Published in final edited form as:

Comput Struct. 2013 June 1; 122: 249–258. doi:10.1016/j.compstruc.2013.03.013.

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

Zhilin Li^{*} and Peng Song[†]

Zhilin Li: zhilin@math.ncsu.edu; Peng Song: psong@ncsu.edu

^{*}Center for Research in Scientific Computation & Department of Mathematics, North Carolina State University, Raleigh, NC 27695, USA.

[†]Operational Research Program, North Carolina State University, Raleigh, NC 27695, USA.

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 $\phi(x, y, t)$. Our adaptive mesh refinement is built within a small band of $|\phi(x, y, t)| \leq \delta$ 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.

Keywords

Adaptive mesh refinement method; immersed interface method; Stokes equations; Navier-Stokes equations; surface tension; bubble deformation; 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

© 2013 Elsevier Ltd. All rights reserved.

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

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]; 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 Helmholtz/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 Lipschitz 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 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 for elliptic interface problems

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

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

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

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

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

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, $x_i = a + ih$, $y_j = c + jh$, $i = 0, 1, \dots, m$, $j = 0, 1, \dots, n$. The interface Γ is implicitly represented by the zero level set of a Lipschitz continuous function $\phi(x, y)$:

$$\Gamma = \{(x, y), \phi(x, y) = 0\}. \quad (4)$$

In the discrete case, $\phi(x, y)$ is defined at grid points as $\phi(x_i, y_j)$. Often $\phi(x, y)$ 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

$$|\phi(x, y)| \leq \lambda h, \quad (5)$$

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: $|x - x_i| \leq h$ and $|y - y_j| \leq h$. Generating the refined square for every parent points yields a refined region around the interface. Its width is flexibly controlled by λ . Figure 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:

$$|\phi(x, y)| \leq \lambda h/r, \quad (6)$$

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.

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 $\phi(x, y) = 0$.

We list finite difference equations at different types of grid points below, see Figure 2, for an example, which can be considered as a zoom-in of one particular part in the AMR mesh in Figure 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 $\phi_{ij}^{\max} \phi_{ij}^{\min} \leq 0$, where

$$\phi_{ij}^{\max} = \max \{\phi_{i-1,j}, \phi_{i+1,j}, \phi_{ij}, \phi_{i,j-1}, \phi_{i,j+1}\}, \quad (7)$$

$$\varphi_{ij}^{\min} = \min \{ \varphi_{i-1,j}, \varphi_{i+1,j}, \varphi_{ij}, \varphi_{i,j-1}, \varphi_{i,j+1} \}. \quad (8)$$

Irregular grid points must be in the finest mesh. For irregular grid points, we use the immersed interface method to get the finite difference equation, see [6, 8, 9] for the derivation. Note that if the coefficient is continuous, that is $[\beta] = 0$, then the IIM uses the standard central finite difference discretization plus a correction term C_{ij} added to the source right hand side,

$$\beta \frac{U_{i-1,j} + U_{i+1,j} + U_{i,j-1} + U_{i,j+1} - 4U_{ij}}{h^2} = f_{ij} + C_{ij}, \quad (9)$$

if β is a constant for simplicity. The correction term C_{ij} depends on the known jump conditions $[u]$, $[\beta u_n]$, $[f]$, and the curvature κ .

- Regular grid points in the fine mesh such as those labeled as 7, 11, 18, at which the grid points in the central 5-point stencil are all from one side of the interface. We use the standard central finite difference scheme for these grid points,

$$\frac{1}{h} \left\{ \left(\beta_{i+1/2,j} \frac{(U_{i+1,j} - U_{ij})}{h} - \beta_{i-1/2,j} \frac{(U_{ij} - U_{i-1,j})}{h} \right) + \left(\beta_{i,j+1/2} \frac{(U_{i,j+1} - U_{ij})}{h} - \beta_{i,j-1/2} \frac{(U_{ij} - U_{i,j-1})}{h} \right) \right\} = f_{ij}. \quad (10)$$

- Hanging nodes such as those labeled as 8, 12, which lie on the border of two different meshes. We can not use the standard central finite difference scheme since the finite difference stencil misses some neighbor grid points in certain directions due to different mesh resolutions. As a common practice, we use an interpolation scheme for hanging nodes by using ghost points such as $g1$ and $g2$. Take the grid point 12 as an example, we use a quadratic interpolation,

$$u_{g1} = \frac{3}{8}u_3 + \frac{3}{4}u_4 - \frac{1}{8}u_5, \quad u_{g2} = -\frac{1}{3}u_{18} + u_{12} + \frac{1}{3}u_{g1}, \quad (11)$$

which has third order accuracy if the values of at the right hand sides are exact. We then plug u_{g2} into the discrete 5-point discrete Laplacian at 12 to get the finite difference equation at the grid point,

$$\beta \frac{\frac{2}{3}u_{18} + u_{11} + u_{13} + \frac{1}{8}u_3 + \frac{1}{4}u_4 - \frac{1}{24}u_5 - 3u_{12}}{h^2/4} = f_{12}. \quad (12)$$

- Regular grid points in the coarse resolutions such as those labeled as 4, 13, 24. We either use the standard central 5-point or 9-point finite difference discretization at these grid points.

