A fast mesh deformation method for fluid–structure interaction problems

A. Jendoubi

PMU, Mechanical Engineering Department, Prince Mohammad Bin Fahd University, P.O. Box 1664 Al Khobar 31952 KSA

*E-mail: ajendoubi@pmu.edu.sa

Abstract. Previous work by the author [1] has developed a very simple moving mesh method, called parabolic extension, allowing for very large deformation in static problems and to very large time steps for time-dependent problems. In this paper, we significantly accelerate the previous method by using a forward Euler scheme with a lumped mass matrix to solve the parabolic extension and we applied it for simulation in fluid-structure interaction. The use of a lumped mass matrix in the forward scheme leads to an explicit algorithm (no linear system to solve) which make the new technique presented here very fast and less computational cost. Numerical benchmarks including complex boundary with irregular deformation are carried out. The results demonstrate that the new method generates the deformed mesh very fast and with high quality even in large deformations, especially concerning the boundary meshes.

1. Introduction

Fluid-structure interaction (FSI) problems occur in various engineering applications and their numerical simulations is still challenging nowadays. This phenomenon occur when a potentially deformable solid interacts with a surrounding fluid. The flow of the fluid deforms the solid and/or changes its position thus modifying the geometry of the fluid domain. Many difficulties arise and we will briefly present some of them in this work. Arbitrary Lagrangian Eulerian (ALE) formulations (see [2], [3]) are frequently used to solve such problems where the solid and the fluid are respectively in Lagrangian and Eulerian coordinates. The ALE approach is based on an arbitrary motion of the reference frame, which is continuously rezoned in order to allow a precise representation of moving interfaces. Original developments were made, among others, by Noh [4], Franck and Lazarus [5] and Hirt et al. [6]. Early applications are to be found in the work of Donea et al., Belytschko and Kennedy [7] and Hughes et al. [8] for free surface flows and FSI problems.

The computer implementation of the ALE technique requires the introduction of a mesh-update technique that assigns mesh-node velocities or displacements at each time step, or load step of a calculation.

Knowing the node displacements on the solid boundary (obtained by solving the structure equations), various techniques have been proposed to determine how to update the fluid mesh. In a finite element method (or any other technique based on a discrete mesh representation of the domain), the ability to modify the mesh without a complete reconstruction is central to an efficient implementation of the method. The most frequently encountered problem being the excessive deformation of the elements adjacent to the boundary, leading to computational problems and
eventually to the divergence of the solution process. In order word, If care is not taken, elements adjacent to the solid boundary can degenerate leading to computational problems and eventually to the divergence of the solution process.

We consider the general case where the flow of the fluid deforms the solid and changes its position thus modifying the geometry of the fluid domain. We present briefly in this work different moving mesh methods such as kriging, harmonic extension, displacement methods based on an elasticity problem, the parabolic extension (which are presented clearly in [1] and we propose a fast method called the modified parabolic extension to accelerate the step of updating the mesh in the ALE method to solve the FSI problems. The outline of this paper is as follows. As a model for problem with moving boundary, and a typical example of usage of mesh-update techniques, we present, in Section 2, the general context of fluid-structure interactions in ALE formulation. In Section 3, we remind briefly some mesh-update methods presented [1] and we present the new accelerated parabolic extension proposed in this paper. The efficiency of the new parabolic extension algorithm is illustrated with numerical FSI examples in Section 4.

2. The coupled FSI problem: governing equations
In order to introduce the general non-linear fluid-structure problem, let us consider a time-dependent domain \( \Omega(t) \subset \mathbb{R}^d \), \( d = 2 \) or 3. We assume, for all time \( t \), that \( \Omega(t) = \Omega_s(t) \cup \Omega_f(t) \) and \( \Omega_f(t) \cap \Omega_s(t) = \emptyset \), where \( \Omega_f(t) \) is occupied by an incompressible viscous fluid and \( \Omega_s(t) \) by an elastic solid. The reference (initial) configuration of the system is defined \( \Omega_0 = \Omega_s(0) \cup \Omega_f(0) \). The fluid-structure interface at time \( t \) is denoted by \( \sum(t) = \Omega_f(t) \cap \Omega_s(t) \) and \( \sum_0 = \Omega_f(0) \cap \Omega_s(0) \) on the initial configuration.

