



A well-conditioned augmented system for solving Navier–Stokes equations in irregular domains

Kazufumi Ito^a, Ming-Chih Lai^b, Zhilin Li^{a,*}

^a Center for Research in Scientific Computation and Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205, United States

^b Department of Applied Mathematics, National Chiao Tung University, 1001, Ta Hsueh Road, Hsinchu 30050, Taiwan

ARTICLE INFO

Article history:

Received 31 October 2007

Received in revised form 20 October 2008

Accepted 17 December 2008

Available online 31 December 2008

AMS classification:

65M06

65M12

76T05

Keywords:

Navier–Stokes equations

Embedding technique

Immersed interface method

Irregular domain

Augmented system

Projection method

Level set representation

ABSTRACT

An augmented method based on a Cartesian grid is proposed for the incompressible Navier–Stokes equations in irregular domains. The irregular domain is embedded into a rectangular one so that a fast Poisson solver can be utilized in the projection method. Unlike several methods suggested in the literature that set the force strengths as unknowns, which often results in an ill-conditioned linear system, we set the jump in the normal derivative of the velocity as the augmented variable. The new approach improves the condition number of the system for the augmented variable significantly. Using the immersed interface method, we are able to achieve second order accuracy for the velocity. Numerical results and comparisons to benchmark tests are given to validate the new method. A lid-driven cavity flow with multiple obstacles and different geometries are also presented.

© 2008 Elsevier Inc. All rights reserved.

1. Introduction

In this paper, we consider the following incompressible Navier–Stokes equations in an irregular domain $R \setminus \Omega$:

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) + \nabla p = \mu \Delta \mathbf{u} + \mathbf{G}, \quad \mathbf{x} \in R \setminus \Omega, \quad (1.1)$$

$$\nabla \cdot \mathbf{u} = 0, \quad \mathbf{x} \in R \setminus \Omega, \quad (1.2)$$

$$\mathbf{u}|_{\partial R} = \mathbf{u}_{\partial R}, \quad \mathbf{u}|_{\partial \Omega} = \mathbf{u}_{\partial \Omega}, \quad \text{BC}, \quad (1.3)$$

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0, \quad \text{IC}, \quad (1.4)$$

where $\mathbf{G}(x, y, t)$ is an external forcing term, R is a rectangular domain, and Ω is a set of inclusions (obstacles), see Fig. 1 for an illustration. We assume that $\rho = 1$ in this paper.

Since the inclusion Ω is irregular inside a regular computational domain R , one interesting numerical issue is how to impose the prescribed velocity condition at the immersed boundary $\partial \Omega$, particularly, when the boundary is moving. While

* Corresponding author.

E-mail address: zhilin@math.ncsu.edu (Z. Li).

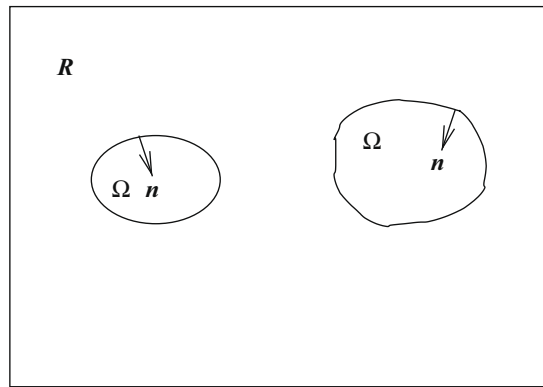


Fig. 1. A diagram for the Navier–Stokes equations on an irregular domain.

there are a number of numerical methods for solving Navier–Stokes equations in irregular domains, we are interested in Cartesian grid based finite difference methods. One of the main motivations using Cartesian grid methods is to avoid mesh generation for moving boundary problems. Cartesian grid methods also enable us to use some commonly used fast solvers such as FFT for solving Poisson/Helmholtz equations.

Among Cartesian grid based finite difference methods, one common approach is to use a streamfunction–vorticity formulation, see for example, [3,20,16]. In these methods, the domain is embedded into a rectangular one so that the immersed interface method can be used to get second order accuracy for the vorticity. But this formulation is often restricted to two-dimensional problems only.

A second type of finite difference methods is based on the projection method for small to medium Reynolds number flow, see for example, [9,21]. This type of method has been modified to compute large eddy simulations as well, see for example, [19] for a brief review and the references therein. In these methods, the boundary is treated as an immersed interface that exerts forces to the surrounding fluid. The force density is chosen so that the boundary condition is satisfied. The problem in finding such force strengths is an inverse problem and it is ill-conditioned. Different strategies have been developed to regularize the problem. In [22], the method was coupled with an optimal control; in [10], singular value decomposition is used. In most of these methods, the boundary is represented by a set of control points (or a cubic spline). The condition number of the system of equations is often very large and depends on the number of control points. The ill-conditioned system may affect the accuracy of the computed solution in an adversely way.

Other related work can be found in [5,17]. The method in [5] used a direct discretization for a similar but different problem. Advantages of the method proposed in [5] are its simplicity and symmetry of the system of equations on the entire domain. The PCG method was applied to the entire system with $O(N^2)$ unknowns at every time step. One of disadvantages is that a fast Poisson solver can not be used. The stair-case approximation of the boundary proposed in [5] is less accurate than our approach that takes into account of the curvature of the boundary. In [18], the authors proposed a Cartesian method to simulate the incompressible flows around stationary or moving immersed boundaries. The finite difference discretization near the interface or boundary is modified so that an algebraic multigrid solver can be applied. Again, a fast Poisson solver based on the FFT can not be applied there. In an earlier work of Ye et al. [23], the authors used a finite volume method for a similar problem.