After we obtain finite difference equations at all grid points, we use the Algebraic Multigrid (AMG) [16] to solve the system of the finite difference equations. The AMG solver is flexible with the coefficient matrix of the linear system.

2.3 Numerical experiments for elliptic interface problem

We presented some examples with simple geometries (circles and ellipses) in [12]. Since then, we have extended our method to general interfaces, variable coefficients, and arbitrary jump conditions. Below we present several such examples.

Variable but discontinuous coefficient—This example is taken from [6, 9] and has been used in quite a few papers as a benchmark problem. We solve the Poisson equation $(\beta u_x)_x + (\beta u_y)_y = f(x, y)$ defined on the rectangular domain $[-1, 1] \times [-1, 1]$ with a circle interface $x^2 + y^2 = 1/4$. The coefficient β is

$$\beta(x, y) = \begin{cases} x^2 + y^2 + 1 & \text{if } r \leq 1/2, \\ b & \text{if } r > 1/2, \end{cases} \quad (13)$$

where $r = \sqrt{x^2 + y^2}$. The Dirichlet boundary condition is given from the exact solution:

$$u(x, y) = \begin{cases} r^2 & \text{if } r \leq 1/2, \\ \left(1 - \frac{1}{8b} - \frac{1}{b}\right)/4 + \left(\frac{r^4}{2} + r^2\right)/b + C \log(2r)/b & \text{if } r > 1/2. \end{cases} \quad (14)$$

We solve the interface problem on an “AMR $m(+1)[2]$ ” mesh, which means that we add one level of refinement based on a uniform $m \times m$ mesh, with the refinement ratio $r = 2$. In Table 1, we show a comparison of a grid refinement analysis using a uniform mesh and the AMR-IIM approach with parameters $b = 10$ and $C = 0.1$. We present the error $\|E_u\|_\infty$ (the largest error over all grid point); the numbers of unknowns; and the CPU times spent in the multigrid solver. From the table, we can see that we can always improve the accuracy by adding one refinement level AMR $m(+1)[2]$ based on the uniform mesh $m \times m$. For example, the error of the approximated solution is $\|E_u\|_\infty = 6.40e-4$ with a uniform 40 by 40 grid and it reduces to $2.69e-4$ after one level mesh refinement. Moreover, AMR can maintain second order accuracy when we refine both the coarse and refined mesh. With the same finest resolution h , it is not surprising that the AMR has smaller system of equation and less CPU time, but larger errors compared with that obtained from the finest mesh everywhere. We can further improve the error constant if we use the 9-point compact FD scheme on the coarse mesh and a higher interpolation scheme at hanging nodes, see the next example where we use the 9-point compact FD scheme on the coarse mesh.

An example of a complicated interface—We consider the Poisson equation $(\beta u_x)_x + (\beta u_y)_y = f(x, y)$ in the rectangular domain $[-1, 1] \times [-1, 1]$. The interface in polar coordinate is,

$$r(\theta) = 0.5 + 0.1 \sin(8\theta), \quad 0 \leq \theta < 2\pi. \quad (15)$$

The coefficient β is a piecewise constant: $\beta^- = 1$ inside the interface and $\beta^+ = 10$ outside the interface.

The right hand side f , the Dirichlet boundary condition, and the jump conditions $[u]$, $[\beta u_n]$ are all determined from the exact solution,

$$u(x, y) = \begin{cases} \frac{r^2}{\beta^-} & \text{if } (x, y) \in \Omega^- \\ \frac{r^4 - 0.1 \log 2r}{\beta^+} & \text{if } (x, y) \in \Omega^+, \end{cases} \quad (16)$$

where $r = \sqrt{x^2 + y^2}$.

In this example, the interface is complicated and the solution is discontinuous, see Figure 3. We solve the interface problem on an “AMR $m(+1)[2]$ ” mesh, which means that we add one level of refinement based on a uniform $m \times m$ mesh, with the refinement ratio $r = 2$.

In Table 2, we show a grid refinement results using a uniform and an AMR $m(+1)[2]$ mesh by displaying the error $\|E_u\|_\infty$, the numbers of unknowns, and the CPU times spent in the multigrid solver. We observe that the errors from the AMR are almost the same (or smaller) as that obtained from the uniform mesh with same finest resolution which is as expected since the IIM is second order accurate. Nevertheless, the AMR reduces the unknowns by more than 50% and we can focus more on the resolution near the interface. That is the motivation using the AMR approach. Note that, we use the fourth order 9-point compact finite difference scheme at the coarse mesh while the standard 5-point finite difference scheme at the finest mesh. That is why the error of the computed solution using AMR $m(+1)[2]$ may be smaller compared with that obtained from the finest mesh.

3 The AMR-IIM for Stokes equations with a moving interface

In this section, we describe how to use the AMR-IIM to solve the Stokes equations with a moving interface using the similar procedure described in [7].

