Journal of Computational Physics 257 (2014) 83–101

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

An immersed boundary method using unstructured anisotropic mesh adaptation combined with level-sets and penalization techniques

R. Abgrall [c,a,b,*], H. Beaugendre [a,b,c], C. Dobrzynski [a,b,c]

[a] *Univ. Bordeaux, IMB, UMR 5251, F-33400 Talence, France*
[b] *CNRS, IMB, UMR 5251, F-33400 Talence, France*
[c] *INRIA, F-33400 Talence, France*

ARTICLE INFO

Article history:
Received 30 May 2012
Received in revised form 2 May 2013
Accepted 19 August 2013
Available online 25 September 2013

Keywords:
Penalization technique
Unstructured mesh
Level-set
Anisotropic mesh
Mesh adaptation
Embedded method
Navier–Stokes equations

ABSTRACT

The interest on embedded boundary methods increases in Computational Fluid Dynamics (CFD) because they simplify the mesh generation problem in the case of the Navier–Stokes equations. The same simplifications occur for the simulation of multi-physics flows, the coupling of fluid–solid interactions in situation of large motions or deformations, to give a few examples. Nevertheless an accurate treatment of the wall boundary conditions remains an issue of the method. In this work, the wall boundary conditions are easily taken into account through a penalization technique, and the accuracy of the method is recovered using mesh adaptation, thanks to the potential of unstructured meshes. Several classical examples are used to demonstrate that claim.

1. Introduction

When dealing with CFD simulations two types of grids are most commonly used: body-fitted grids and embedded grids. In the case of body-fitted grids, the external mesh faces match up with the body surfaces and external boundary faces of the domain. This is different for embedded approach also known as fictitious domain, immersed boundary method (IBM) or Cartesian method. Indeed when considering embedded techniques, the bodies are immersed inside a large mesh, most of the time a Cartesian mesh, and special treatments of the elements close to the body surfaces are performed. When considering general cases of moving or deforming surfaces along with topological changes, both approaches have complementary strengths and weaknesses.

When dealing with moving bodies using body-fitted grids, the partial differential equations (PDEs) describing the flow have to be cast in an Arbitrary Lagrangian–Eulerian frame of reference (ALE), see e.g. [5,37,32,16,17]. The idea is to move the mesh in such a way as to minimize its distortion and if required the mesh can be regenerated and/or adapted and the solution interpolated [31] or not such as in [29] where a tricky space–time formalism is used. Each of these steps (ALE, mesh movement, interpolation) have been optimized but the topology reconstruction may fail for singular surface points and the interpolation required between grids may lead to some loss of information.

On the contrary, embedded grids methods are attractive: the PDE formulation remains in an Eulerian frame of reference even when moving bodies are considered, this formulation simplify *a priori* the meshing issue. Most developments made

* Corresponding author.

0021-9991/$ – see front matter
http://dx.doi.org/10.1016/j.jcp.2013.08.052

using embedded-grids are performed using structured grids [3,36,35,28], though some work using unstructured grids using fixed embedded grids have already been proposed to solve fluid/structure interactions by Wang et al. [42] and Farhat et al. [42], or in Löhner et al. [33] for Computational Structural Dynamics (CSD), with special boundary treatments in order to prevent penetration during contact. In the last decades, different embedded approaches (fictitious domain, IBM, penalization, etc.) have been developed such that flows around complex geometries can be computed [10]. Immersed boundary methods can now deal with incompressible flows [10] or compressible viscous flows [24] and even turbulent flows using mesh stretching and wall law turbulence formulation [30]. However, the drawback of embedded boundary methods remains the complicated treatment of wall boundary conditions in general. Recent developments in embedded boundary methods have focused on that point with algorithms for interface treatment to improve high order accuracy [25,28,34,41] and/or ghost-cell technique [24].