In order to overcome the difficulty, many attempts have been made to utilize a Helmholtz/Poisson solver on irregular domains based on a second order augmented immersed interface method in [6,12,15]. In an augmented immersed interface method, an augmented variable (often is one dimension lower than that of the solution) is introduced so that the IIM can be easily implemented. The solution then is a functional of the augmented variable and should satisfy the boundary or interface condition, which is the augmented equation. In this paper we develop such a method for the incompressible Navier–Stokes equations in irregular domains.

By investigating the source of the stiffness, we have found that the stiffness is from the force in the normal direction that corresponds to the jump in the pressure. Based on the fact that the pressure in Navier–Stokes equations is not a free variable, we propose a new method that sets the jump in the normal derivative of the velocity as augmented variables. This new method improves the condition of the Schur complement system for the augmented variables significantly compared to the method that sets the force density as the augmented variable.

In an augmented method, it is relatively easy to choose and test different augmented variables with a few modifications, see [11,12]. In [13], the jump in the velocity was chosen as the augmented variable for the two-phase incompressible Stokes equations. We choose the jump in the normal derivative of the velocity for the Navier–Stokes equations because it is easier to deal with the nonlinear term since the velocity is continuous.

We use a zero level set of a Lipschitz continuous function to represent the boundary which is more flexible for multi-connected domains. At each time step, the linear system of equations for the augmented variable can be solved by the GMRES method, or by LU decomposition if we form the coefficient matrix at the first time step. The latter approach is especially efficient for stationary boundary problems and long time runs.

The rest of the paper is organized as follows. In Section 2, we present an augmented method for solving (1.1)–(1.4) within the framework of the projection method. The novelty of our approach and the implementation issues are described in detail. In Section 3, numerical results for non-trivial examples are presented and analyzed. Conclusions will be given in Section 4.

2. The numerical algorithm

Our numerical method is based on the projection method for solving the incompressible Navier–Stokes equations. There are several versions of the projection method for solving the incompressible Navier–Stokes equations, see for example, [1,7,8] and many others. The projection method that we used in our method is the one described in [2] which is based on the pressure increment formulation of [1,7].

As mentioned before, we assume that the domain R is a rectangle $[a, b] \times [c, d]$ with holes Ω . The spatial spacing is chosen as $h_x = (b - a)/M, h_y = (d - c)/N$, where M and N are the number of grid points in the x and y -directions, respectively. We use a standard uniform Cartesian grid for simplicity. The time integration from t^k to t^{k+1} can be written as:

$$\frac{\mathbf{u}^* - \mathbf{u}^k}{\Delta t} + (\mathbf{u} \cdot \nabla \mathbf{u})^{k+\frac{1}{2}} = \begin{cases} -\nabla p^{k-\frac{1}{2}} + \frac{\mu}{2}(\Delta \mathbf{u}^* + \Delta \mathbf{u}^k) + \mathbf{G}^{k+\frac{1}{2}} & \text{if } \mathbf{x} \in R \setminus \Omega \\ \frac{\mu}{2}(\Delta \mathbf{u}^* + \Delta \mathbf{u}^k) & \text{if } \mathbf{x} \in \Omega \end{cases} \tag{2.5}$$

$$\mathbf{u}^*|_{\partial R} = \mathbf{u}_{\partial R}(\mathbf{x}_{\partial R}, t^{k+1}), \tag{2.6}$$

$$[\mathbf{u}^*]_{\partial \Omega} = 0, \quad \left[\frac{\partial \mathbf{u}^*}{\partial n} \right]_{\partial \Omega} = \mathbf{q}^{k+1}, \quad \mathbf{u}^*|_{\partial \Omega} = \mathbf{u}_{\partial \Omega}(\mathbf{x}_{\partial \Omega}, t^{k+1}), \tag{2.7}$$

$$\begin{cases} \Delta \phi^{k+1} = \frac{\nabla \cdot \mathbf{u}^*}{\Delta t}, & \mathbf{x} \in R, \\ \left[\frac{\partial \phi^{k+1}}{\partial n} \right]_{\partial R} = 0, & [\phi^{k+1}]_{\partial \Omega} = 0, \quad \left[\frac{\partial \phi^{k+1}}{\partial n} \right]_{\partial \Omega} = 0, \end{cases} \tag{2.8}$$

$$\mathbf{u}^{k+1} = \mathbf{u}^* - \Delta t \nabla \phi^{k+1}, \quad \mathbf{x} \in R, \tag{2.9}$$

where $(\mathbf{u} \cdot \nabla \mathbf{u})^{k+\frac{1}{2}}$ is approximated by

$$(\mathbf{u} \cdot \nabla \mathbf{u})^{k+\frac{1}{2}} = \frac{3}{2}(\mathbf{u}^k \cdot \nabla) \mathbf{u}^k - \frac{1}{2}(\mathbf{u}^{k-1} \cdot \nabla) \mathbf{u}^{k-1}. \tag{2.10}$$

The solution \mathbf{u}^* above is a functional of the augmented variable \mathbf{q}^{k+1} which should be determined by imposing the boundary condition

$$\mathbf{u}^*|_{\partial \Omega} = \mathbf{u}_{\partial \Omega}(\mathbf{x}_{\partial \Omega}, t^{k+1}). \tag{2.11}$$

The Eqs. (2.5)–(2.11) now become a complete system for $(\mathbf{u}^*, \mathbf{q}^{k+1}, \phi^{k+1}, \mathbf{u}^{k+1})$. Once we have solved this system, then the pressure is determined from

$$p^{k+1/2} = p^{k-1/2} + \phi^{k+1}, \quad \mathbf{x} \in R \setminus \Omega. \tag{2.12}$$

Table 1

A grid refinement analysis against the exact solution at final time $T = 5$. $\|E_u\|_\infty$ is the maximal error in the velocity, $\|E_p\|_\infty$ is the maximal error in the pressure, $cond_1$ is the condition number of the coefficient matrix for the unknown jump in the normal derivative of the velocity, while $cond_2$ is the condition number of the coefficient matrix when we set the normal and tangential force strengths as the unknowns.