Consider the 2D Stokes equations for an incompressible flow with a moving interface:

$$\begin{cases} \nabla p = \mu \Delta \mathbf{u} + \mathbf{g} + \int_{\Gamma(s,t)} \mathbf{f}(\mathbf{X}(s, t)) \delta(\mathbf{x} - \mathbf{X}(s, t)) ds, \\ \nabla \cdot \mathbf{u} = 0, \\ \frac{d\Gamma}{dt} = \mathbf{u}(\mathbf{X}(s, t)), \end{cases} \quad (17)$$

in a rectangle domain R that contains the interface Γ with a known boundary condition on R , where $\mathbf{u}$ is the fluid velocity, p is the pressure, μ is the fluid viscosity, $\mathbf{g}$ is an external force, and $\mathbf{f}(\mathbf{X}(s, t))$ is the source strength of the force defined only on the interface. For two dimensional problems, we use the local coordinate system $\xi-\eta$ on the interface in the normal and tangential directions, respectively. The normal direction is denoted as $\mathbf{n} = (\cos\theta \sin\theta)$, where θ is the angle between the normal direction and the x -axis. We use $\hat{f}_1$ and $\hat{f}_2$ to represent the force density on the interface in the normal and tangential directions. Note that, this model has been studies extensively in the literature and have wide range of applications, see for example, [7, 14].

In [7] and [10], the authors used the three Poisson equations approach to solve the Stokes flow with a moving interface. Using the approach, at each time step, the pressure and the velocity are computed by solving three Poisson equations with derived jump conditions in the solution and the normal derivative in terms of the force strength:

$$\begin{cases} p_{xx} + p_{yy} = (g_1)_x + (g_2)_y, \\ [p] = \hat{f}_1, \\ [p_n] = \frac{\partial \hat{f}_1}{\partial \eta} + [\mathbf{g} \cdot \mathbf{n}], \end{cases} \quad (18)$$

$$\begin{cases} \mu(u_{xx} + u_{yy}) - p_x = g_1, \\ [u] = 0, \\ [\mu u_n] = \hat{f}_2 \sin\theta; \end{cases} \quad (19)$$

$$\begin{cases} \mu(v_{xx} + v_{yy}) = p_y - g_2, \\ [v] = 0, \\ [\mu v_n] = -\hat{f}_2 \cos\theta, \end{cases} \quad (20)$$

where $\frac{\partial}{\partial \eta}$ is the tangential derivative.

Usually we know g_1 , g_2 , $\hat{f}_1$, and $\hat{f}_2$ either from the partial differential equations or from the physics of problems. Thus we can solve the three Poisson equations for p , u , v respectively by the method proposed in Section 2.

3.1 Couple the AMR with the level set method for moving interface problems

Using the level set method, the interface is updated by solving the Hamilton-Jacobi equation:

$$\varphi_t + \mathbf{u} \cdot \nabla \varphi = 0. \quad (21)$$

Since our AMR approach is also based on level set representations, the AMR can be easily incorporated into the level set method in evolving the moving interface. The basic idea is that at one time step we solve the velocity (u , v) on the adaptive mesh with temporarily frozen interface. Then we use the computed velocity to solve the Hamilton-Jacobi equation to update the level set function $\varphi(x, y, t)$. The process then is repeated.

We explain some implementation details here. For the Hamilton-Jacobi equation, we use the third-order WENO scheme for the spatial discretization and the Adam-Bashforth scheme for the time discretization

$$\varphi^{k+1} = \varphi^k + \Delta t(3R^k - R^{k-1})/2, \quad (22)$$

where $R^k = -(\mathbf{u} \cdot \mathbf{n}) |\nabla \varphi^k|$.

As a common practice nowadays, the level set function is only updated within a narrow band $|\varphi(x, y, t)| \leq \lambda_1 h$, where h is the finest resolution in the adaptive mesh.

We generate the finest mesh from the band $|\varphi(x, y, t)| \leq \lambda_2 h$ using the method explained in Section 2, and we require $\lambda_2 > \lambda_1$ to guarantee that the level set is updated within the finest mesh. We do not generate a new AMR at each time step. We only update the AMR when the moving band $|\varphi(x, y, t)| \leq \lambda_1 h$ tends to go outside of the current finest mesh level. In this way, we save considerably the computational cost as well as the storage while keeping the advantages of the AMR approach.

After the level set function $\varphi(x, y, t)$ is updated in each time step, it is crucial to re-initialize the level set function to an approximate signed distance function $|\nabla \varphi| \sim 1$, see for example, [5, 20]. The purpose is to avoid the level set function being too flat or steep. We do the re-initialization within a band $|\varphi(x, y, t)| \leq \lambda_3 h$, where $\lambda_3 > \lambda_2$ to avoid the pollution error for the cut-off boundary of the computational band. For the test examples in this paper, we use $|\varphi(x, y, t)| \leq 6h$ for updating the level set function, $|\varphi(x, y, t)| \leq 8h$ for the refined mesh (AMR), and $|\varphi(x, y, t)| \leq 12h$ for the re-initialization process.

