\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_t}\mathbf{v}\cdot\mathbf{w}\,\mathrm{d}\mathbf{x}-\int_{\Omega_t}\mathbf{v}\cdot\mathbf{w}(\nabla\cdot\mathfrak{V})\,\mathrm{d}\mathbf{x} \tag{8}$$

The proof is based on the Euler expansion formula (see [11, Lemma 7.1, p. 23]), the chain rule and changing the variables formula.

Remark 2
If we replace in (6) the time derivative by the right-hand side of (8), we get an equivalent weak form under the hypotheses $\partial\mathbf{w}/\partial t|_{\widehat{\mathbf{x}}}=0$. In order to obtain the last equality, we can take $\mathbf{w}(\mathscr{A}_t(\widehat{\mathbf{x}}),t)=\widehat{\mathbf{w}}(\widehat{\mathbf{x}})$, where $\widehat{\mathbf{w}}$ does not depend on t.

3. TIME-ADVANCING SCHEME

Let $N\in\mathbb{N}^*$ be the number of time steps and $\Delta t=T/N$ the time step. We set $t_n=n\Delta t$ for $n=0,1,\ldots,N$. We will indicate $\mathbf{v}^{n+1}(\mathbf{x})$, $p^{n+1}(\mathbf{x})$ the approximations of $\mathbf{v}(\mathbf{x},t_{n+1})$, $p(\mathbf{x},t_{n+1})$ for $\mathbf{x}\in\Omega_{t_{n+1}}^F$. We set $\mathbf{x}=\mathscr{A}_{t_{n+1}}(\widehat{\mathbf{x}})$ and therefore $\widehat{\mathbf{x}}=\mathscr{A}_{t_{n+1}}^{-1}(\mathbf{x})$.

DOI: 10.1002/cnm

The time-advancing scheme is: knowing the velocity $\mathbf{v}^n:\Omega_{t_n}\to\mathbb{R}^2$ at the previous time step, the current domain $\Omega_{t_{n+1}}$ and the domain velocity $\mathfrak{V}^{n+1}$, find the velocity $\mathbf{v}^{n+1}:\Omega_{t_{n+1}}\to\mathbb{R}^2$ verifying $\mathbf{v}^{n+1}=\mathbf{g}^{n+1}$ on $\Sigma^D_{t_{n+1}}$ and the pressure $p^{n+1}:\Omega_{t_{n+1}}\to\mathbb{R}$, such that

$$\begin{aligned}&\int_{\Omega_{t_{n+1}}}\rho\left(\frac{\mathbf{v}^{n+1}-\mathbf{V}^n}{\Delta t}\right)\cdot\mathbf{w}\,\mathrm{d}\mathbf{x}+\int_{\Omega_{t_{n+1}}}\rho(((\mathbf{V}^n-\mathfrak{V}^{n+1})\cdot\nabla)\mathbf{v}^{n+1})\cdot\mathbf{w}\,\mathrm{d}\mathbf{x}\\&\quad+\int_{\Omega_{t_{n+1}}}2\mu\,\boldsymbol{\varepsilon}(\mathbf{v}^{n+1}):\boldsymbol{\varepsilon}(\mathbf{w})\,\mathrm{d}\mathbf{x}-\int_{\Omega_{t_{n+1}}}(\nabla\cdot\mathbf{w})\,p^{n+1}\,\mathrm{d}\mathbf{x}\\&=\int_{\Omega_{t_{n+1}}}\mathbf{f}^{n+1}\cdot\mathbf{w}\,\mathrm{d}\mathbf{x}+\int_{\Sigma^N_{t_{n+1}}}\mathbf{h}^{n+1}\cdot\mathbf{w}\,\mathrm{d}s\quad\forall\mathbf{w}=0\quad\text{on }\Sigma^D_{t_{n+1}}\end{aligned}\tag{9}$$

$$-\int_{\Omega_{t_{n+1}}}(\nabla\cdot\mathbf{v}^{n+1})q\,\mathrm{d}\mathbf{x}=0\quad\forall q\tag{10}$$

where $\mathbf{V}^n(\mathbf{x})=\mathbf{v}^n(\mathscr{A}_{t_n}\circ\mathscr{A}^{-1}_{t_{n+1}}(\mathbf{x}))$.

Remark 3
This algorithm is based on the backward Euler scheme in order to approach the time derivative and on a semi-implicit treatment of the convective term. We will see that if we apply the backward Euler scheme to the right-hand side of (8), we will obtain a different algorithm, in general.

In practice, the domain velocity can be approached by the first-order finite difference scheme

$$\mathfrak{V}^{n+1}(\mathbf{x})=\frac{\mathscr{A}_{t_{n+1}}(\widehat{\mathbf{x}})-\mathscr{A}_{t_n}(\widehat{\mathbf{x}})}{\Delta t}=\frac{\mathbf{x}-\mathscr{A}_{t_n}\circ\mathscr{A}^{-1}_{t_{n+1}}(\mathbf{x})}{\Delta t}$$

Proposition 2
Let $\mathbf{v}^{n+1}:\Omega_{t_{n+1}}\to\mathbb{R}^2$, $\mathbf{v}^n:\Omega_{t_n}\to\mathbb{R}^2$ and $\widehat{\mathbf{w}}:\widehat{\Omega}\to\mathbb{R}^2$ be arbitrary fields. We set $\mathbf{w}(\cdot,t_{n+1}):\Omega_{t_{n+1}}\to\mathbb{R}^2$ and $\mathbf{w}(\cdot,t_n):\Omega_{t_n}\to\mathbb{R}^2$ as following:

$$\mathbf{w}(\mathscr{A}_{t_{n+1}}(\widehat{\mathbf{x}}),t_{n+1})=\mathbf{w}(\mathscr{A}_{t_n}(\widehat{\mathbf{x}}),t_n)=\widehat{\mathbf{w}}(\widehat{\mathbf{x}})\tag{11}$$

If $\widehat{J}_{\mathscr{A}_{t_{n+1}}}=\widehat{J}_{\mathscr{A}_{t_n}}$ and $(\nabla\cdot\mathfrak{V}^{n+1})=0$, then

$$\begin{aligned}&\int_{\Omega_{t_{n+1}}}\frac{\mathbf{v}^{n+1}(\mathbf{x})-\mathbf{v}^n(\mathscr{A}_{t_n}\circ\mathscr{A}^{-1}_{t_{n+1}}(\mathbf{x}))}{\Delta t}\cdot\mathbf{w}(\mathbf{x},t_{n+1})\,\mathrm{d}\mathbf{x}\\&=\frac{1}{\Delta t}\int_{\Omega_{t_{n+1}}}\mathbf{v}^{n+1}(\mathbf{x})\cdot\mathbf{w}(\mathbf{x},t_{n+1})\,\mathrm{d}\mathbf{x}-\frac{1}{\Delta t}\int_{\Omega_{t_n}}\mathbf{v}^n(\mathbf{y})\cdot\mathbf{w}(\mathbf{y},t_n)\,\mathrm{d}\mathbf{x}\\&\quad-\int_{\Omega_{t_{n+1}}}\mathbf{v}^{n+1}(\mathbf{x})\cdot\mathbf{w}(\mathbf{x},t_{n+1})(\nabla\cdot\mathfrak{V}^{n+1})(\mathbf{x})\,\mathrm{d}\mathbf{x}\end{aligned}\tag{12}$$