In this work, we introduce the Immersed Boundary Method with Level Sets and Adapted Unstructured Meshes (IBM-LS-AUM) method. The idea is to combine the strength of mesh adaptation, that is to provide an accurate flow description especially when dealing with wall boundary conditions, to the simplicity of embedded grids techniques, that is to simplify the meshing issue and the wall boundary treatment when combined with penalization term to enforce boundary conditions. The bodies are described using the level-set method [40] and are embedded in an unstructured grid. The wall boundary conditions are enforced by a penalization term [3]. Once a first numerical solution is computed, mesh adaptation [20,9] using quality criteria on the level-set and the solution is re-evaluated. In our opinion, the advantage of this strategy is the following: though simple to implement, the IBM techniques suffer a lack of accuracy near the walls. There, the method is consistent but not very accurate. This is why mesh adaptation is employed, in order to remedy to this behavior by a reduced mesh size near the boundaries of interest. Since most of the IMB schemes use Cartesian meshes, this leads to AMR-like grids, see for example [39], where the mesh is no longer conformal. Here we have chosen a different strategy. In order to have simple data structure, we have focused on a numerical method that uses conformal meshes. Our particular choice is not fundamental. In order to adapt the mesh easily, we focus on simplicial meshes using triangles and tets. The localization of the solid bodies is done via a level set method, as in [42], but with mesh adaptation in order to improve both the quality of the surface representation and of the solution. The quality of the adapted mesh is controlled automatically. Of course, the challenge is to control the increase of refined elements, so that the CPU cost of the simulation remains of the same order as the cost of a simulation with a similar mesh, but with a body-fitted geometry. Up to our knowledge, none of the existing embedded method combine these three features together.

This paper is organized as follows: we first describe the system we use, including the penalization terms, and we explain the numerical strategy for one given mesh. We discuss the structure of the penalization. Section 3 describes our anisotropic mesh adaptation strategy and how it is coupled to the system of interest. The IBM-LS-AUM method is tested against several classical problems, the results and the discussion are reported in Section 4. We discuss the choice of the penalization parameter and the cost associated to the method in term of memory requirements. We also discuss qualitatively the quality of the solution, and show that the accuracy of the IBM-LS-AUM solver is similar to the same CFD solver where the boundary conditions are weakly imposed, as in [18]. Some conclusions and perspectives for future work follow.

2. Penalization

2.1. Penalization method for compressible flows

We consider in this work a laminar flow described by the compressible Navier–Stokes equations. The conservative form of the Navier–Stokes equations can be written as follows (Gravity effects are assumed negligible):

$$\begin{aligned}
&\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial \boldsymbol{x}} \cdot (\rho \boldsymbol{u}) = 0 \\
&\frac{\partial}{\partial t}(\rho \boldsymbol{u}) + \frac{\partial}{\partial \boldsymbol{x}} \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) + \frac{\partial p}{\partial \boldsymbol{x}} = \frac{\partial}{\partial \boldsymbol{x}} \cdot \underline{\boldsymbol{\pi}} \\
&\frac{\partial}{\partial t}(\rho e) + \frac{\partial}{\partial \boldsymbol{x}} \cdot \big((\rho e + p)\boldsymbol{u}\big) = \frac{\partial}{\partial \boldsymbol{x}} \cdot (\underline{\boldsymbol{\pi}} \boldsymbol{u} + \boldsymbol{q})
\end{aligned} \tag{1a}$$

where ρ is the density, $\boldsymbol{u}$ the velocity, e the specific total energy. The pressure p and the heat flux $\boldsymbol{q}$ are respectively defined by the perfect gas equation of state and the Fourier law:

$$p = (\gamma - 1)\rho T \quad \text{and} \quad \boldsymbol{q} = -\frac{c_p(\mu)}{Pr}\frac{\partial T}{\partial \boldsymbol{x}} \quad \text{with} \quad \epsilon = e - \frac{1}{2}|\boldsymbol{u}|^2 \text{ and } \epsilon = c_v T.$$

Pr is the laminar Prandtl number, c_p is the heat capacity at constant pressure, c_v is the heat capacity at constant volume (here set to unity), ϵ the specific internal energy, T is the temperature and μ is the laminar viscosity. Here, $Pr = 0.72$. For Newtonian compressible fluids the stress tensor is given by

$$\underline{\boldsymbol{\pi}} = \mu\left(\left[\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}}\right] + \left[\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}}\right]^T - \frac{2}{3}\left[\frac{\partial}{\partial \boldsymbol{x}} \cdot \boldsymbol{u}\right]\underline{\mathbf{Id}}\right).$$