Using the level set method, it is known that the area may not be exactly preserved and the error can accumulate. There are variety of ways proposed in the literature to overcome this problem, particularly, the one described in [19, 21]. In this paper, we propose a different but simpler area preserving strategy as discussed below. We use the initial enclosed area A_0 as the base area. After we update the level set function at the time step t_k , we first carry out the re-initialization process, and then compute the enclosed area A_k the interface length L_k . According to the area loss $A_0 - A_k$, we need to expand current interface in normal direction

by $\varepsilon = \frac{A_0 - A_k}{L_k}$ by making correction $\phi(x, y, t) = \phi(x, y, t) - \varepsilon$, since $|\nabla\phi| \sim 1$ after the re-initialization. If $A_0 < A_k$ then $\varepsilon < 0$, that means we should compress interface by ε to reduce enclosed area. Our numerical experiments show that the mass loss after this correction is always less than 0.01% for long time simulations.

As a summary of this section, we outline our AMR-IIM coupled with the level set method for moving interface problems below.

This procedure can also be used to solve Navier-Stokes equation for moving interface problems if we solve p, u, v by projection method in step 4.

3.2 An example of moving interface driven by the surface tension

In this example, we use the Stokes equations to simulate the motion of an interface initially distorted as a star shape, immersed in a fluid. There is no external forces, that is, $g_1 = g_2 = 0$. Due to the surface tension, the interface will relax to its equilibrium which is the circle that has the same area as that enclosed by the initial interface. The only driven force on the fluid is the surface tension with $\hat{f}_1 = \gamma\kappa$ and $\hat{f}_2 = 0$, where γ is the surface tension coefficient and κ is the interface curvature.

Algorithm 1

Solving Stokes equations by the AMR-IIM for moving interface problems

-
- 1: Start with an initial level set $\phi(x, y, 0)$, compute enclosed area A_0 .
 - 2: Generate the AMR based on the band $|\phi(x, y, t)| \leq \lambda_2 h$.
 - 3: **while** $t < t_{stop}$ **do**
 - 4: Obtain the jump conditions from the most updated interface $\phi(x, y, t) = 0$, and then solve the three Poisson equations for p, u, v on the most updated AMR.
 - 5: Use the velocity (u, v) to update the level set $\phi(x, y, t)$ by solving the Hamilton-Jacobi equation $\phi_t + \mathbf{u} \cdot \nabla\phi = 0$ within a narrow band $|\phi(x, y, t)| \leq \lambda_1 h$, where $\lambda_1 < \lambda_2$ to guarantee the interface is moving within the finest level mesh.
 - 6: Make correction on $\phi(x, y, t)$ within updating band $|\phi(x, y, t)| \leq \lambda_1 h$ against initial enclosed area A_0 for mass conservation
 - 7: Re-initialize the level set function within a wide band $|\phi(x, y, t)| \leq \lambda_3 h$, where $\lambda_3 > \lambda_2$ to cover all refinement region.
 - 8: **if** updating band $|\phi(x, y, t)| \leq \lambda_1 h$ tends to move beyond the finest level mesh **then**
 - 9: Generate new AMR based on most updated level set $|\phi(x, y, t)| \leq \lambda_2 h$
 - 10: **end if**
 - 11: $t = t + \Delta t$
 - 12: **end while**
-

We set up the initial shape of interface to be

$$r(\theta) = 0.6 + 0.2 \sin(5\theta), 0 \leq \theta < 2\pi, \quad (23)$$

in the domain $[-1, 1] \times [-1, 1]$. The parameters are chosen as: the surface tension coefficient $\gamma = 0.1$, the fluid viscosity $\mu = 1$. The time step is chosen using the criterion

$$\Delta t = \frac{1}{8} \min\{h, h/\|\mathbf{u}\|_\infty\} \text{ to satisfy the CFL condition.}$$

In Figure4 (a)–(d), we show some snap-shots of the motion of the deformable interface (red solid) and the underlying meshes computed by AMR 40(+2)[2]. We see that the refinement

region can successfully track the moving interface. Our computation records show that the AMR is updated only 8 from the initial shape to the equilibrium state. The interface is evolved hundreds of time steps within each AMR until the narrow band $|\phi(x, y, t) - 0.5|$ tends to touch the border of refinement region.

In Table 3, we show some important quantities at four different times corresponding to the four snapshots in Figure 4. As we expected, the magnitude of the curvature $|\kappa|$ decreases to the one of the equilibrium circle; and the velocity decreases towards zero. The area enclosed by the interface is conserved very well with less than 0.01% area loss. The advantage of AMR can be seen from the last column which is the numbers of grid points in the four snapshots. Using the AMR 40(+2)[2], the number of grid points where we approximate the solution (u, v, p) is about 8000. The resolution in the finest mesh is $h_f = 2/160$ around the interface. If we use the uniform mesh for the same finest resolution, we need at least $160^2 = 25600$ grid points. The AMR usually helps us to reduce more than half the computation load without losing accuracy.