Proof
We denote by E the difference between the left- and right-hand sides of the equality (12). We obtain that

$$\begin{aligned}E=&-\frac{1}{\Delta t}\int_{\Omega_{t_{n+1}}}\mathbf{v}^n(\mathscr{A}_{t_n}\circ\mathscr{A}^{-1}_{t_{n+1}}(\mathbf{x}))\cdot\mathbf{w}(\mathbf{x},t_{n+1})\,\mathrm{d}\mathbf{x}+\frac{1}{\Delta t}\int_{\Omega_{t_n}}\mathbf{v}^n(\mathbf{y})\cdot\mathbf{w}(\mathbf{y},t_n)\,\mathrm{d}\mathbf{x}\\&+\int_{\Omega_{t_{n+1}}}\mathbf{v}^{n+1}(\mathbf{x})\cdot\mathbf{w}(\mathbf{x},t_{n+1})(\nabla\cdot\mathfrak{V}^{n+1})(\mathbf{x})\,\mathrm{d}\mathbf{x}\end{aligned}$$

DOI: 10.1002/cnm

Using the change of variable for the first two integrals, it follows

$$E = -\frac{1}{\Delta t}\int_{\widehat{\Omega}} \mathbf{v}^n(\mathscr{A}_{t_n}(\widehat{\mathbf{x}}))\cdot\mathbf{w}(\mathscr{A}_{t_{n+1}}(\widehat{\mathbf{x}}),t_{n+1})\widehat{J}_{\mathscr{A}_{t_{n+1}}}\,\mathrm{d}\widehat{\mathbf{x}} + \frac{1}{\Delta t}\int_{\widehat{\Omega}} \mathbf{v}^n(\mathscr{A}_{t_n}(\widehat{\mathbf{x}}))\cdot\mathbf{w}(\mathscr{A}_{t_n}(\widehat{\mathbf{x}}),t_n)\widehat{J}_{\mathscr{A}_{t_n}}\,\mathrm{d}\widehat{\mathbf{x}} + \int_{\Omega_{t_{n+1}}} \mathbf{v}^{n+1}(\mathbf{x})\cdot\mathbf{w}(\mathbf{x},t_{n+1})(\nabla\cdot\mathfrak{I}^{n+1})(\mathbf{x})\,\mathrm{d}\mathbf{x}$$

From the relation (11), we get

$$E = \frac{1}{\Delta t}\int_{\widehat{\Omega}} \mathbf{v}^n(\mathscr{A}_{t_n}(\widehat{\mathbf{x}}))\cdot\widehat{\mathbf{w}}(\widehat{\mathbf{x}})(-\widehat{J}_{\mathscr{A}_{t_{n+1}}}+\widehat{J}_{\mathscr{A}_{t_n}})\,\mathrm{d}\widehat{\mathbf{x}} + \int_{\Omega_{t_{n+1}}} \mathbf{v}^{n+1}(\mathbf{x})\cdot\mathbf{w}(\mathbf{x},t_{n+1})(\nabla\cdot\mathfrak{I}^{n+1})(\mathbf{x})\,\mathrm{d}\mathbf{x}$$

We conclude that E vanishes when $\widehat{J}_{\mathscr{A}_{t_{n+1}}} = \widehat{J}_{\mathscr{A}_{t_n}}$ and $(\nabla\cdot\mathfrak{I}^{n+1})=0$. □

Remark 4
Left- and right-hand parts of relation (12) represent backward Euler scheme applied to the left- and right-hand parts of equality (8), respectively. As a consequence, the algorithm presented in this paper and that one obtained discretizing the right-hand part of equality (8) produce two different approximations of the Navier–Stokes equations, in general. But, if the volume of the domain Ω_t is constant in time, then $\widehat{J}_{\mathscr{A}_t}$ is constant also and according to the Euler expansion formula, we obtain that $\nabla\cdot\mathfrak{I}=0$. In this case, both algorithms produce the same approximations. If an incompressible, homogeneous fluid is limited by an impermeable boundary, then the volume of the domain is constant in time.

Remark 5
If either $\widehat{J}_{\mathscr{A}_{t_{n+1}}} \neq \widehat{J}_{\mathscr{A}_{t_n}}$ or $(\nabla\cdot\mathfrak{I}^{n+1})\neq 0$, there exits particular flow such that the equality (12) holds but also there exits flow such that equality (12) is violated. The last case will be illustrated in Section 5.3. A numerical scheme satisfies so-called *Discrete Geometrical Conservation Laws*, if it can reproduce a constant solution. Assuming that the solution is constant in time and space, from (12), it follows that

$$\frac{1}{\Delta t}\left(\int_{\Omega_{t_{n+1}}} \mathbf{w}(\mathbf{x},t_{n+1})\,\mathrm{d}\mathbf{x} - \int_{\Omega_{t_n}} \mathbf{w}(\mathbf{x},t_n)\,\mathrm{d}\mathbf{x}\right) = \int_{\Omega_{t_{n+1}}} \mathbf{w}(\mathbf{x},t_{n+1})(\nabla\cdot\mathfrak{I}^{n+1})(\mathbf{x})\,\mathrm{d}\mathbf{x} \tag{13}$$

But the reciprocal is not truth in general: assuming that (13) holds, we do not get (12) for an arbitrary non-constant solution.

4. FINITE ELEMENT APPROXIMATION WITH REMESHING

In the commonly used ALE framework, the triangulation $\mathscr{T}^{n+1}$ of the computational domain $\Omega_{t_{n+1}}$ is obtained as the image of a fixed mesh $\widehat{\mathscr{T}}$ of the reference domain $\widehat{\Omega}$ by the ALE mapping. But, if the domain is expanding particularly fast, the size of the mesh $\mathscr{T}^{n+1}$ increases drastically and consequently, the approximation looses in accuracy. This inconvenient feature disappears when we employ the algorithm (9)–(10).

First, let us remark that the time derivative approximation appears inside the integral sign in the first term of Equation (9). In the classical approach, the time derivative is out of the integral sign and the shape functions have to verify the constraint $\partial\mathbf{w}/\partial t\big|_{\widehat{\mathbf{x}}}=0$. If all the meshes are obtained from a fixed mesh of the reference domain through the ALE map, this condition holds.