For the fluid, since we are dealing with a moving interface, we consider the incompressible Navier-Stokes equations in ALE formulation (see [9]):

\[
\begin{aligned}
\rho_f \left( \frac{\partial v_f}{\partial t} + (v_f - v_s) \cdot \nabla v_f \right) + \nabla p - \nabla \cdot (2\eta(v_f)) &= 0 \quad \text{in} \quad \Omega_f(t) \\
\nabla \cdot v_f &= 0 \quad \text{in} \quad \Omega_f(t) \\
v_f(x, t) &= v_s(x, t) \quad \text{on} \quad \sum(t),
\end{aligned}
\]

where \( v_f(x, t) \) is the Eulerian velocity field of the fluid. The vector \( v_s(x, t) \) is the Eulerian domain (mesh) velocity that will be described later on. As we will see, the domain velocity is naturally defined in the solid domain and must somehow be extended to the fluid domain thus the subscript \( s \rightarrow f \). Note that in general \( v_s(x, t) \neq v_f(x, t) \). The continuity of the fluid and solid velocities is imposed on the interface \( \sum(t) \). The system is completed by initial and boundary conditions on the other parts of the boundary \( \Gamma_f^D(t) \) (Dirichlet) and \( \Gamma_f^N(t) \) (Neumann). The complete problem is thus defined (and solved) on the current geometry \( \Omega_f(t) \).

The motion of the structure is described by the following equations:

\[
\begin{aligned}
p_{\rho_0} \dot{U} - \nabla \cdot (\Pi_s(U, s)) &= p_{\rho_0}r_s \quad \text{in} \quad \Omega_s(0) \\
U &= V_s \quad \text{in} \quad \Omega_s(0)
\end{aligned}
\]

(2)

where \( U_s(X, t) \) and \( V_s(X, t) \) are respectively the Lagrangian displacement and velocity fields of the solid, \( \rho_0 \) its density in the undeformed geometry and \( r_s \) a given body force (usually vanishing). The problem is thus written in a total Lagrangian formulation on \( \Omega_s(0) \).

The tensor \( \Pi_s = \Pi_s(U_s) \) is the first Piola-Kirchhoff tensor (see Bonet [10]) and is related to the Cauchy stress tensor \( \sigma_s \) by the relation:

\[
\Pi_s = J_s \sigma_s \cdot F_s^{-T}
\]

(3)

where \( F_s \) is the gradient of deformation tensor and \( J_s \) is the Jacobian of the transformation from the initial geometry \( \Omega_s(0) \) to the current configuration \( \Omega_s(t) \). These quantities come out naturally from the
solution of problem (2). This problem is also completed with a proper set of initial and boundary conditions (Dirichlet on $\Gamma^D_s$ and Neumann on $\Gamma^N_s$).

From a mechanical point of view, the coupling between the systems (1) and (2) is realized by imposing the equilibrium of the stresses at the interface:

$$\sigma_s \cdot n_s = -\sigma_f \cdot n_f \quad \text{on } \sum(1),$$

The Cauchy tensor $\sigma_f$ comes out of the solution of system (1) and is defined on the current configuration. System (2) is however solved on the reference configuration. $\sigma_f$ must therefore be transferred on the reference geometry. It is easily seen from (3) and from the transformation of the normal vectors that the above equilibrium condition on the deforms geometry becomes:

$$\prod_s N_s = -J_{s \rightarrow f}F_{s \rightarrow f}^{-T} \cdot N_f \quad \text{on } \sum_0.$$  \hspace{1cm} (4)