We also show that our AMR-IIM can preserve second order accuracy for the Stokes flow. Since the exact solution is not available, we compare the solution obtained using $m = 40, 80, 160$ with that obtained by using $m = 320$. We can get an approximate order of accuracy from the ratios of successive errors as explained in [7], for a q th-order accurate method, the ratios should close to

$$\left\| \frac{p_{40} - p_{320}}{p_{80} - p_{320}} \right\| \approx \frac{C(1/40)^q - C(1/320)^q}{C(1/80)^q - C(1/320)^q} = \frac{8^q - 1}{4^q - 1} \quad (24)$$

which is 2.3 for a first order method ($q = 1$) and 4.2 for a second order method ($q = 2$). Similarly,

$$\left\| \frac{p_{80} - p_{320}}{p_{160} - p_{320}} \right\| \approx \frac{C(1/80)^q - C(1/320)^q}{C(1/160)^q - C(1/320)^q} = \frac{4^q - 1}{2^q - 1} \quad (25)$$

which is 3 for a first order method and 5 for a second order method. We computed the solution by the AMR with coarse mesh $m = 40, 80, 160$ and 320 and then got their successive error ratios. Table 4 shows that our AMR-IIM method maintains second order accuracy of the IIM.

3.3 Bubble rising through another fluid

In this example, we use the Stokes equations to simulate a bubble rising through a fluid. The motion is due to the density difference in the fluid ρ_{out} and in the bubble ρ_{in} . We study the case in which $\rho_{in} < \rho_{out}$ so that the bubble will rise and meanwhile its shape will be distorted by the surface tension and the gravity. We assume the same viscosity μ for both fluids.

The gravity serves as vertical external force in this example, thus, $g_2 = -\rho g$ ($g = 9.8m/s^2$). There is no horizontal external force, thus, $g_1 = 0$. The singular driven forces are the same as last example, $\hat{f}_1 = \gamma \kappa$, and $\hat{f}_2 = 0$, where γ is surface tension coefficient and κ is the interface curvature. We use the AMR 80(+1)[2] for the simulation on the domain $[-4, 4] \times [-4, 4]$. The initial bubble is set up to be an ellipse centered at $(0, -3.4)$ with major and minor axis being 2×0.6 and 2×0.25 , respectively. Other parameters are $\rho_{in} = 1$, $\rho_{out} = 2$, and $\mu = 1$.

It is known that the deformation of the bubble depends on the surface tension, see for example, [7]. For large surface tension γ , the bubble tends to move to a nearly circular shape as it rises. For small surface tension γ , the bubble tends to be distorted. If γ is

sufficiently small, the bubble would eventually split into pieces. We tested different sizes of the surface tension and obtained consistent results with those in the literature. We are more interested in the case when the bubble splits because it shows the advantage of the level set method in dealing with topological changes.

Figure 5 shows several snap-shots of the bubble with surface tension $\gamma = 0.01$, and the underline meshes as it rises. The refinement region can automatically follow the changes in both the position and the shape of the bubble. The time step size is again chosen as

$\Delta t = \frac{1}{8} \min\{h, h/\|\mathbf{u}\|_\infty\}$. The AMR is updated every 30 steps on average. When the bubble splits into two pieces, the AMR also tend to split. The area loss is less than 0.01% with our new mass conservation strategy (it was 4.32% if nothing is done) by the end of the simulation. even with the singularity and very large magnitude of the curvature during the splitting process.

The advantage of the AMR is significant in reducing the numbers of grid points without affecting the solution accuracy in the neighborhood of the interface. To obtain the finest resolution $4/160$ around the interface, the adaptive mesh requires about 7000 grid points, while the uniform mesh with the finest mesh requires $160^2 = 25600$ grid points. We believe that the AMR would be more advantageous in problems when the simulated objects are small in size but move across a large domain. In those cases, it might be impossible to refine the whole domain uniformly due to the restriction of computer storages. The AMR can overcome this difficulty by utilizing the computational resource more efficiently.

We are also interested in whether the creeping bubble simulation computed on AMR matches that by using the uniform mesh, particularly, after the bubble splits. The bubble creeping problem computed by the IIM on uniform mesh was studied for years, see for example, [7] and the references therein, and the results are believed to be reliable. We compare the simulation results by using the AMR $80(+1)[2]$ with that obtained from the finest uniform 160×160 mesh in Figure 6, which shows the bubble interfaces at $t = 3s$ simulated using the two meshes. We can see that the bubble shape and position computed by the AMR match those obtained by the uniform mesh very well.

4 The AMR-IIM for Navier-Stokes equations with moving interfaces

In this section, we describe how to use the AMR-IIM to solve the full Navier-Stokes equations with fixed or moving interface. Consider the Navier-Stokes equations for an incompressible flow with a moving interface:

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p = \mu \Delta \mathbf{u} + \mathbf{g} + \int_{\Gamma} \mathbf{f}(\mathbf{X}(s, t)) \delta(\mathbf{x} - \mathbf{X}(s, t)) ds, \\ \nabla \cdot \mathbf{u} = 0, \\ \frac{d\Gamma}{dt} = \mathbf{u}(\mathbf{X}(s, t)). \end{cases} \quad (26)$$

The notations are the same as those used for the Stokes equations.