DOI: 10.1002/cnm

Secondly, the algorithm (9)–(10) is defined independently of the finite element triangulation of $\Omega_{t_{n+1}}$. Let $\mathscr{T}^{n+1}$ and $\mathscr{T}^n$ be independent triangulations of $\Omega_{t_{n+1}}$ and Ω_{t_n}, respectively. We emphasize the fact that the two triangulations do not have necessary the same number of vertices or triangles. The problem is how to define the discrete ALE map and how to compute $\mathscr{A}_{t_n}\circ\mathscr{A}_{t_{n+1}}^{-1}(\mathbf{x})$ which appears in the definition of $\mathbf{V}^n(\mathbf{x})=\mathbf{v}^n(\mathscr{A}_{t_n}\circ\mathscr{A}_{t_{n+1}}^{-1}(\mathbf{x}))$ and $\mathfrak{V}^{n+1}(\mathbf{x})=(\mathbf{x}-\mathscr{A}_{t_n}\circ\mathscr{A}_{t_{n+1}}^{-1}(\mathbf{x}))/\Delta t$.

Let $\widehat{\Omega}$ be a reference domain chosen by the user. We denote by $\mathbf{d}_{h_n}^n$ the $\mathbb{P}_1$ finite element approximation of $\mathbf{d}^n:\Omega_{t_n}\to\widehat{\Omega}$ verifying

$$\Delta\mathbf{d}^n=0 \quad \text{in } \Omega_{t_n}$$

$$\mathbf{d}^n=\mathbf{g}^n \quad \text{on } \partial\Omega_{t_n}$$

where $\mathbf{g}^n$ is the displacement that moves $\partial\Omega_{t_n}$ to $\partial\widehat{\Omega}$. The function $\mathbf{d}_{h_n}^n$ is continuous on Ω_{t_n} and linear on each triangle of $\mathscr{T}^n$. In a similar way, we can construct $\mathbf{d}_{h_{n+1}}^{n+1}$ piecewise linear approximation of $\mathbf{d}^{n+1}:\Omega_{t_{n+1}}\to\widehat{\Omega}$.

We denote by A_i the vertices of $\mathscr{T}^n$ and by $\widehat{A}_i$ the image of A_i by the map $\mathbf{d}_{h_n}^n$. Let $\widehat{\mathscr{T}}^n$ be the triangulation of vertices $\widehat{A}_i$ of the domain $\widehat{\Omega}$. Analogous, from $\mathscr{T}^{n+1}$ using $\mathbf{d}_{h_{n+1}}^{n+1}$, we get $\widehat{\mathscr{T}}^{n+1}$ another triangulation of $\widehat{\Omega}$. We emphasize that the two triangulations $\widehat{\mathscr{T}}^n$ and $\widehat{\mathscr{T}}^{n+1}$ of $\widehat{\Omega}$ have not the same number of vertices.

For each vertex $\widehat{A}_i$ of $\widehat{\mathscr{T}}^n$, we define

$$(\mathbf{d}_{h_n}^n)^{-1}(\widehat{A}_i)=\widehat{A}_i-\mathbf{d}_{h_n}^n(A_i)$$

then we can define $(\mathbf{d}_{h_n}^n)^{-1}:\widehat{\mathscr{T}}^n\to\mathscr{T}^n$ by linear interpolation on each triangle of $\widehat{\mathscr{T}}^n$. If $\mathbf{x}\in\Omega_{t_{n+1}}$, then $\mathbf{d}_{h_{n+1}}^{n+1}(\mathbf{x})\in\widehat{\Omega}$ and as a consequence, the expression $(\mathbf{d}_{h_n}^n)^{-1}\circ\mathbf{d}_{h_{n+1}}^{n+1}(\mathbf{x})$ is well defined. Finally, we set

$$\mathscr{A}_{t_n}\circ\mathscr{A}_{t_{n+1}}^{-1}(\mathbf{x})=(\mathbf{d}_{h_n}^n)^{-1}\circ\mathbf{d}_{h_{n+1}}^{n+1}(\mathbf{x})$$

In general, $\mathbf{d}_{h_{n+1}}^{n+1}(\mathbf{x})$ is not a vertex of $\widehat{\mathscr{T}}^n$, so in order to compute its image by $(\mathbf{d}_{h_n}^n)^{-1}$, we need to interpolate on the mesh $\widehat{\mathscr{T}}^n$. In practice, in order to simplify the computation, we can take $\widehat{\Omega}=\Omega_{t_n}$ and we have $\widehat{\mathscr{T}}^n=\mathscr{T}^n$.

In this way, we can triangulate the domain $\Omega_{t_{n+1}}$ independently of the mesh used at the previous time step $\mathscr{T}^n$. Using this feature, we could adopt one of the following strategies.

- *Uniform size meshes.* Some triangulation generator provides mesh with size controlled by the length of the segments on the boundary of the domain. If we want a mesh size of about h, we will approach the boundary of the domain by a polygon of about (length of $\partial\Omega$)$/h$ segments of length close to h.
- *Mixing meshes obtained through the ALE map with uniform size meshes.* If

$$\frac{|\text{length of } \partial\widehat{\Omega}-\text{length of } \partial\Omega_{t_{n+1}}|}{\text{length of } \partial\widehat{\Omega}}\leqslant 0.1 \quad \text{(for example)}$$

  we can use meshes obtained from a fixed mesh of the reference domain $\widehat{\Omega}$ through the ALE map. If not, we use uniform size meshes.
- *Adapted meshes.* A criterion to adapt the mesh to the computed solution is proposed in [13]: generate iteratively a mesh $\mathscr{T}_h$ until

$$\sup_{T\in\mathscr{T}_h,(x,y)\in T}(x-y)^{\mathrm{T}}\mathscr{H}(x-y)\leqslant\text{eps} \tag{14}$$

  where eps is a positive parameter and $\mathscr{H}$ is the Hessian of the computed solution.

DOI: 10.1002/cnm

5. NUMERICAL BENCHMARK

We are interested by fluid–structure interaction problems with application to hemodynamics. When the displacement of the structure is small, such in simulating the blood flow in arteries, the classical ALE method could be successfully employed [14, 15]. However, this method is not appropriate in the case of large displacement of the structure such in simulating the blood flow in the left ventricle of the heart.

A second application that we have in mind is the growth in biological development [16]. For example, the growth of the limb bud is very important during the first seven days, so the classical ALE method fails.

But before simulating real multiphysics applications where the fluid domain grows drastically, it is important to test the accuracy of the ALE method with uniform size mesh on academic cases.