(a) Dirichlet BC on all sides, $\mu = 0.5$.							
$M \times N$	$\ E_u\ _\infty$	r_1	$\ E_p\ _\infty$	r_2	$cond_1$	$cond_2$	CPU(s)
64 × 32	4.8913 × 10 ⁻³		3.3494 × 10 ⁻²		17.686	4.4674 × 10 ³	0.8656
128 × 64	1.1499 × 10 ⁻³	2.0887	8.3891 × 10 ⁻³	1.9973	19.206	1.0545 × 10 ⁴	5.2305
256 × 128	2.7371 × 10 ⁻⁴	2.0708	5.8427 × 10 ⁻³	0.5219	28.046	8.5386 × 10 ⁵	42.072
512 × 256	6.6709 × 10 ⁻⁵	2.0367	2.5283 × 10 ⁻³	1.2085	46.338	1.0199 × 10 ⁷	281.12
(b) Dirichlet BC on all sides, $\mu = 0.005$.							
$M \times N$	$\ E_u\ _\infty$	r_1	$\ E_p\ _\infty$	r_2	$cond_1$	$cond_2$	
64 × 32	1.0918 × 10 ⁻¹		3.1598 × 10 ⁻²		58.888	6.4600 × 10 ⁴	
128 × 64	1.7605 × 10 ⁻²	2.6326	6.5403 × 10 ⁻³	2.2724	19.356	1.1110 × 10 ⁶	
256 × 128	2.9241 × 10 ⁻³	2.5899	1.2525 × 10 ⁻³	2.3845	13.447	3.1600 × 10 ⁷	
512 × 256	3.4913 × 10 ⁻⁴	3.0622	2.7237 × 10 ⁻⁴	2.2012	9.7787	1.006 × 10 ¹⁰	
(c) Dirichlet BC for u on $x = a$, for v on $x = a, y = c$, and $y = d$. Neumann BC for u on $x = c, y = c$, and $y = d$, for v on $x = b, \mu = 0.005$. The condition numbers are almost the same as in table (b) above so we did not list them.							
$M \times N$	$\ E_u\ _\infty$	r_1	$\ E_p\ _\infty$	r_2			
64 × 32	8.8951 × 10 ⁻²		1.3054 × 10 ⁻¹				
128 × 64	1.7636 × 10 ⁻²	2.3344	3.4977 × 10 ⁻²				1.9000
256 × 128	2.9041 × 10 ⁻³	2.6024	1.4110 × 10 ⁻²				1.3097
512 × 256	3.4473 × 10 ⁻⁴	3.0745	6.6851 × 10 ⁻³				1.0785

There are two novel ideas that are significantly different from other methods in the literature. First, we set the jump in the normal derivative of the velocity as the augmented variable so that the system (2.5)–(2.7) is complete for $(\mathbf{u}^*, \mathbf{q}^{k+1})$. Here, \mathbf{q}^{k+1} represents the augmented variable, and the augmented equation is $\mathbf{u}^*|_{\partial\Omega} = \mathbf{u}_{\partial\Omega}(\mathbf{x}_{\partial\Omega}, t^{k+1})$. Secondly, we apply the augmented approach for the coupled system of $(\mathbf{u}^*, \mathbf{q}^{k+1})$ so that a fast Helmholtz solver can be applied when we use GMRES to solve the Schur-complement system for the augmented variable \mathbf{q}^{k+1} , see Section 2.3 for more details.

We also have tested the method by embedding the Navier–Stokes equations to the entire domain and selecting the normal and tangential forces along the boundary as augmented variables as described in several papers in the literature. The resulting Schur-complement system is severely ill-conditioned, see Table 1 for an illustration. But our selection of the augmented variable $[\frac{\partial \mathbf{u}^*}{\partial \mathbf{n}}]_{\partial\Omega} = \mathbf{q}^{k+1}$ results in a very well-conditioned system. Since we are only interested in the solution in the region of $R \setminus \Omega$, the other terms such as $(\mathbf{u} \cdot \nabla \mathbf{u})^{k+1/2}$ and $\nabla p^{k-1/2}$ can be treated as a forcing term in the elliptic equations that do not need to be extended. These quantities at irregular grid points where the boundary $\partial\Omega$ cuts through the standard central 5-point stencil, can be approximated by a one-sided interpolation scheme.

Note that Eq. (2.8) is also solved on the entire rectangular domain R so that the same fast Poisson solver can be used. Note also that, the above algorithm is equivalent to the following formulation without introducing any augmented variable in the domain $R \setminus \Omega$:

$$\begin{aligned} \frac{\mathbf{u}^* - \mathbf{u}^k}{\Delta t} + (\mathbf{u} \cdot \nabla \mathbf{u})^{k+1/2} &= -\nabla p^{k-1/2} + \frac{\mu}{2}(\Delta \mathbf{u}^* + \Delta \mathbf{u}^k) + \mathbf{G}^{k+1/2}, \quad \mathbf{x} \in R \setminus \Omega \\ \mathbf{u}^*|_{\partial R} &= \mathbf{u}_{\partial R}(\mathbf{x}_{\partial R}, t^{k+1}), \quad \mathbf{u}^*|_{\partial\Omega} = \mathbf{u}_{\partial\Omega}(\mathbf{x}_{\partial\Omega}, t^{k+1}) \\ \Delta \phi^{k+1} &= \frac{\nabla \cdot \mathbf{u}^*}{\Delta t}, \quad \mathbf{x} \in R \setminus \Omega, \quad \left. \frac{\partial \phi^{k+1}}{\partial \mathbf{n}} \right|_{\partial R} = 0, \\ \mathbf{u}^{k+1} &= \mathbf{u}^* - \Delta t \nabla \phi^{k+1}, \quad \mathbf{x} \in R \setminus \Omega, \\ \nabla \cdot \mathbf{u}^{k+1} &= 0, \quad \mathbf{x} \in R \setminus \Omega. \end{aligned}$$