on the reference geometry. In the above equation, $\mathbf{N}$ and $\mathbf{n}$ stand for the normal vectors to the initial and deformed geometries respectively. The equilibrium condition (4) is added to system (2) as a Neumann boundary condition. Note that $J_{s \rightarrow f}$ and $F_{s \rightarrow f}$ are extensions to $\Omega_0$ of similar quantities already defined in $\Omega_0$. Their computation will require the definition of a Lagrangian displacement in $\Omega_0$. In practice, any reasonable extension (denoted $U_s^*$) of $U_s\big|_{\sum_0}$ over $\Omega_0$ can be used and we will see in Section 3 a number of ways to achieve that goal. It is however quite natural to impose the continuity of the displacements at the interface:

$$U_s^* = U_s \quad \text{on } \sum_0.$$  \hspace{1cm} (5)

The Lagrangian mesh velocity $V_{s \rightarrow f}$ in the fluid domain can then be easily obtained from a finite difference in time. For instance, if an Euler scheme is used, then:

$$V_{s \rightarrow f}(X,t) = \frac{U_s^*(X,t+\Delta t) - U_s^*(X,t)}{\Delta t}.$$  \hspace{1cm} (6)

Which, together with (5), imposes the continuity of the velocities on the interface.

We therefore suppose that we have defined an extension $U_s^*$ of the mesh displacement in $\Omega_0$ from which we can evaluate $J_{s \rightarrow f}$ and $F_{s \rightarrow f}$. As a consequence, the current configuration of the fluid domain, $\Omega_f(t)$, is parametrized by:

$$\Omega_f(t) = \Omega_{f0} + U_s^*(\Omega_{f0}).$$  \hspace{1cm} (7)

In other terms, each node $X$ of the initial fluid domain has a corresponding node $x$ in the deformed configuration satisfying:

$$x = X + U_s^*(X,t).$$

In this way, we can follow each node of the fluid mesh. Consequently, the Eulerian domain velocity $v_s^*$ at node $x$ satisfies:

$$v_{s \rightarrow f}(x,t) = V_{s \rightarrow f}(X,t)$$  \hspace{1cm} (8)

and will be used in System (1).

3. Some Mesh-update techniques for FSI problems

We propose in this section different constructions of the extension $U_s^*$. The main difficulty is to avoid the occurrence of overly distorted elements. Once $U_s^*$ has been constructed, quantities such as $J_{s \rightarrow f}$ and $F_{s \rightarrow f}$ are easily computed. The mesh velocity is then obtained from (6) and (8). The update of the mesh is thus a crucial step. In the following, we summarize briefly some methods but more details regarding these different methods are presented in details in [1].
Let us emphasize that we are interested in comparing those methods in the context of time-dependent (or quasi-static) phenomena. In this work, we want to accelerate the parabolic extension presented in [1] and thus not all mesh-update procedures are presented here. For example, the evaluation of the computational performance of Radial Basis Functions (RBF) methods, described in de Boer et al. [11], Gong et al. [12], in Rendall and Allen [13, 14] and in Romani et al. [15], is not straightforward and is outside the scope of this paper. The only methods considered in this paper are Laplacian smoothing, elasticity model and the proposed new method which is accelerated parabolic extension.

3.1. Laplacian smoothing

Laplacian smoothing is a classical method for updating meshes (see [16]). It consists in solving a Laplace (or Poisson) equation for each component of the node position. The following problem is solved separately for each component of the displacement.

\[-\Delta U_s^e = 0 \text{ in } \Omega_0,\]
\[U_s^e = U_s \text{ on } \Sigma_0,\]
\[U_s^e = 0 \text{ on } \partial \Omega_0 \setminus \Sigma_0.\] (9)

The idea is to harmonically expand the evolution of the boundary onto the whole of \(\Omega_0\). This technique has the merit of simplicity and lead to a low computational cost.