5.1. Test 1. All the meshes are obtained from a fixed mesh of the reference domain through the ALE map

In order to test the time accuracy of the algorithm presented here, we will perform the benchmark proposed in [12, pp. 92–94]. The reference domain is $\widehat{\Omega}=[0,1]\times[0,L]$, where $L=6$ for instant. We denote by $\widehat{\Sigma}^N=[0,1]\times\{L\}$ and $\widehat{\Sigma}^D=\partial\widehat{\Omega}\setminus\widehat{\Sigma}^N$ the boundaries of the reference domain. The ALE map is given by

$$\mathscr{A}_t(\widehat{x}_1,\widehat{x}_2)=\left(\widehat{x}_1,\left(1-0.4\sin\left(\frac{2\pi t}{10}\right)\right)(\widehat{x}_2-0.5)+0.5\right)$$

For the physical parameters $\rho=1$, $\mu=1$, $\mathbf{f}=(f_1,f_2)=(0,0)$, the exact solution is

$$\mathbf{v}(x_1,x_2,t)=\left(-\frac{2V(x_1-L)}{(1+2Vt)},\ \frac{2V(x_2-0.5)}{(1+2Vt)}\right),\quad p(x_1,x_2,t)=-\left(\frac{2V(x_1-L)}{(1+2Vt)}\right)^2$$

where $V=0.2$. On the top, left and bottom boundary, we have imposed the velocity profile, while on the right boundary, we have imposed external forces

$$\mathbf{h}=(h_1,h_2)=\left(-\frac{4V}{(1+2Vt)},0\right)$$

We have performed the simulation in the time interval $[0,T=10]$, for the time steps $\Delta t=\frac{1}{2},\frac{1}{4},\frac{1}{8},\frac{1}{16},\frac{1}{32}$ as in [12], where the time accuracy of a conservative algorithm was tested.

The reference mesh (see Figure 1) is of 696 vertices and 1250 triangles, it is unstructured and its size is $h=0.152882$. The $\mathbb{P}_1+$*bubble* finite element was employed for the approximation of the fluid velocity, while the fluid pressure was approximated by the $\mathbb{P}_1$ finite element. The computations have been produced employing the software FreeFem++ [17]. All the meshes are images of the mesh plotted in Figure 1 through the ALE map. Consequently, all the meshes have the same number of vertices, triangles and the same connections. During the simulation, the mesh size varies from $h_{\min}=0.15$ to $h_{\max}=0.212897$. Video-animations from the numerical simulation results can be found on the web site of the author `http://www.edp.lmia.uha.fr/murea/simulation.html`.

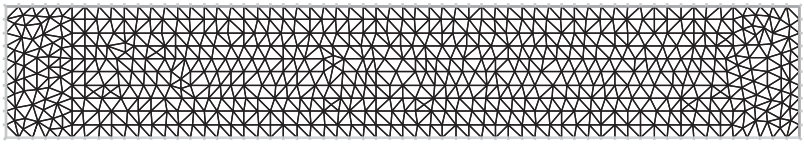

Figure 1. The reference mesh.

DOI: 10.1002/cnm

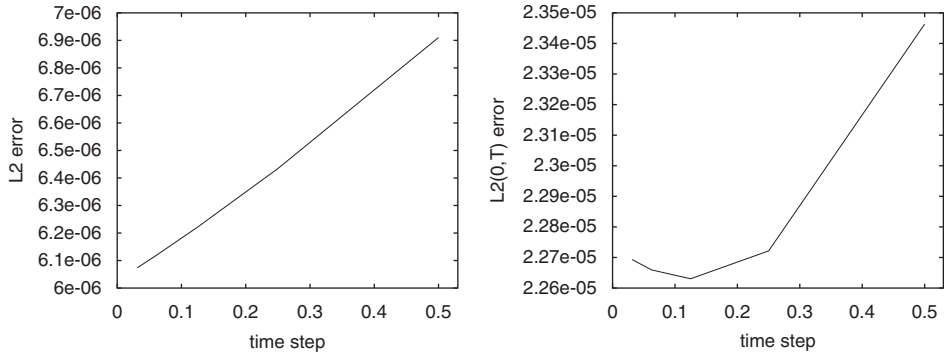

Figure 2. The error in the norm $L^2(\Omega_t)$ between the computed and the exact velocity as function of the time step at the time instant $t=2$ (left). The global error in the norm $L^2(0,T;L^2(\Omega_t))$ as function of the time step (right).

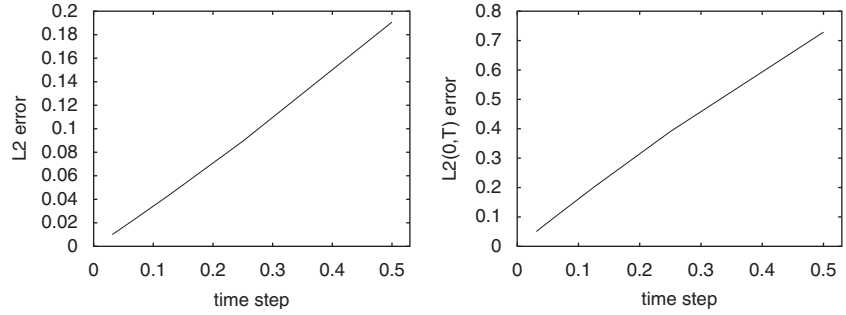

Figure 3. The error in the norm $L^2(\Omega_t)$ between the computed and the exact pressure as function of the time step at the time instant $t=2$ (left). The global error in the norm $L^2(0,T;L^2(\Omega_t))$ as function of the time step (right).

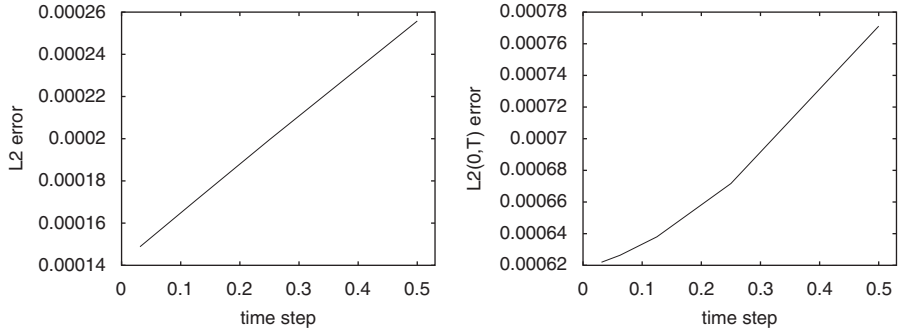

Figure 4. The errors in the norm $L^2(\Omega_t)$ between the computed and the exact forces exerted on the left boundary in function of the time step at the time instant $t=2$ (left). The global errors in the norm $L^2(0,T;L^2(\Omega_t))$ in function of the time step (right).