Remark 1. The projection method that we currently used is a version based on the pressure increment formulation [1,2,7] in which the no-slip boundary conditions at the embedded boundary $\partial\Omega$ and the physical boundary ∂R are imposed for the intermediate velocity \mathbf{u}^* instead of \mathbf{u}^{k+1} . This kind of simplification is made to be consistent with the efficiency of the projection method of Navier–Stokes solver and has also been used in [21]. Since we are interested in the solution outside the embedded boundary, (we disregard the computed solution inside the boundary), all we need to care is whether the solution satisfies the no-slip boundary condition in a suitable accuracy comparable with the fluid solver accuracy. As discussed in [4], the spurious slip velocity error of \mathbf{u}^{k+1} at the boundary is of second order. One may think of such error could cause the possible fluid penetration (leakage) into or out of the embedded boundary. In fact, we have also implemented the following alternative approach by setting

$$\mathbf{u}^{k+1}|_{\partial\Omega} = \mathbf{u}_{\partial\Omega}(\mathbf{x}_{\partial\Omega}, t^{k+1})$$

as the constraint (the augmented equation) instead of using

$$\mathbf{u}^*|_{\partial\Omega} = \mathbf{u}_{\partial\Omega}(\mathbf{x}_{\partial\Omega}, t^{k+1}).$$

In such case, the entire system for solving $(\mathbf{u}^{k+1}, \mathbf{q}^{k+1}, p^{k+1})$ with the augmented variable $[\frac{\partial \mathbf{u}^*}{\partial \mathbf{n}}]_{\partial\Omega} = \mathbf{q}^{k+1}$ are all coupled together. This approach is more expensive since each iteration requires to solve one more Poisson equation for ϕ^{k+1} . However, the results are almost the same. From the numerical experiments that we conduct in this paper, the effect of fluid penetration in present simulations caused by the enforcement of no-slip condition for \mathbf{u}^* instead of \mathbf{u}^{k+1} is not that significant. As a matter of fact, for the simulation of flow past a circular cylinder in Section 3, our present method obtains comparable results with others by comparing the drag, lift and Strouhal numbers.

2.1. Some implementation details

The method can be implemented with different representations of the boundary $\partial\Omega$. If a front-tracking method is used, then the augmented variable $\mathbf{q}^{k+1} = [\partial \mathbf{u}^* / \partial \mathbf{n}]$ is defined at a set of control points (Lagrangian markers) as discussed in [21,10,22]. In our present scheme, the boundary $\partial\Omega$ is represented by a zero level set of a Lipschitz continuous function (often an approximated signed distance function). The augmented variable is defined at those orthogonal projections of irregular grid points¹ located outside the domain Ω . The level set representation is more flexible to deal with multi-connected domains, or moving objects with possible topological changes (merging and splitting).

¹ In reference to the standard 5-point centered difference stencil.

2.2. Solving an elliptic interface problem with singular sources

As described before, our scheme consists of solving two generalized Helmholtz equations for the intermediate velocity \mathbf{u}^* in (2.5), and one Poisson equation for the pressure increment ϕ^{k+1} in (2.8) with a given jump condition in the normal derivative of the velocity. The details for solving Helmholtz/Poisson equations with jump conditions in the solution and its normal derivative (or the elliptic PDEs with singular sources) can be found in [11,14,12]. Here, we just give a brief sketch on how we solve such problems in an efficient and accurate way. Without loss of generality, we consider the following generalized Helmholtz equation

$$\begin{aligned} \psi_{xx} + \psi_{yy} - \lambda\psi &= f, (x, y) \in R, \\ [\psi]_{\partial\Omega} &= 0, \quad \left[\frac{\partial\psi}{\partial n} \right]_{\partial\Omega} = q. \end{aligned} \tag{2.13}$$

In our application, we have $\lambda = 2/(\mu\Delta t)$ for the prediction step, and $\lambda = 0$ for ϕ^{k+1} in (2.8). Since λ is a constant, a fast Poisson solver can be used.

The finite difference discretization using the immersed interface method can be simply written as

$$\frac{\Psi_{i+1,j} + \Psi_{i-1,j} - 2\Psi_{ij}}{h_x^2} + \frac{\Psi_{i,j+1} + \Psi_{i,j-1} - 2\Psi_{ij}}{h_y^2} - \lambda\Psi_{ij} = f(x_i, y_j) + C_{ij}, \tag{2.14}$$

where the correction term C_{ij} is zero at regular grid points where the boundary $\partial\Omega$ does not cut through the standard centered 5-point stencil. The correction term C_{ij} at those irregular grid points can be determined in a dimension by dimension fashion, see [14] for the formula of C_{ij} .

