Computers and Structures 122 (2013) 249–258

Contents lists available at SciVerse ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/compstruc

Adaptive mesh refinement techniques for the immersed interface method applied to flow problems

Zhilin Li [a,b,*], Peng Song [c]

[a] Center for Research in Scientific Computation & Department of Mathematics, North Carolina State University, Raleigh, NC 27695, USA
[b] Nanjing Normal University, China
[c] Operations Research Program, North Carolina State University, Raleigh, NC 27695, USA

ARTICLE INFO

Article history:
Received 4 July 2012
Accepted 19 March 2013
Available online 12 April 2013

Keywords:
Adaptive mesh refinement method
Immersed interface method
Stokes and Navier–Stokes equations
Surface tension
Bubble deformation
Level set method

ABSTRACT

In this paper, we develop an adaptive mesh refinement strategy of the Immersed Interface Method for flow problems with a moving interface. The work is built on the AMR method developed for two-dimensional elliptic interface problems in the paper [12] (CiCP, 12(2012), 515–527). The interface is captured by the zero level set of a Lipschitz continuous function $\varphi(x,y,t)$. Our adaptive mesh refinement is built within a small band of $|\varphi(x,y,t)| \leqslant \delta$ with finer Cartesian meshes. The AMR-IIM is validated for Stokes and Navier–Stokes equations with exact solutions, moving interfaces driven by the surface tension, and classical bubble deformation problems. A new simple area preserving strategy is also proposed in this paper for the level set method.

1. Introduction

In this paper, we develop an adaptive mesh refinement (AMR) technique for the immersed interface method (IIM) for elliptic interface problems with discontinuous coefficients across arbitrary interfaces, Stokes and Navier–Stokes equations with a fixed or moving interface.

Advantages using uniform Cartesian grids include simplicity, robustness, and no additional cost in the grid generation for free boundary and moving interface problems. Another consideration is that many fast solvers on uniform grids can be employed. However, uniform meshes may not be efficient or sufficient for some problems that require high resolutions in some parts of the solution domain, particularly in the neighborhood of the interface, which is main interest for many applications. In order to capture the fine structures of the interface, the AMR is natural choice.

Local grid refinement may be effective for interface problems since (1) often we are mainly interested in the solution near or on the interface; (2) the solution away from the interface is often smooth enough and therefore does not require a fine grid to resolve it; (3) often an AMR can provide more accurate gradient computation near the interface which is important for many applications. There are a few adaptive techniques developed for the Immersed Boundary (IB) method using a Lagrangian formulation, see for example [1,3,4,15]; and using the zero level set representation of a Lipschitz continuous function, see for example [13,17,18].

Many AMR techniques use the information from a-prior or a-posterior error estimate based on the solution to determine where and when to employ a local mesh refinement technique without explicitly using the interface information. However, if the location of the interface is known in advance, then using the priori information (known interface) to guide the AMR process may result in more efficient methods.

There is little research on AMR for IIM. In [12], we developed AMR techniques for IB and IIM for elliptic interface problems with a fixed and circular interface. Since then, the AMR-IIM has been extended to problems with general interfaces, which is also reported in this paper. Nevertheless, the focus of this paper is on the AMR-IIM for Stokes and Navier–Stokes equations with fixed and moving interfaces. Since our solver for the Stokes and Navier–Stokes equations is composed of solving several Helmoltz/Poisson equations, for which we can use the AMR-IIM that we have developed for elliptic interface problems. It is certainly non-trivial to extend the AMR-IIM for Poisson equations to Stokes and Navier–Stokes equations with interfaces. In this manuscript, we use the zero level set of a Liptschitz continuous function to represent the moving interface and the level set method to evolve the moving interface.

The rest of paper is organized as follows. In the next section, we review the AMR-IIM for elliptic interface problems and present some examples. In Section 3, we describe the AMR-IIM for Stokes

* Corresponding author.
E-mail addresses: zhilin@math.ncsu.edu (Z. Li), psong@ncsu.edu (P. Song).

0045-7949/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compstruc.2013.03.013

flows with a moving interface and present some examples. In Section 4, we present the AMR-IIM for the Navier–Stokes equations with a moving interface. We conclude and make some acknowledgments in the last section.

2. AMR-IIM for elliptic interface problems

In this section, we review the AMR-IIM developed in [12] for elliptic interface problems of the form

$$\nabla \cdot (\beta \nabla u) = f(\mathbf{x}), \quad \mathbf{x} \in R, \tag{1}$$

$$[u]_\Gamma = w(s), \qquad \left[\frac{\partial u}{\partial n}\right]_\Gamma = v(s), \tag{2}$$

with a prescribed boundary condition of $u(\mathbf{x})$ along ∂R, where R is a rectangular domain, $\Gamma \in C^2$ is a smooth interface that can be parameterized by a one-dimensional variable s, say the arc-length, within the domain R; $w(s) \in C^2$ and $v(s) \in C^1$ are two functions defined along the interface Γ. Note that, when $w(s) = 0$, then the problem can be written as

$$\nabla \cdot (\beta \nabla u) = f(\mathbf{x}) + \int_\Gamma v(\mathbf{X}(s))\delta(\mathbf{x} - \mathbf{X}(s))ds, \quad \mathbf{x} \in R, \tag{3}$$

where $\mathbf{X}(s)$ is a point on the interface Γ. The results of the AMR-IIM in [12] are only for simple geometries such as circles/ellipses and the PDEs with constant coefficients. Now we have generalized the AMR-IIM to more general interfaces and more general elliptic interface problems. Below, we first review how to generate the AMR mesh using a level set function.