The norm of $L^2(0,T;L^2(\Omega_t))$ was approximated by $\sqrt{\sum_{n=1}^{N}\Delta t\|p^n\|^2_{L^2(\Omega_{t_n})}}$. Figures 2 and 3 show good agreement between the exact and the computed velocity and pressure. In applications like fluid–structure interaction, it is required to know the forces from the fluid acting on the wall, given by $-\boldsymbol{\sigma}\mathbf{n}$, where $\boldsymbol{\sigma}=-p\mathbb{I}_2+2\mu\boldsymbol{\varepsilon}(\mathbf{v})$. Figure 4 shows that the differences are small between the computed and the exact applied stresses on the left boundary.

DOI: 10.1002/cnm

Supposing that $\widehat{\mathbf{v}}(\widehat{\mathbf{x}},\cdot)$ is of class $\mathscr{C}^2$, we have

$$\frac{v(x,t_{n+1})-v(\mathscr{A}_{t_n}\circ\mathscr{A}_{t_{n+1}}^{-1}(\mathbf{x}),t_n)}{\Delta t}=\frac{\widehat{\mathbf{v}}(\widehat{\mathbf{x}},t_{n+1})-\widehat{\mathbf{v}}(\widehat{\mathbf{x}},t_n)}{\Delta t}=\frac{\partial\widehat{\mathbf{v}}}{\partial t}(\widehat{\mathbf{x}},t_{n+1})+\mathscr{O}(\Delta t)$$

It follows that the algorithm is first-order accurate in time.

Remark 6

In the right plot of Figure 2, we observe that the error in velocity in function of the time step is not a monotone function. We try to explain this phenomenon.

We have seen before that the errors between the exact and the computed solution are small, so most probably our code is bug free. Also, we have seen that the algorithm is consistent.

In the case of a convex and *fixed* domain Ω, for the $\mathbb{P}_1+\text{bubble}/\mathbb{P}_1$ finite elements, we have (see [18], for example)

$$\|\mathbf{v}(n\Delta t)-\mathbf{v}_h^n\|_{L^2(\Omega)}\leqslant\mathscr{O}(\Delta t)+\mathscr{O}(h^2),\quad \|p(n\Delta t)-p_h^n\|_{L^2(\Omega)}\leqslant\mathscr{O}(\Delta t)+\mathscr{O}(h)$$

Consequently, when the time step decreases and the mesh size is constant, the errors diminish. General works on Navier–Stokes approximations are [18–20]. But in our case, the size of the meshes of the computational domain Ω_n is not constant in time. Since the error depends on the time step as well as on the mesh size, if the time step decreases, the error need not necessarily decrease. We will see in the following section that, when the domain *grows* drastically, the space discretization error dominates the time discretization error and the global error increases when the time step decreases.

5.2. Test 2. Comparing the standard ALE mesh generation with the uniform size meshes strategy

In order to prove the importance of using uniform size meshes, we will take a case when the deformation of the domain is more important. We use the same benchmark as in the previous section, but for domain length $L=0.6$ and the time interval $[2.5,7.5]$.

In the first case, all the meshes are obtained through the ALE map from a fixed mesh of 57 vertices and 88 triangles (Figure 5, the mesh at the left). During the simulation, we obtain the mesh size: $h=0.159099$ at $t=2.5$, $h=0.25$ at $t=5$, $h=0.35$ at $t=7.5$ (see Figure 5).

In the second case, we retriangulate at each time step in order to get meshes with similar size h. We obtain at time instant $t=2.5$ a mesh of 57 vertices, 88 triangles and size $h=0.159099$, at $t=5$ a mesh of 84 vertices, 134 triangles and size $h=0.15298$, at $t=7.5$ a mesh of 114 vertices, 186 triangles and size $h=0.159099$ (see Figure 6).

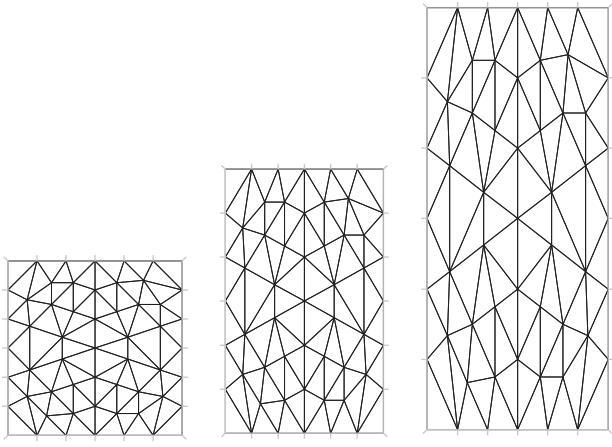

Figure 5. Meshes obtained through the ALE map from a fixed mesh at time instants: $t=2.5$ (left), $t=5$ (middle), $t=7.5$ (right).

DOI: 10.1002/cnm

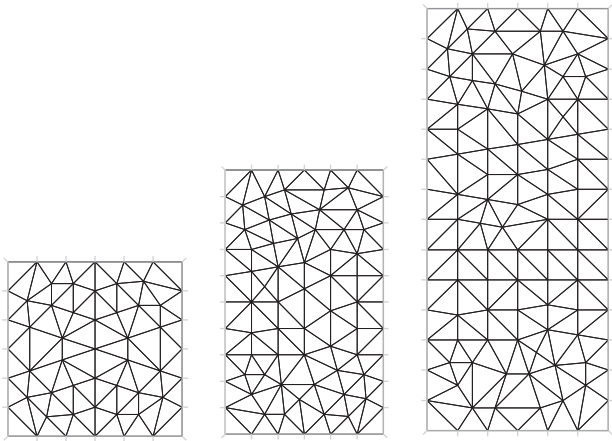

Figure 6. Meshes obtained by retriangulation at time instants: $t=2.5$ (left), $t=5$ (middle), $t=7.5$ (right).

In Figure 7, we plot the error of velocity and pressure in the norm $L^2(0,T;L^2(\Omega_t))$ and the error of the forces exerted on the left boundary in the norm $L^2(0,T;L^2(\Gamma_t^N))$. We recall that the norm of $L^2(0,T;L^2(\Omega_t))$ was approximated by $\sqrt{\sum_{n=1}^{N}\Delta t\|p^n\|^2_{L^2(\Omega_{ln})}}$, therefore it depends on the time step Δt. In the case when all the meshes are obtained through the ALE map from a fixed mesh, the error increases when the time step decreases. On the contrary, when we employ meshes with similar size, when the time step decreases, the error also decreases. We have announced and explained this phenomenon in Remark 6. Another important point to note for our test is the following: the uniform size mesh technique provides more accurate results than the classical ALE framework for small time steps.