2.3. The augmented method

The procedure of an augmented approach varies with different applications and can be found in [12,13]. Here, we give a brief description for our particular problem in this paper. From time t^k to t^{k+1} , given the jump $[\partial\mathbf{u}^*/\partial\mathbf{n}] = \mathbf{q}^{k+1}$, or \mathbf{Q}^{k+1} in the discrete form, the discrete solution \mathbf{U}^* for \mathbf{u}^* satisfies a linear system

$$A\mathbf{U}^* + B\mathbf{Q}^{k+1} = \mathbf{F}_1, \tag{2.15}$$

where the matrix A corresponds to the prediction step (2.5)–(2.7), and the vector $B\mathbf{Q}^{k+1}$ corresponds to the correction term C_{ij} in (2.14) due to the jump condition $[\frac{\partial\mathbf{u}^*}{\partial\mathbf{n}}]_{\partial\Omega} = \mathbf{q}^{k+1}$. The boundary condition (2.11) (or the augmented equation) $\mathbf{u}^*|_{\partial\Omega} = \mathbf{u}_{\partial\Omega}^{k+1}$ is discretized via a least squares interpolation, and can be written in the discrete form

$$C\mathbf{U}^* + D\mathbf{Q}^{k+1} = \mathbf{U}_{\partial\Omega}^{k+1}. \tag{2.16}$$

If we put those two matrix–vector Eqs. (2.15) and (2.16) together, we get

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \mathbf{U}^* \\ \mathbf{Q}^{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_1 \\ \mathbf{U}_{\partial\Omega}^{k+1} \end{bmatrix}. \tag{2.17}$$

In practice, we do not need to form the matrices $A, B, C,$ and D . Note that the dimension of \mathbf{Q}^{k+1} is $O(2N)$ (assuming $M \sim N$), which is much smaller than that of \mathbf{U}^* ($O(2N^2)$).

Eliminating \mathbf{U}^* from Eq. (2.17), we get the Schur-complement system for \mathbf{Q}^{k+1} ;

$$(D - CA^{-1}B)\mathbf{Q}^{k+1} = \mathbf{U}_{\partial\Omega}^{k+1} - CA^{-1}\mathbf{F}_1 \stackrel{\text{def}}{=} \mathbf{F}_2, \quad \text{or} \quad E\mathbf{Q}^{k+1} = \mathbf{F}_2. \tag{2.18}$$

Note that E is not symmetric in general. We can either use a direct method or an iterative method to solve the linear system (2.18) for \mathbf{Q}^{k+1} depending on different situations. If the boundary $\partial\Omega$ is stationary and the time step is fixed, it is more efficient to use a direct method (say, Gaussian elimination with partial pivoting) since the coefficient matrix E is a constant matrix. Thus, we can form the coefficient matrix E explicitly and apply the LU decomposition to it directly. This is also advantageous for long time runs since the costly LU decomposition needs to be done just one time in the beginning. For a moving boundary problem, the matrix E is not a constant so it is more efficient to use an iterative method. In this case, it is recommended to use GMRES iteration.

Three Helmholtz/Poisson solvers are needed to evaluate $E\mathbf{Q}^{k+1}$ or $A^{-1}\mathbf{F}_1$. First, the prediction step for \mathbf{u}^* requires a fast Helmholtz solver on a rectangular domain for each velocity component. Secondly, the projection step requires a fast Poisson solver for ϕ^{k+1} given \mathbf{u}^* . For stationary boundary $\partial\Omega$ and fixed time step Δt , all the matrices are constant matrices that depend on the fast Helmholtz/Poisson solver, and the interpolation scheme for the boundary condition.

Since we do not form the matrices $A, B, C,$ and D explicitly, one question is how to use an iterative or direct method. This has been explained in detail in our recent work [12,13]. First, we set $\mathbf{Q}^{k+1} = 0$ and then solve the Navier–Stokes equations. The residual of the linear system (2.18) (or the difference between the exact and the computed boundary condition), is actually the right hand side of the Schur complement with an opposite sign. Next, we explain how to find the matrix–vector multiplication given \mathbf{Q} , a guess of \mathbf{Q}^{k+1} . This again involves only two steps: (1) solving (2.15) by given \mathbf{Q} , to get $\mathbf{U}^{k+1,*}(\mathbf{Q})$, where

we use the superscript $(k + 1, *)$ to indicate that the solution is not the final solution at time t^{k+1} yet; (2) interpolating $\mathbf{U}^{k+1,*}(\mathbf{Q})$ at $\partial\Omega$ via the least squares interpolation. Once we know the matrix–vector multiplication, we can apply the GMRES or other iterative method easily.

Also, we can form the coefficient matrix E of the Schur-complement by setting $\mathbf{Q}^{k+1,*} = e_l$, the l th unity vector, $l = 1, 2, \dots$. For a stationary interface and a fixed time step, this is needed just initially.

The idea of the augmented method has been explained in [11,12] for elliptic interface problems with piecewise constant coefficient, and Poisson equations on irregular domains. The Fortran code is available to the public either by request or by anonymous ftp.

3. Numerical examples

In this section, we present some numerical results for solving the velocity and pressure in (1.1)–(1.4) up to some fixed time T . All the computations were performed at the North Carolina State University using notebook or desktop computers. Most simulations are done within minutes to a couple of hours depending on the mesh, the geometric properties of $\partial\Omega$, and the final time T .

Example 1. Validation of the method against an exact solution

We first consider an example with the exact solutions as

$$u(x, y, t) = \begin{cases} \sin(t)(\frac{y}{r} - 2y) & \text{if } r \geq 1/2 \\ 0, & \text{otherwise,} \end{cases} \tag{3.19}$$

$$v(x, y, t) = \begin{cases} \sin(t)(-\frac{x}{r} + 2x) & \text{if } r \geq 1/2 \\ 0, & \text{otherwise,} \end{cases} \tag{3.20}$$

$$p(x, y, t) = \begin{cases} \cos(\pi x) \cos(\pi y) & \text{if } r > 1/2 \\ 0, & \text{otherwise,} \end{cases} \tag{3.21}$$

where $r = \sqrt{x^2 + y^2}$. The obstacle is a circular disk whose boundary is a circle $r = 1/2$. The solution domain is $R = [-2, 2] \times [-1, 1]$. The source term \mathbf{G} is derived directly from the exact solution.