3.2. Elasticity model

Another possibility (see [17]) is to solve the small deformation elasticity system:

\[\begin{aligned}
-\nabla \cdot \left[ 2\mu (U_s^e) - \frac{2\mu}{\lambda + \mu} \nabla \cdot U_s^e \right] + \nabla p &= 0 \\
\frac{1}{\lambda + \mu} \nabla \cdot U_s^e &= 0 \text{ on } \Sigma_0 \\
U_s^e &\cdot n = U_s \text{ on } \Sigma_0 \\
U_s^e &\cdot n = 0 \text{ on } \partial \Omega_0 \setminus \Sigma_0
\end{aligned}\] (10)

where \(\lambda\) and \(k\) (bulk modulus) are obtained from the Poisson coefficient \(\nu\) and Young’s modulus \(E\) as

\[k = \frac{E}{3(1 - 2\nu)} \text{ and } \mu = \frac{E}{2(1 + \nu)}\]

This problem cannot be solved component per component. It is thus more expensive that the Laplacian smoothing. The value of the Poisson coefficient \(\nu\) can be chosen close to 0.5 in order to preserve the area (volume) of the elements. The computational cost of such an approach would however be prohibitive as shown in [18]. Notice that in this elasticity approach, we cannot construct the extension component wise. This method is clearly more expensive (from a computational point of view) then the harmonic extension.

3.3. Accelerated parabolic extension

We propose an extension method based on the heat equation using an artificial and arbitrary time.

\[\begin{aligned}
\frac{\partial U_s^e}{\partial t} - \nabla \cdot (k(x) \nabla U_s^e) &= 0 \text{ in } \Omega_0, t \geq 0 \\
U_s^e \cdot n &= U_s \text{ on } \partial \Omega_0 \setminus \Sigma_0 \\
U_s^e &= U_s \text{ on } \Sigma_0
\end{aligned}\] (11)

The basic idea and starting point of this method was to keep the simplicity of the harmonic extension while improving its behavior by avoiding the generation of invalid elements near the moving boundary.
3.3.1. Solving the parabolic extension

The accuracy of the approximate solution of (11) is not essential and a first order time scheme (Euler) is largely sufficient and will lead to a low computational cost. Two methods were explored, a backward Euler scheme ([1]) and a forward Euler scheme with a lumped mass matrix. The first method was clearly studied in [1]. For this paper we retain the last method since the purpose is to accelerate the first method.

This method is extremely efficient: the use of a lumped mass matrix in the forward scheme leads to an explicit algorithm (no linear system to solve). However the conditional stability of this scheme makes it difficult to use. From our experience, only a few very small time steps can be used. A strict observance of the CFL stability condition is not necessary. Monitoring the mesh validity and quality is more relevant.

3.3.2. Artificial time step strategy

Solving (11) over a long period of time (eventually with many time steps) would lead to meshes very close to those obtained with the harmonic extension. It would be indeed a very inefficient way to obtain an harmonic extension.

In the FSI numerical examples presented in the paper, to limit the number of time steps (for computational reason), we need to use a step length sufficient for the diffusion effect to take place while avoiding the steady state. From the definition of the diffusion coefficient, we have

\[ \kappa = 1 = \frac{D^2}{\tau} \]

where \( \tau \) is the time needed for the diffusion to spread over a distance \( D \). In practice, \( D \) can be chosen as the maximal value of the imposed displacement thus giving \( \tau \).

4. Numerical results

4.1. Case #1: 2D Turek et Hron problem

We present a numerical example proposed by Turek et Hron [19]. An elastic bar is attached behind a circular cylinder. The vortex shedding behind the cylinder will provoke oscillations of the flexible structure. The geometry is described as below.

The same dimensions as in [19] were used: \( H = 0.41 \text{m} \), \( L = 2.5 \text{m} \), \( h = 0.02 \text{m} \), and \( l = 0.35 \text{m} \). The cylinder (with radius 0.05m) is centered at (0.2, 0.2). A parabolic velocity profile is imposed at the inlet of the fluid domain as in [19]. The fluid is Newtonian and the solid is modelled using a Saint-Venant-Kirchhoff hyperelastic model:

\[ \prod = F_s (\lambda_s (tr \cdot E))I + 2\mu_s E \]

where \( E = \frac{F'F - I}{2} \) is the Green-Lagrange tensor. The parameters for the problem are:

| Parameters for the solid | Values | Parameters for the fluid | Values |
|--------------------------|--------|--------------------------|--------|
| Density \( \rho_s \) [kg \cdot m^{-3}] | \( 10^4 \) | Density \( \rho_f \) [kg \cdot m^{-3}] | \( 10^3 \) |
| Lamé coefficient \( \lambda_s \) [Pa] | \( 2.0 \times 10^6 \) | Kinematic viscosity \( \nu_f \) [m^2 \cdot s^{-1}] | \( 10^{-3} \) |
| Shear modulus \( \mu_s \) [Pa] | \( 0.5 \times 10^6 \) | Reynold’s number \( Re \) | 100 |
As an illustration of the displacement methods described in section 3, we present the fluid meshes obtained after a rigid rotation of $\pi/8$ of the structure. Figure 1 presents the different meshes near the tip of the structure where the displacement is the largest. As can be easily observed, all meshes contain reversed and unusable elements except for the accelerated parabolic extension method.

Using thus the accelerated parabolic extension method with lumped matrix mass, we have successfully solved Turek’s benchmark problem. We have been able to reproduce accurately the amplitude and frequency of the displacement of the tip of the structure. Figure 2 shows the velocity field and the position of the structure at three time steps. Note that the kriging method can also be used for the update of the fluid domain.

Let us finish this first numerical example with a simple look at the clock time and memory requirements for the harmonic, elastic and accelerated parabolic extensions. This is done in table 1. We have not indicated the computing time for the kriging method since our implementation is not at all optimal and the comparison would be misleading. The parabolic extension produces valid meshes at a very low cost (comparable to the harmonic extension) and appears as one of the best method for all cases.

![Harmonic smoothing.](image1)

![Elasticity equations.](image2)

![Kriging method.](image3)

![Accelerated parabolic smoothing.](image4)

**Figure 1.** Mesh displacement.

![Velocity field and position of the structure.](image5)

**Figure 2.** Velocity field and position of the structure.
Table 1. Case #1: Computing times for one iteration. All computations performed on a typical 3GHz 64 bit Intel processor.

| Methods               | Case #1: Turek et Hron |
|-----------------------|------------------------|
| Harmonic extension    | 16                     |
| Elastic extension     | 86                     |
| Parabolic extension   | 18                     |

| Methods               | Clock time (s) | Memory (KB) |
|-----------------------|----------------|-------------|
| Harmonic extension    | 16             | 1 186 508   |
| Elastic extension     | 86             | 4 157 202   |
| Parabolic extension   | 18             | 1 190 042   |

4.2. Case #2: 3D Tire interaction with Turbulent flow

We present a second 3D numerical example which a tire interacted with a flow. This problem can be seen as a flow of water around a rolling tire for evaluating the aquaplaning of the vehicle. The purpose of the paper here again is to study the aquaplaning phenomena but to show that the accelerated parabolic extension works well even in a very complicated problem. The problem simulation is given in figure 3. Using thus the accelerated parabolic extension method with lumped matrix mass, we have successfully solve the problem. Table 2 presents the clock time and memory requirements for the harmonic, elastic and accelerated parabolic extensions.

![Figure 3. Velocity field and position of tire.](image)

Table 2. Case #2: Computing times for one iteration. All computations performed on a typical 3GHz 64 bit Intel processor.

| Methods               | Case #2: 3D Tire interaction with Turbulent flow |
|-----------------------|-----------------------------------------------|
| Harmonic extension    | Clock time (s) | Memory (KB) |
| Elastic extension     | 786             | 40 157 202   |
| Parabolic extension   | 90              | 10 190 042   |

5. Conclusions

In this work, we have introduced a new mesh-update technique called the accelerated parabolic extension with lumped mass matrix. The new method performs very well in all the tests cases considered. It can sustain very large deformations thus avoiding frequent remeshing of the whole domain for fluidstructure interactions problems. The main advantages of the accelerated parabolic extension are its simplicity, its robustness and its low computational cost.