2.1. Adaptive mesh generation

We assume that the interface problem is defined on a rectangular domain $\Omega = [a,b] \times [c,d]$. We start with a coarse Cartesian grid, $x_i = a + ih$, $y_j = c + jh$, $i = 0,1,\ldots,m$, $j = 0,1,\ldots,n$. The interface Γ is implicitly represented by the zero level set of a Lipschitz continuous function $\varphi(x,y)$:

$$\Gamma = \left\{(x,y), \varphi(x,y) = 0\right\}. \tag{4}$$

In the discrete case, $\varphi(x,y)$ is defined at grid points as $\varphi(x_i,y_j)$. Often $\varphi(x,y)$ is the signed distance function from Γ.

To generate a finer mesh around interface Γ, we first select parent points within a band of the interface according to

$$|\varphi(x,y)| \leqslant \lambda h, \tag{5}$$

where λ is a control coefficient to adjust the width of refinement band. The grid points $\mathbf{x}_{ij} = (x_i,y_j)$ within the band are selected as parents. We build a refined mesh with a new mesh resolution h/r (r is refinement ratio, $r = 2$ or 4, for example) within the square: $|x - x_i| \leqslant h$ and $|y - y_j| \leqslant h$. Generating the refined square for every parent points yields a refined region around the interface. Its width is flexibly controlled by λ. Fig. 1 shows an example of such a refinement mesh around a circular interface. If a finer mesh is needed, we can select from the second level grid points by:

$$|\varphi(x,y)| \leqslant \lambda h/r, \tag{6}$$

where h/r is the resolution of the second level refined mesh. We can repeat the process to get finer and finer meshes.

We described in [12] how to index grid points from multiple levels and store the related information by an efficient data structure.

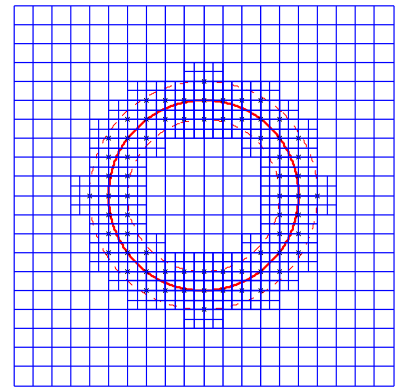

Fig. 1. Adaptive mesh around a circular interface (red solid). Parent grid points (starred) are selected within band $|\varphi_{ij}| = |\varphi(x_i,y_j)| \leqslant h$ (red dashed). Then, finer level square meshes are generated from parent grid points. The refinement ratio $r = 2$, and control coefficient $\lambda = 1$ for refinement width. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

2.2. Finite difference schemes on adaptive meshes

We explain the finite difference (FD) scheme for solving an elliptic interface problem $\beta(u_{xx} + u_{yy}) = f$ at different mesh levels since our solver for Stokes and Navier–Stokes equations is composed of solving several elliptic interface problems (Poisson and Helmholtz equations). For our applications, for example, the Stokes and Navier–Stokes equations with a moving interface, we know the jump conditions $[u] = w$ and $[\beta u_n] = v$ along the interface Γ given as the zero level set of $\varphi(x,y) = 0$.

We list finite difference equations at different types of grid points below, see Fig. 2, for an example, which can be considered as a zoom-in of one particular part in the AMR mesh in Fig. 1.

- Irregular grid points such as the two labeled as 10 and 16. A grid point (x_i,y_j) is called irregular if the interface cuts through the central 5-point stencil centered at (x_i,y_j). Using the level set function, a grid point (x_i,y_j) is irregular if $\varphi_{ij}^{\max}\varphi_{ij}^{\min} \leqslant 0$, where

$$\varphi_{ij}^{\max} = \max\left\{\varphi_{i-1,j}, \varphi_{i+1,j}, \varphi_{ij}, \varphi_{i,j-1}, \varphi_{i,j+1}\right\}, \tag{7}$$

$$\varphi_{ij}^{\min} = \min\left\{\varphi_{i-1,j}, \varphi_{i+1,j}, \varphi_{ij}, \varphi_{i,j-1}, \varphi_{i,j+1}\right\}. \tag{8}$$

Irregular grid points must be in the finest mesh. For irregular grid points, we use the immersed interface method to get the finite dif-

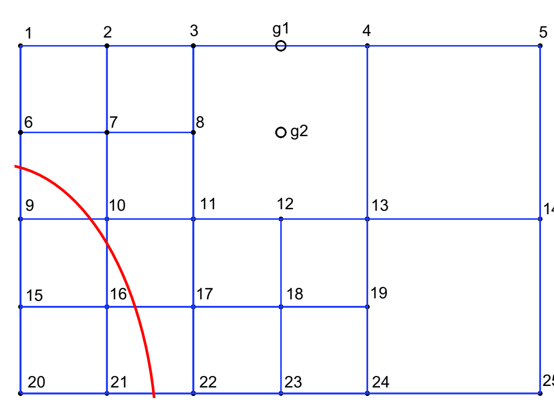

Fig. 2. A part of an adaptive mesh that includes the fine level mesh around an interface (red curve) and the coarse mesh. Hanging nodes 8, 12, 19 lie on border of two mesh levels. Ghost points $g1$ and $g2$ are used to derive the finite difference equation for node 12. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)