5.3. Test 3. Radial flow in a disk with exponential growth

Let Ω_t be the disk of radius $R_0\exp(t)$. Let us consider a flow such that $\nabla\cdot\mathbf{v}=\mathbf{S}$ is a known function, but not necessarily the null function. The Navier–Stokes equations become (see [21, p. 147]): find $\mathbf{v}(\cdot,t):\Omega_t\to\mathbb{R}^2$ and $p(\cdot,t):\Omega_t\to\mathbb{R}$, verifying $\int_{\Omega_t}p(x_1,x_2,t)\,dx_1\,dx_2=0$, such that

$$\rho\left(\frac{\partial\mathbf{v}}{\partial t}+(\mathbf{v}\cdot\nabla)\mathbf{v}\right)-2\mu\nabla\cdot\boldsymbol{\varepsilon}(\mathbf{v})+\nabla p=\mathbf{f}+\frac{2\mu}{3}\nabla\cdot\mathbf{S}\quad\forall t\in(0,T)\quad\forall\mathbf{x}\in\Omega_t$$

$$\nabla\cdot\mathbf{v}=\mathbf{S}\quad\forall t\in(0,T)\quad\forall\mathbf{x}\in\Omega_t$$

$$\mathbf{v}=\mathbf{g}\quad\forall t\in(0,T)\quad\forall\mathbf{x}\in\partial\Omega_t$$

$$\mathbf{v}(\mathbf{x},0)=\mathbf{v}^0(\mathbf{x})\quad\text{in }\Omega_0$$

For $\rho=1$, $\mu=1$, $R_0=1$, we compute $\mathbf{f}$, $\mathbf{S}$ and $\mathbf{g}$, such that the radial flow given by

$$\mathbf{v}(x_1,x_2,t)=(2x_1(x_1^2+x_2^2),2x_2(x_1^2+x_2^2)),\quad p(x_1,x_2,t)=\frac{(x_1^2+x_2^2)}{R_0\exp(t)}-\frac{R_0\exp(t)}{2}$$

is the exact solution of the Navier–Stokes equations.

The reference domain is $\widehat{\Omega}=\Omega_0$ and we consider the ALE map

$$\mathscr{A}_t(\widehat{x}_1,\widehat{x}_2)=\exp(t)(\widehat{x}_1,\widehat{x}_2)$$

We can obtain straightforwardly that

$$\widehat{J}_{\mathscr{A}_t}(\widehat{\mathbf{x}})=\exp(2t),\quad\boldsymbol{\vartheta}(\mathbf{x},t)=\mathbf{x},\quad\nabla\cdot\boldsymbol{\vartheta}(\mathbf{x},t)=2$$

DOI: 10.1002/cnm

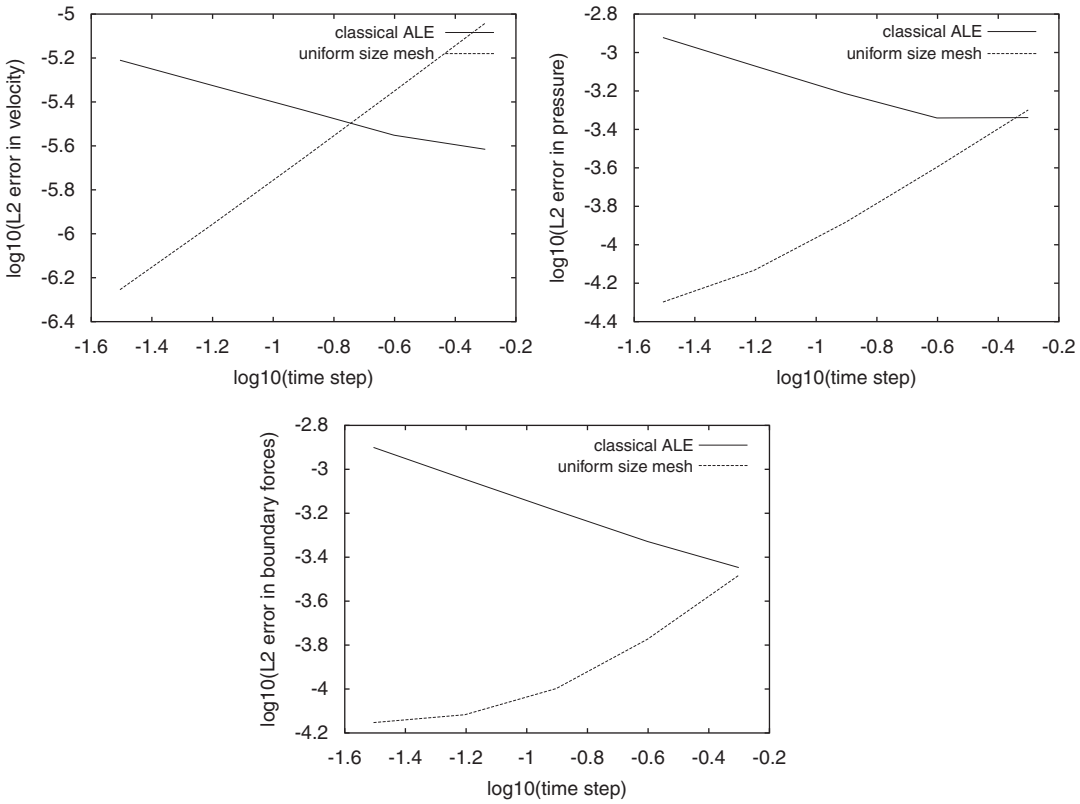

Figure 7. The error of velocity (top left), pressure (top right), forces exerted on the left boundary (bottom) as function of the time step using the $\log_{10}$ scale.

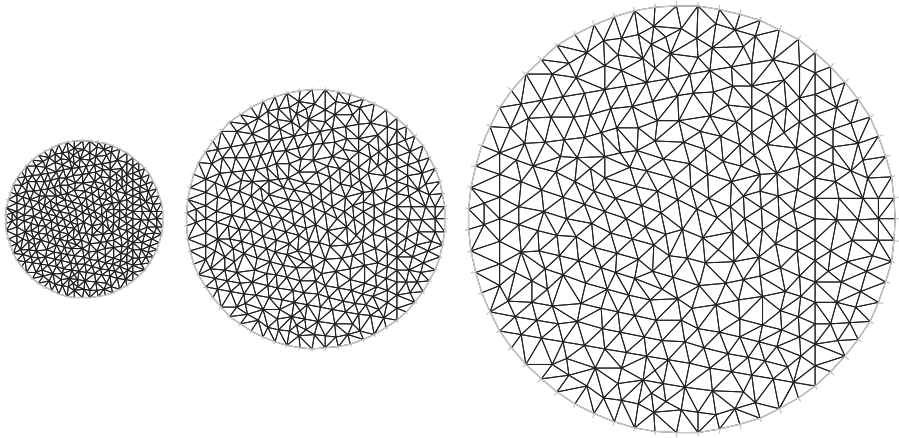

Figure 8. Meshes obtained through the ALE map from a fixed mesh at time instants: $t=0$ (left), $t=0.5$ (middle), $t=1$ (right).