In Table 1, we show the grid refinement analysis for different viscosity and boundary conditions. Since we are interested in the computed solutions in the domain $R \setminus \Omega$, we set

$$\|E_u\|_\infty = \max_{x_i^2 + y_j^2 \geq 1/4} \{|U_{ij} - u(x_i, y_j, T)|\} + \max_{x_i^2 + y_j^2 \geq 1/4} \{|V_{ij} - v(x_i, y_j, T)|\}$$

to be the error in the velocity at time T , and

$$\|E_p\|_\infty = \max_{x_i^2 + y_j^2 \geq 1/4} \{|P_{ij} - p(x_i, y_j, T)|\},$$

to be the error in the pressure. The numbers r_1 and r_2 are the approximated order of accuracy from the two consecutive errors for the velocity and pressure, respectively. The number $cond_1$ is the condition number of the coefficient matrix for the unknown jump in the normal derivative of the velocity proposed in this paper, while $cond_2$ is the condition number of the coefficient matrix that we set the normal and tangential force strengths as the unknowns. One can easily see that our proposed method has much better conditioned system than those that set the normal and tangential force strengths as the unknowns.

In Table 1(a) and (b), the velocity are prescribed along all four sides of the rectangular domain and $\partial\Omega$ with the final time $T = 5$. In Table 1(a), the viscosity is $\mu = 0.5$ which corresponds to a small Reynolds number while in Table 1(b), the viscosity is $\mu = 0.005$ which corresponds to a larger Reynolds number. In Table 1(c), we use the Dirichlet boundary condition from the exact solution for u and $x = a$, for v on $x = a, y = c$, and $y = d$. We use the Neumann boundary condition from the exact solution for u on $x = c, y = c$, and $y = d$, for v on $x = b$. The viscosity is $\mu = 0.005$ and the final time is $T = 5$. The purpose of this simulation is to mimic the set-up for the next example, flow past a circular cylinder.

In all cases, we clearly obtain second order accuracy for the velocity. The accuracy of the pressure is between first and second order. Note that, the pressure correction scheme proposed in [2] does not seem to improve the accuracy of the pressure because the boundary $\partial\Omega$ is arbitrary.

Example 2. Flow past a circular cylinder

We now present our numerical simulations for a classical benchmark problem of simulating the flow around a circular cylinder and compare our results with those published in the literature. For small Reynolds number ($Re < 47$), the flow structure remains symmetric with stationary recirculating vortices behind the cylinder. As the Reynolds number increases, the symmetry breaks down and the vortex starts to shed up and down alternately.

The problem set-up is similar to those in the literature [21]. The infinite domain is truncated to a rectangular domain $[a, b] \times [c, d]$ that contains a circular cylinder centered at (x_0, y_0) . The far-field boundary condition from the left is given by $u = U_\infty, v = 0$ for $t > 0$. We have tested a number of parameters and set-ups. Unless stated otherwise, the plots shown in this section for flow past a circular cylinder are all calculated using the finest mesh 900×450 in the domain $[-10, 10] \times [-5, 5]$

with the cylinder center $(-5, 0)$ to minimize the effect from the approximated boundary conditions. We also fix $U_\infty = 1$ and the radius of the cylinder $r_0 = 0.5$, but vary the Reynolds number $Re = 2r_0U_\infty/\mu = 1/\mu$. All of the results agree with those in the literature very well, particularly with the latest second order method using vorticity–streamfunction formulation [3]. We run our tests to a final time $T = 60$ or longer. In the GMRES approach (without forming the coefficient matrix), the time step is chosen as

$$(\Delta t)^k = \min \left\{ \frac{h}{2}, \frac{h}{2 \max\{|U_{ij}^k| + |V_{ij}^k|\}} \right\}, \quad h = \min\{h_x, h_y\}, \quad (3.22)$$

where U_{ij}^k and V_{ij}^k are the computed velocity at a grid point (x_i, y_j) at time level t^k . For long time computations, it is faster to form the coefficient matrix if Δt is fixed. We take the CFL condition as 0.5 for $Re \leq 100$, and 1/6 for $Re = 150, 200$. We show our result at the final time $T = 60$ or longer. Most of computations were done within a few minutes to a couple of hours depending on the mesh size. For example, for $Re = 100$ and the mesh size 900×450 , the entire simulation took about 132 min.

In Fig. 2, we show the plots of the vorticity contours. The plots show very good agreement with the results in [3] where the streamfunction–vorticity formulation is used. The contours in the wake of cylinder for $Re = 20$ has slightly wider opening than that of $Re = 40$. For $Re = 80$, the vortex shedding behavior has been fully developed.

In Fig. 3, we show a grid convergence analysis of the drag coefficient between $0 \leq t \leq 60$ for the case $Re = 40$. The dash-dotted line is the result obtained from the coarse grid 226×113 ; the dashed line is from a finer grid 450×225 ; and the solid line is the result from the finest grid 900×450 . One can see that as the grid refines, the drag coefficient approaches to $C_D = 1.6622$ which is also listed in Table 2.

In Fig. 4, we show the plots of a few selected streamlines for $Re = 20$ and $Re = 40$. The flow remains perfectly symmetric. The size and the length of the stationary recirculating vortices behind the cylinder are also in a very good agreement with known results in the literature.

In Fig. 5, we show the vorticity contour plots for different Reynolds numbers $Re = 100, 150, 200$. The characteristic vortex shedding has been fully developed for all the cases.