In the IIM for Navier-Stokes equations, see [9, 11], the projection method, see for example, [2] is modified to treat various discontinuities in the velocity and the pressure. The idea is to add correction terms to the right hand sides of the finite difference equations so that the discretization is still second order accurate even with the presence of the interface. Assume a Dirichlet boundary condition for the velocity along Γ is given, the prediction step of the modified projection method is,

$$\begin{cases} \frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + (\mathbf{u} \cdot \nabla_h \mathbf{u})^{n+1/2} + \nabla_h p^{n-1/2} = \frac{\mu}{2}(\Delta_h \mathbf{u}^* + \Delta_h \mathbf{u}^n) + \mathbf{g}^{n+1/2} + \mathbf{C}_1^n, \\ \mathbf{u}^*|_{\partial\Omega} = \mathbf{u}_b^{n+1}, \end{cases} \quad (27)$$

where $(\mathbf{u} \cdot \nabla_h \mathbf{u})^{n+1/2}$ is approximated by,

$$(\mathbf{u} \cdot \nabla_h \mathbf{u})^{n+1/2} = \frac{3}{2}(\mathbf{u}^n \cdot \nabla_h) \mathbf{u}^n - \frac{1}{2}(\mathbf{u}^{n-1} \cdot \nabla_h) \mathbf{u}^{n-1} + \mathbf{C}_2^n. \quad (28)$$

The prediction step involves solving two Helmholtz equations with finite discontinuities in the pressure and the gradient of the velocity. The projection step is the following,

$$\begin{cases} \nabla_h \phi^{n+1} = \frac{\nabla_h \cdot \mathbf{u}^*}{\Delta t} + \mathbf{C}_3^n, \\ \frac{\partial \phi^{n+1}}{\partial n}|_{\partial\Omega} = 0. \end{cases} \quad (29)$$

The velocity and the pressure are updated by,

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla_h \phi^{n+1} + \mathbf{C}_4^n, \quad (30)$$

$$\nabla_h p^{n+1/2} = \nabla_h p^{n-1/2} + \nabla_h \phi^{n+1} + \mathbf{C}_5^n, \quad (31)$$

$$p^{n+1/2} = p^{n-1/2} + \phi^{n+1} - \frac{\mu}{2}(\nabla_h \cdot \mathbf{u}^* + \mathbf{C}_6^n) \quad (32)$$

The correction terms $\mathbf{C}_1^n - \mathbf{C}_6^n$ are computed similarly with what we did for the elliptic interface problems in Section 2, see also [11]. The jump conditions of p , u , v in Navier-Stokes equations are the same as in the Stokes equation (18)–(20) since the material derivative is continuous. It is also shown in [11] that the projection-IIM method is second order accurate.

When we use the projection-IIM method on the AMR meshes, we only need all the correction terms at irregular grid points in the finest mesh. We implement the standard projection method among multiple mesh levels at other grid points. We need to find the discrete gradients and Laplacians such as $\nabla_h \mathbf{u}$, ∇_{hp} , and $\Delta_h \mathbf{u}$ in different mesh resolutions. The procedure is similar to what we described in Section 2. For moving interface problems, we use the same interface evolution algorithm demonstrated in Section 3 except replacing the three Poisson approach to be projection-IIM method.

4.1 An example of a fixed interface: Circular flow

We test the adaptive version of the projection-IIM method through the example from [10] to check the convergence of our method. We solve the Navier-Stokes equation involving a fixed circular interface in the domain $[-1, 1] \times [-1, 1]$. The exact solution is known as:

$$u(x, y, t) = \begin{cases} \sin(t)(\frac{y}{r} - 2y) & \text{if } r \geq 1/2, \\ 0 & \text{otherwise;} \end{cases} \quad (33)$$

$$v(x, y, t) = \begin{cases} \sin(t)(-\frac{x}{r} + 2x) & \text{if } r \geq 1/2, \\ 0 & \text{otherwise;} \end{cases} \quad (34)$$

$$p(x, y, t) = \begin{cases} \cos(\pi x) \cos(\pi y) & \text{if } r \geq 1/2, \\ 0 & \text{otherwise;} \end{cases} \quad (35)$$

where $r = \sqrt{x^2 + y^2}$. The external force $\mathbf{g} = (g_1, g_2)$, singular force $\hat{f}_1, \hat{f}_2$ and the Dirichlet boundary condition in the Navier-Stokes equation can be all derived from the exact solution. For example, the normal singular force $\hat{f}_1 = [p] = \cos(\pi x) \cos(\pi y)$, and the tangential singular force $\hat{f}_2 = [\mu u_n] / \sin \theta = -2\mu \sin(\theta)$. The rest of the jump conditions and the correction terms are described in [11] for the Navier-Stokes equation.

We tested the adaptive version of projection-IIM for two cases, $\mu = 0.5$ (small Reynolds number) and $\mu = 0.005$ (modest Reynolds number) from $t = 0$ to $t = 4$ with the time step

$\Delta t = \frac{1}{2} \min\{h, h/\|\mathbf{u}\|_\infty\}$. In Table 5 and Table 6, we show the grid refinement analysis of the convergence of the velocity and the pressure against the exact solutions. The order of accuracy is estimated by,

$$\text{order} = \frac{\log(E(m)/E(2m))}{\log 2}$$

From Table 5 and 6, we can see that second order accuracy is achieved for both the velocity and the pressure by the adaptive version of projection-IIM.