Next, we compare the standard ALE mesh generation with the uniform size meshes strategy for the time step $\Delta t=10^{-2}$ and $t\in[0,1]$.

First, all the meshes are obtained through the ALE map from a fixed mesh of 383 vertices and 701 triangles (Figure 8, the mesh at the left). During the simulation, we obtain the mesh size: $h=0.158012$ at $t=0$, $h=0.260518$ at $t=0.5$, $h=0.429522$ at $t=1$ (see Figure 8).

In the second simulation, we retriangulate at each time step in order to get meshes with similar size h. We obtain at time instant $t=0$ a mesh of 383 vertices, 701 triangles and size $h=0.158012$,

DOI: 10.1002/cnm

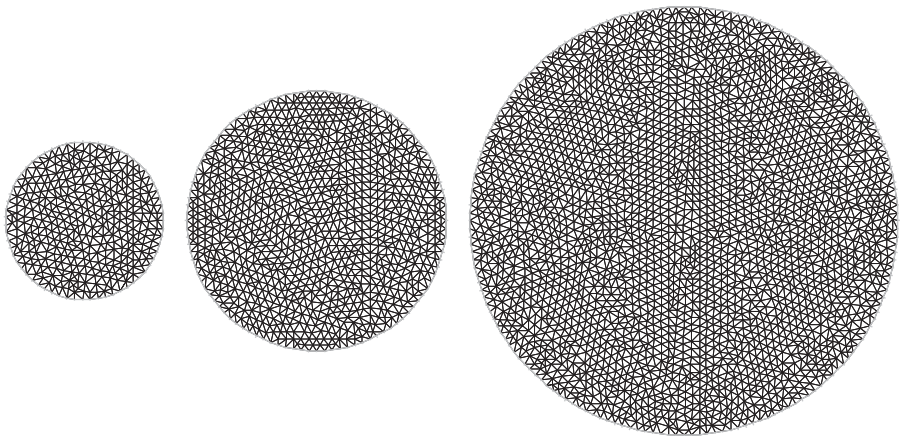

Figure 9. Meshes with uniform size obtained by retriangulation at time instants: $t=0$ (left), $t=0.5$ (middle), $t=1$ (right).

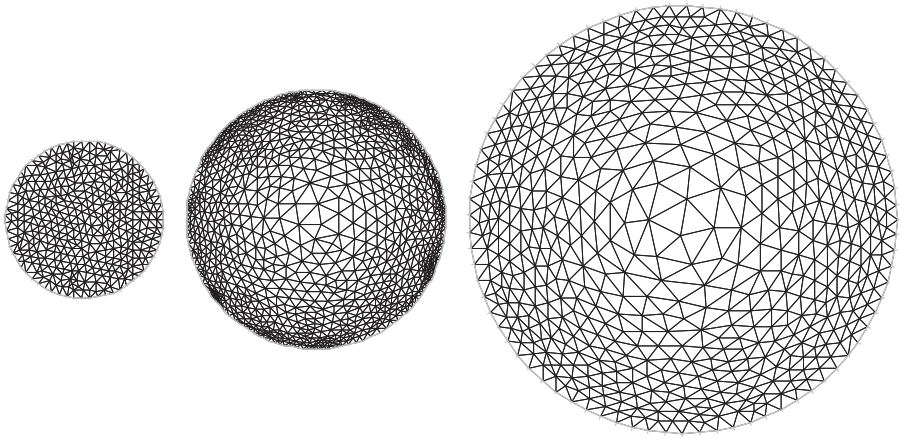

Figure 10. Adapted meshes at time instants: $t=0$ (left), $t=0.5$ (middle), $t=1$ (right).

at $t=0.5$ a mesh of 1006 vertices, 1906 triangles and size $h=0.162088$, at $t=1$ a mesh of 2675 vertices, 5176 triangles and size $h=0.168175$ (see Figure 9).

Remark 7

Contrary to the Tests 1 and 2, the exact solution of the Test 3 is non-linear in space. As done previously, we have used $\mathbb{P}_1+$bubble finite element for the approximation of the fluid velocity, while the fluid pressure was approximated by the $\mathbb{P}_1$ finite element. A drawback of remeshing at each time step is the following: there is error in interpolating the computed solution from a mesh to another. On the other hand, when the meshes are obtained through an ALE map from a fixed mesh, we do not solve accurately because of the very coarse meshes. The error of velocity in the norm $L^2(0,T;\,L^2(\Omega_t))$ is 0.153018 when we map a fixed mesh, while it is only 0.0260057 when we employ meshes with uniform size.

In the third simulation, we use adapted mesh. The software FreeFem++ generates iteratively meshes adapted to the computed solution until (14) holds, where eps $=0.02$. Upto four adaptions are performed at every time step. We obtain at time instant $t=0$ a mesh of 383 vertices, 701 triangles and size $h=0.158012$, at $t=0.5$ a mesh of 1135 vertices, 2121 triangles and size $h=0.344794$, at $t=1$ a mesh of 631 vertices, 1180 triangles and size $h=0.565564$ (see Figure 10).

The objective of mesh adoption is refining or coarsering some zones of the domain in order to capture the behavior of the flow. Generally, the adapted mesh requires fewer number of nodes than

DOI: 10.1002/cnm

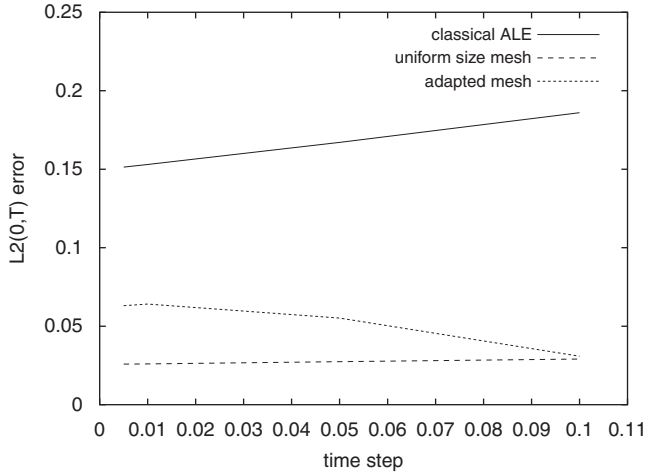

Figure 11. The error of velocity in the norm $L^2(0,T;L^2(\Omega_t))$ as a function of the time step.