Acknowledgments
The author would like to thank the Mechanical Engineering Department at Prince Mohammad Bin Fahd University, Al Khobar City, Kingdom of Saudi Arabia for all supports into pursuing this work.

References

[1] A. Jendoubi, J. Deteix, and A. Fortin. A simple mesh-update procedure for fluid–structure interaction problems. Computers & Structures, 169:13 – 23, 2016.

[2] M.A. Fernández and J.-F. Gerbeau. Algorithms for fluid-structure interaction problems. In L. Formaggia, A. Quarteroni, and A. Veneziani, editors, Cardiovascular Mathematics, volume 1 of MS&A, pages 307–346. 2009.

[3] M. Astorino. Interaction fluide-structure dans le systèmes cardiovasculaire. Analyse numérique et simulation. PhD thesis, Département de mathématiques appliquées, Université Pierre et Marie Curie, Paris VI, France, 2010.

[4] W.F. Noh. CEL : A time-dependent two-space dimensional coupled Eulerian-Lagrangian code. In Fernbach S Alder B and Rotenberg M, editors, Methods in Computational Physics, volume 3, pages 117–179, New York, 1964. Academic Press.

[5] R.M. Franck and R.B. Lazarus. Mixed Eulerian-Lagrangian method. In Fernbach S Alder B and Rotenberg M, editors, Fundamental methods in Hydrodynamics, volume 3, pages 47–67, New York, 1964. Academic Press.

[6] C.W. Hirt, A.A. Amsden, and J.L. Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. J. Comput. Phys., 135:203–216, 1997.

[7] T. Belytschko and J.M. Kennedy. Computer methods for subassembly simulation. Nucl. Eng. Des., 49:17–38, 1978.

[8] T.J.R. Hughes, W.K. Liu, and T.K. Zimmermann. Lagrangian-Eulerian finite element formulation for incompressible viscous flows. Comput. Methods Appl. Mech. Eng., 29(3):329–349, 1981.

[9] M.A. Fernández. Modèles simplifiés d’interaction fluide-structure. PhD thesis, University of Paris IX, 2001.

[10] J. Bonet and R. D. Wood. Nonlinear Continuum Mechanics for Finite Element Analysis. Cambridge University Press, Cambridge (UK), New York, 2008.

[11] A. De Boer, M.S. Van der Schoot, and H. Bijl. Mesh deformation based on radial basis function interpolation. Comp. & Struct., 85:784–795, 2007.

[12] D. Gong, J. Chang, and C. Wei. An adaptive method for choosing center sets of RBF interpolation. Journal of Computational, 6(10):2112–2120, 2011.

[13] T.C.S. Rendall and C.B. Allen. Unified fluid–structure interpolation and mesh motion using radial basis functions. International Journal for Numerical Methods in Engineering, 74(10):1519–1559, 2008

[14] T. C. S. Rendall and C. B. Allen. Efficient mesh motion using radial basis functions with data reduction algorithms. Journal of Computational Physics, 228:6231–6249, 2009.

[15] L. Romani, M. Rossini, and D. Schenone. Edge detection methods based on rbf interpolation. Journal of Computational and Applied Mathematics, 2018.

[16] J.U. Brackbill and J.S. Saltzman. Adaptive zoning for singular problems in two dimensions. J. Comput. Phys., 46(3):342–368, 1982.

[17] J.S. Chen, W.K. Liu, and T. Belytschko. Arbitrary Lagrangian–Eulerian methods for materials with memory and friction. In Hughes TJR Tezduyar TE, editor, Recent developments in computational fluid dynamics, volume 95, San Francisco, 1988.

[18] T. Wick. Fluid-structure interactions using different mesh motion techniques. Comput. Struct., 89(13-14):1456–1467, July 2011.

[19] S. Turek and J. Hron. Proposal for numerical benchmarking of fluidstructure interaction between an elastic object and laminar incompressible flow. In H.-J. Bungartz and M. Schäfer, editors, Fluid-Structure Interaction, volume 53 of Lecture Notes in Computational Science and Engineering, pages 371–385. Springer Berlin Heidelberg, 2006.