4.2 Oscillations of the interface near the equilibrium

We revisit the surface tension problem described in Section 3.2 again to see what would be different if we modeled the flow by the Navier-Stokes equation. The initial shape of the interface as an ellipse: $rx^2/r_a^2 + y^2/r_b^2 = 1$ at the center of domain $[-1, 1] \times [-1, 1]$. We take the initial semi-major axes $r_a = 0.8$, and the semi-minor axes $r_b = 0.2$. Due to the surface tension on the interface, we expect that the ellipse would relax to a circle interface to minimize the energy. That means both r_a and r_b would converge to $\sqrt{0.8 \times 0.2} = 0.4$ eventually due to the area conservation.

We run this simulation on the AMR 80(+1)[2] with various viscosity μ but same surface tension coefficient $\gamma = 1$. In Figure 7, we show the history of r_a and r_b as functions of time. When μ is relatively large, say $\mu = 1$, the viscosity term $\mu \nabla^2 \mathbf{u}$ is the dominant term in the Navier-Stokes equation and the interface converges to its equilibrium position monotonically. In this case, the motion of interface is very close to that modeled by the Stokes equations discussed in Section 3.2. When μ becomes smaller, the inertia term

$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u}$ becomes dominant and some oscillations of the interface around the equilibrium can be observed clearly.

We are more interested in the oscillation case when viscosity $\mu = 0.02$. Eight snapshots are shown in Figure 8 for the motion of interface. It can be observed that the shape of interface are not always ellipses, but sometimes like dumbbells. The results are consistent with the observations in the literature.

5 Conclusion

In this paper, we have developed an adaptive mesh refinement strategy of the Immersed Interface Method applied for flow problems with a fixed or moving interface. The work is

built on the AMR method developed for two-dimensional elliptic interface problems in the paper [12]. Using this approach, we can reduce the number of unknowns significantly by focusing on the resolution near the interface while maintain second order accuracy. Numerical experiments verified that the AMR-IIM is working efficiently for some flow problems.

Acknowledgments

The work of this paper was partially supported by the US ARO grants 550694-MA, the AFSOR grant FA9550-12-1-0188, the US NSF grant DMS-0911434, and the NIH grant 096195-01.

References

- Berger M, Colella P. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.* 1989; 82:64–84.
- Brown DL, Cortez R, Minion ML. Accurate projection methods for the incompressible Navier-Stokes equations. *J. Comput. Phys.* 2001; 168:464.
- Griffith, BE. Technical report. New York University; 2009. A comparison of two adaptive versions of the immersed boundary method.
- Griffith BE, Hornung RD, McQueen DM, Peskin CS. An adaptive, formally second order accurate version of the immersed boundary method. *J. Comput. Phys.* 2007; 223:10–49.
- Jiang G, Peng D. Weighted ENO schemes for Hamilton–Jacobi equations. *SIAM J. Sci. Comput.* 2000; 21:2126–2143.
- LeVeque RJ, Li Z. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM J. Numer. Anal.* 1994; 31:1019–1044.
- LeVeque RJ, Li Z. Immersed interface method for Stokes flow with elastic boundaries or surface tension. *SIAM J. Sci. Comput.* 1997; 18:709–735.
- Li Z, Ito K. Maximum principle preserving schemes for interface problems with discontinuous coefficients. *SIAM J. Sci. Comput.* 2001; 23:1225–1242.
- Li, Z.; Ito, K. The Immersed Interface Method – Numerical Solutions of PDEs Involving Interfaces and Irregular Domains. *SIAM Frontier Series in Applied mathematics*, FR33; 2006.
- Li Z, Ito K, Lai M-C. An augmented approach for Stokes equations with a discontinuous viscosity and singular forces. *Computers and Fluids*. 2007; 36:622–635.
- Li Z, Lai M-C. The immersed interface method for the Navier-Stokes equations with singular forces. *J. Comput. Phys.* 2001; 171:822–842.
- Li Z, Song P. An adaptive mesh refinement strategy for immersed boundary/interface methods. *Commun. Comput. Phys.* 2011 in press.
- Losasso F, Fedkiw R, Osher S. Spatially adaptive techniques for level set methods and incompressible flow. *Computers and Fluids*. 2006; 35:995–1010.
- Peskin CS, McQueen DM. A general method for the computer simulation of biological systems interacting with fluids. *Symposia of the Society for Experimental Biology*. 1995; 49:265. [PubMed: 8571229]
- Roma A, Peskin CS, Berger M. An adaptive version of the immersed boundary method. *J. Comput. Phys.* 1999; 153:509–534.
- Ruge, JW.; Stuben, K. Algebraic multigrid. In: McCormick, SF., editor. *Multigrid Method*. Philadelphia: SIAM; 1987.
- Strain J. Fast tree-based redistancing for level set computations. *J. Comput. Phys.* 1999; 152:664–686.
- Sussman M, Almgren A, Bell JB, Colella P, Howell LH, Welcome ML. An adaptive level set approach for incompressible two-phase flows. *J. Comput. Phys.* 1999; 148:81–124.
- Sussman M, Fatemi E, Smereka P, Osher S. An improved level set method for incompressible two-phase flows. *Computers and Fluids*. 1998; 27:663–680.