In Tables 2 and 3, we list the drag and lift coefficients, Strouhal numbers, and compare with some of recent results in the literature. The drag and lift coefficients are computed according to the following formulas given in [3]:

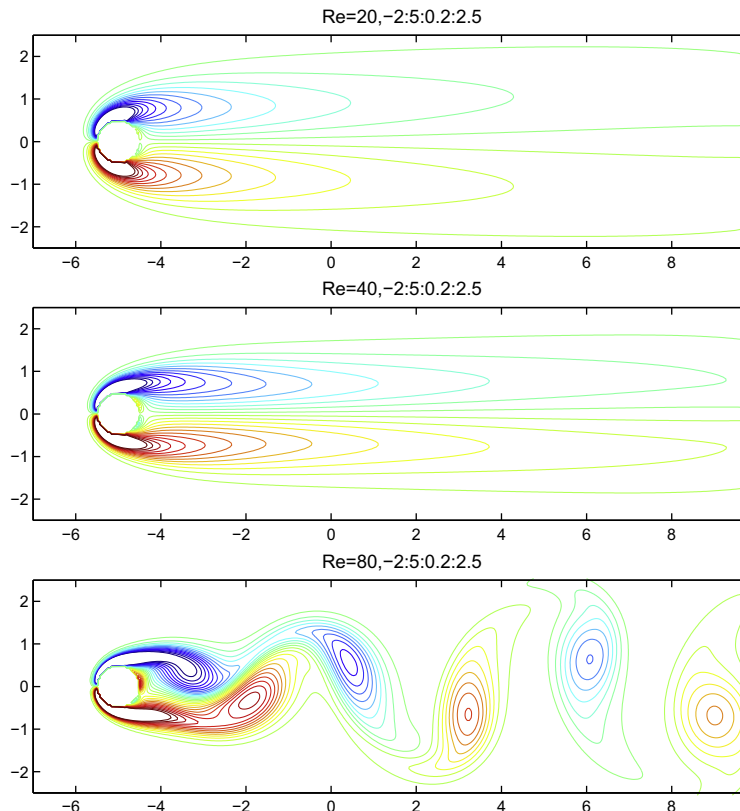


Fig. 2. Vorticity plots $(-2.5:0.2:2.5)$ for $Re = 20, 40, 80$ at time $T = 60$.

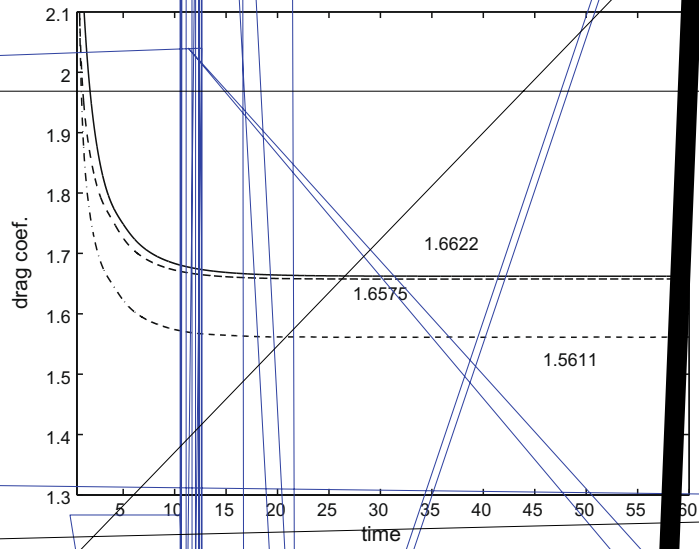
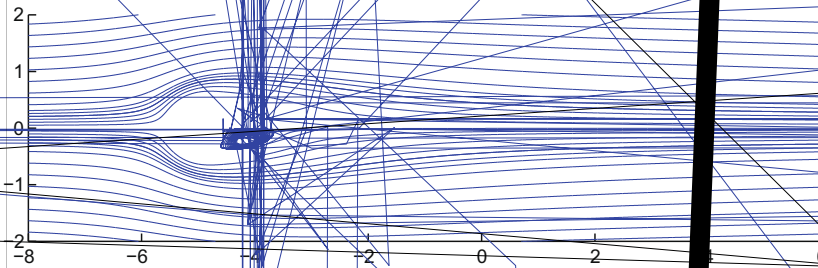
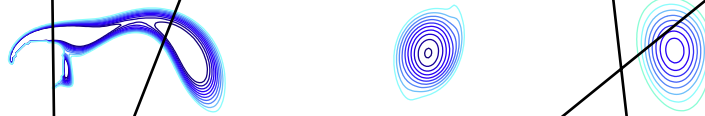
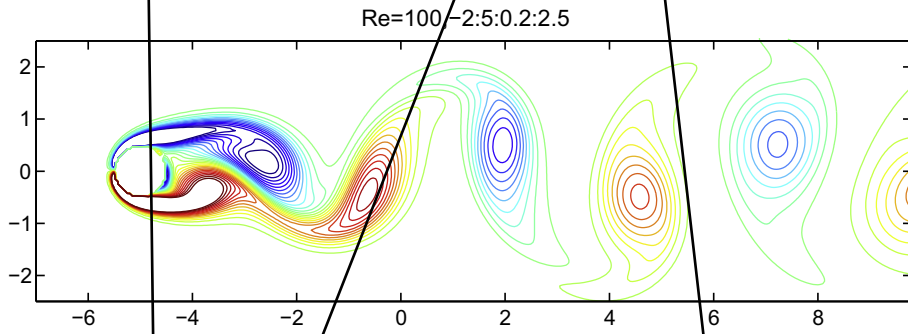


Fig. 3. A grid refinement analysis for the drag coefficient between $0 \leq t \leq 60$ for the case $Re = 40$. The solid line is the result from the finest grid; the dash-dotted line is the result from the coarsest grid.

Table 2
The comparison of drag coefficients for different Reynolds numbers.