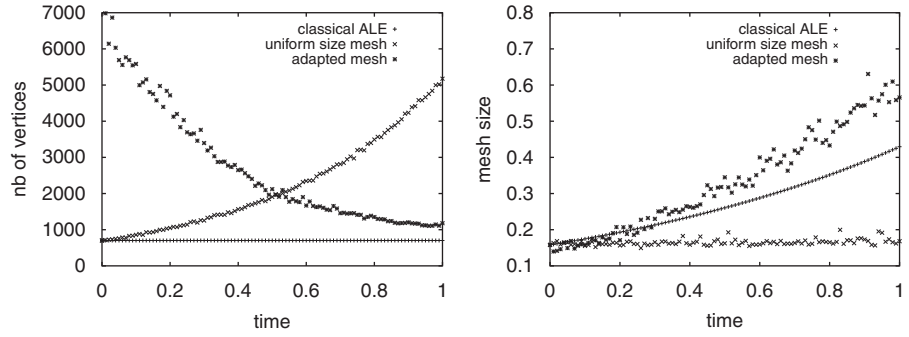

Figure 12. Time histories: the number of vertices (left) and the mesh size (right).

the non-adapted mesh, without loss of accuracy. In this example, for the time step 0.01, the error of velocity in the norm $L^2(0,T;L^2(\Omega_t))$ is 0.0641112, which is less than the error when the meshes are obtained through an ALE map and greater than the error when we employ meshes with uniform size. If we diminish eps in (14) for the adapted mesh strategy, we can obtain a smaller error but, at some time instants, the number of nodes is more important than in the case of uniform size meshes.

In addition, we have performed simulation in the time interval $[0,1]$, for the time steps 0.005, 0.01, 0.05, 0.1, using the three strategies: standard ALE mesh generation, uniform size meshes and adapted meshes, where eps $=0.02$ in (14).

We can see in Figure 11 that, for this particular application, the uniform size mesh technique provides more accurate results compared with two other methods. We note that for the standard ALE mesh generation and uniform size meshes strategy, when the time step decreases, the global error also decreases. It is not the case when adapted meshes are employed. We have explained this phenomenon in Remark 6. We observe in Figure 12 at the right side that the mesh size, when adapted mesh is employed, is greater than the one when the classical ALE mesh generation is used. This situation appears because the adapted mesh is refining near the boundary, but it is coarsering in the center.

6. CONCLUSIONS

We have presented an algorithm that allows to use meshes with uniform size in an ALE framework. If we retriangulate the domain at every time step, an interpolation error is introduced when the

DOI: 10.1002/cnm

solution is transferred from one mesh to another. In spite of this drawback, using uniform size meshes produces more accurate results than the classical ALE method, in particular when the domain is expanding particularly fast.

REFERENCES

1. Hughes HJR, Liu WK, Zimmermann TK. Lagrangian–Eulerian finite element formulation for incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering* 1981; **29**(3):329–349.
2. Donea J. Arbitrary Lagrangian Eulerian finite element methods. In *Computational Methods for Transient Analysis*, Belytschko T, Hughes TJR (eds). Elsevier: Amsterdam, 1983; 473–516.
3. Batina JT. Unsteady Euler airfoil solutions using unstructured dynamic meshes. *American Institute of Aeronautics and Astronautics Journal* 1990; **28**(8):1381–1388.
4. George PL. *Génération Automatique de Maillages*, *Applications aux Méthodes d'Éléments Finis*. Masson: Paris, 1991.
5. Murea CM. Dynamic meshes generation using the relaxation method with applications to fluid–structure interaction problems. *Analele Universitatii din Bucuresti Matematica* 1998; **47**(2):177–186.
6. Johnson AA, Tezduyar TE. Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *Computer Methods in Applied Mechanics and Engineering* 1994; **119**:73–94.
7. Mosler J, Ortiz M. On the numerical implementation of variational arbitrary Lagrangian–Eulerian (VALE) formulations. *International Journal for Numerical Methods in Engineering* 2006; **67**(9):1272–1289.
8. Saksono PH, Dettmer WG, Peric D. An adaptive remeshing strategy for flows with moving boundaries and fluid–structure interaction. *International Journal for Numerical Methods in Engineering* 2007; **71**:1009–1050.
9. Moyle KR, Ventikos Y. Local remeshing for large amplitude grid deformations. *Journal of Computational Physics* 2008; **227**:2781–2793.
10. Formaggia L, Nobile F. A stability analysis for the arbitrary Lagrangian Eulerian formulation with finite elements. *East–West Journal of Numerical Mathematics* 1999; **8**(2):105–131.
11. Quarteroni A, Formaggia L. Mathematical modelling and numerical simulation of the cardiovascular system. In *Handbook of Numerical Analysis*, Ciarlet PG (ed.), vol. XII. Elsevier, North-Holland: Amsterdam, 2004; 3–127.
12. Nobile F. Numerical approximation of fluid–structure interaction problems with application to haemodynamics. *Ph.D. Thesis*, EPFL, Switzerland, 2001.
13. Hecht F. A fews snags in mesh adaptation loops. *The 14th International Meshing Roundtable*, San Diego, 11–14 September 2005.
14. Murea CM. Numerical simulation of a pulsatile flow through a flexible channel. *ESAIM*: *Mathematical Modeling and Numerical Analysis* 2006; **40**:1101–1125.
15. Mbaye I, Murea CM. Numerical procedure with analytic derivative for unsteady fluid–structure interaction. *Communications in Numerical Methods in Engineering* 2008; **24**(11):1257–1275.
16. Murea CM, Hentschel HGE. A finite element method for growth in biological development. *Mathematical Biosciences and Engineering* 2007; **4**(2):339–353.
17. Hecht F, Pironneau O. A finite element software for PDE: FreeFem++. Available from: http://www.freefem.org.
18. Gunzburger MD. *Finite Element Methods for Viscous Incompressible Flows*: *A Guide to Theory*, *Practice*, *and Algorithms*. Academic Press: New York, 1989.
19. Pironneau O. *Finite Element Methods for Fluids*. Wiley, Masson: Chichester, Paris, 1989.
20. Marion M, Temam R. Navier–Stokes equations: theory and approximation. In *Handbook of Numerical Analysis*, Ciarlet PG, Lions JL (eds), vol. VI. Elsevier, North-Holland: Amsterdam, 1998; 96–102.
21. Batchelor GK. *An Introduction to Fluid Dynamics*. Cambridge Mathematical Library. Cambridge University Press: Cambridge, 2000.

DOI: 10.1002/cnm