20. Sussman M, Smereka P, Osher S. A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.* 1994; 114:146–159.
21. Xu J, Li Z, Lowengrub J, Zhao H. A level-set method for interfacial flows with surfactant. *J. Comput. Phys.* 2006; 212:590–616.

Figure 1.

Adaptive mesh around a circular interface (red solid). Parent grid points (starred) are selected within band $|\phi_{ij}| = |\phi(x_i, y_j)| - h$ (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.

Figure 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.

(a)

(b)

Figure 3.

(a), Solution plot of (16); (b) The interface (red) and the AMR $80(+1)[2]$ mesh.

(a) $t = 0.001s$ (b) $t = 6.250s$ (c) $t = 12.50s$ (d) $t = 31.25s$ **Figure 4.**

Snap-shots of a deformable interface (red-solid in the middle of the fine mesh) relaxing to its equilibrium and underline AMR $40(+2)[2]$ meshes.

Figure 5.

A rising bubble (the red solid curve in the middle of the fine mesh) at time $t = 0s, 0.720s, 1.30s, 2.28s, 3.64s$, computed by using the AMR $80(+1)[2]$.

Figure 6.
The bubble interface at $t = 3s$ computed by uniform mesh 160×160 and by the AMR $80(+1)[2]$.

Figure 7.

The history of semi-major axes r_a (red) and semi-minor axes r_b (blue) over time for different viscosity μ : (a), $\mu = 1$; (b), $\mu = 0.1$; and (c), $\mu = 0.02$. Oscillations are observed when the viscosity is small.

19

Figure 8. Snapshots of oscillating interface (red solid) around equilibrium states (blue dashed) when viscosity $\mu = 0.02$, surface tension coefficient $\gamma = 1$.

Table 1

A comparison of the grid refinement analysis for a variable coefficient with a discontinuity using a uniform mesh and AMR-IIM meshes with the same finest resolution h .

	Uniform				AMR: $m(+1)/2$			
	m	$\ E_h\ _{\infty}$	Grid points	CPU(s)	m	$\ E_h\ _{\infty}$	Grid points	CPU(s)
Finest h								
2/40	40	6.40e-4	1521	0.02	20	1.07e-3	805	0.02
2/80	80	1.40e-4	6241	0.09	40	2.69e-4	2745	0.03
2/160	160	2.98e-5	25281	0.39	80	6.54e-5	10201	0.17
2/320	320	7.20e-6	101761	1.51	160	1.69e-5	39225	0.65
2/640	640	1.75e-6	408321	5.66	320	4.18e-6	153805	2.67

Table 2

A comparison of the grid refinement analysis for a star-shaped interface using uniform and AMR-IIM meshes with the same finest resolution h .

	Uniform				AMR: $m(+1)/2$			
	m	$\ E_h\ _{\infty}$	Grid points	CPU(s)	m	$\ E_h\ _{\infty}$	Grid points	CPU(s)
Finest h								
2/40	40	1.07e-2	1521	0.02	20	1.06e-2	1093	0.02
2/80	80	1.53e-3	6241	0.09	40	1.46e-3	3225	0.08
2/160	160	1.16e-4	25281	0.34	80	9.26e-5	9705	0.16
2/320	320	2.99e-5	101761	1.5	160	2.35e-5	32253	0.53
2/640	640	3.46e-6	408321	5.8	320	5.52e-6	115757	2.06

Some important quantities at four different times corresponding to the four snapshots in Figure 4 (a)–(d).

Table 3

Figure	t	$ \kappa _{\max}$	$p_{\max}$	$l_{\text{ull}}_{\max}$	Area	Grid points
a	0.001	28.40	1.6010	0.2455e-1	1.194	8116
b	6.250	5.864	0.4282	0.1611e-1	1.194	8181
c	12.50	3.276	0.2509	0.6240e-2	1.194	7876
d	31.25	1.669	0.1645	0.1563e-3	1.194	7761

Table 4

The errors of p , u , v computed by the AMR $m(+1)[2]$ compared to that obtained by the AMR 320(+1)[2] at $t = 0$.

m	$\ p_m - p_{320}\ _\infty$	ratio	$\ u_m - u_{320}\ _\infty$	ratio	$\ v_m - v_{320}\ _\infty$	ratio
40	3.23e-2		5.66e-3		2.30e-3	
80	7.25e-3	4.45	1.32e-3	4.28	4.32e-4	5.32
160	9.51e-4	7.62	2.94e-4	4.49	5.53e-5	7.81

Table 5

A grid refinement analysis for $\mu = 0.5$.

AMR $m(+1)[2]$	$\ E_u\ _\infty$	order	$\ E_p\ _\infty$	order
32	1.12e-3		1.01e-2	
64	3.07e-4	1.9	2.34e-3	2.1
128	8.40e-5	1.9	6.31e-4	1.9

Table 6

A grid refinement analysis for $\mu = 0.005$

AMR $m(+1)[2]$	$\ E_u\ _\infty$	order	$\ E_p\ _\infty$	order
32	1.49e-2		9.31e-3	
64	2.52e-3	2.5	2.13e-3	2.1
128	6.34e-4	2.0	5.42e-4	2.0