Re	Present	Su et al. [21]	Calhoun [3]	Xu et al. [22]	Russell et al. [20]	Le et al. [10]
20	2.2069	2.20	2.19	2.23	2.1	2.05
40	1.6622	1.63	1.62	1.48	1.6	1.56
80	1.4503	1.43	-	-	-	-
100	1.4027	1.40	1.330	1.423	1.4	1.37
150	1.3494	1.39	-	-	-	-
200	1.2722	-	1.172	1.42	1.3	1.34





$$C_D = \frac{\rho}{U_\infty^2} \int_0^{2\pi} \left(-\omega(\theta) + r \frac{\partial \omega(\theta)}{\partial n} \right) \sin(\theta) d\theta, \quad (3.23)$$

$$C_L = -\frac{\rho}{U_\infty^2} \int_0^{2\pi} \left(-\omega(\theta) + r \frac{\partial \omega(\theta)}{\partial n} \right) \cos(\theta) d\theta, \quad (3.24)$$

where r is the radius of the circular obstacle, and θ is the angle between the outward-directed normal to $\partial\Omega$ and the x -axis. Here, the vorticity ω and its normal derivative $\frac{\partial \omega(\theta)}{\partial n}$ can be computed using the velocity as $\omega = v_x - u_y$ and

$$\frac{\partial \omega(\theta)}{\partial n} = \frac{\partial \omega}{\partial x} \cos(\theta) + \frac{\partial \omega}{\partial y} \sin(\theta) = (v_{xx} - u_{xy}) \cos(\theta) + (v_{xy} - u_{yy}) \sin(\theta) = (v_{xx} + v_{yy}) \cos(\theta) - (u_{xx} + u_{yy}) \sin(\theta).$$

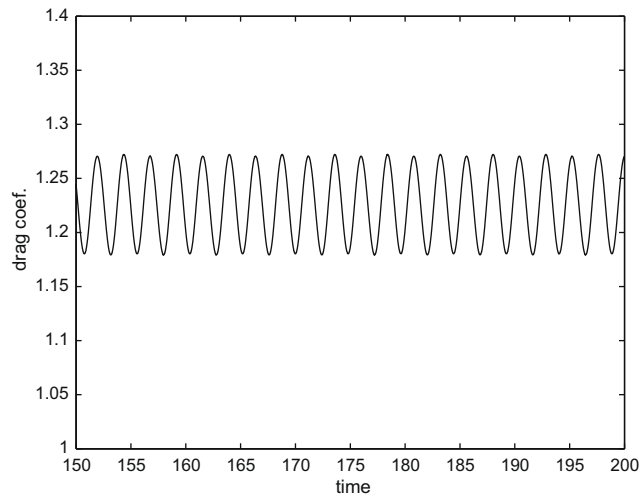


Fig. 6. Plot of the drag coefficient as the function time for $Re = 200$ between $150 \leq t \leq 200$.

Note that, since the vorticity and its normal derivative is needed only on the interface, we use a second order one-sided least squares interpolation, for instance,

$$u_x(X^*, Y^*) = \sum_{(x_i, y_j) \in R \setminus \Omega}^{ns} \gamma_{ij} U_{ij},$$

$$u_{xx}(X^*, Y^*) = \sum_{(x_i, y_j) \in R \setminus \Omega}^{ns} \tilde{\gamma}_{ij} U_{ij},$$

to approximate those partial derivatives at a number of equally spaced points (X^*, Y^*) on $\partial\Omega$. The number of grid points (in the neighborhood of (X^*, Y^*)) n_s is taken to be $9 \sim 12$ but only requires a second order accurate scheme (that is, matching up to second partial derivatives). The linear system of equations for the interpolation coefficients γ_{ij} and $\tilde{\gamma}_{ij}$ is under-determined but is solvable. We use the singular value decomposition (SVD) to solve for the coefficients. Using this approach, the computed partial derivatives have almost the same order of accuracy as the interpolation function itself.

We can see good agreements of our results in the drag coefficients. The lift coefficient also agrees with those in the literature for $Re = 100$. It is a little under-estimated for $Re = 200$, more so as in the results presented in [10,20]. One of possible explanations may be that the projection method works well for small Reynolds numbers, but not for large ones, or/and that the domain is not large enough. We have observed that the position of the cylinder has some effect on C_L when $Re = 200$. In Fig. 6, we plot the time evolution of the drag coefficient for the case $Re = 200$.

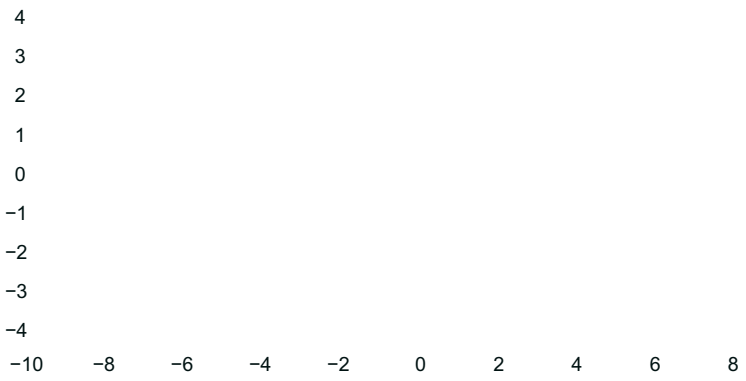
In Fig. 7(a)–(c), we show the vorticity plots for the flow past different dumbbell-shaped objects. In (a) and (b), we have the same geometry but different μ , $\mu = 0.05$ for (a) and $\mu = 0.01$ for (b). In (c), we have the same μ as in (b) but with a different geometry. It is clear that no vortex shedding is observed in this flow. In Fig. 7(d), we show the vorticity plot of the flow around two circular cylinders located at different positions in the domain with $\mu = 0.01$. The radius of the front cylinder is $r = 0.5$, while the second one is $r = 0.6$. The flow behind the second cylinder behaves similarly as the case where only one cylinder exists if those two cylinders are far apart enough.

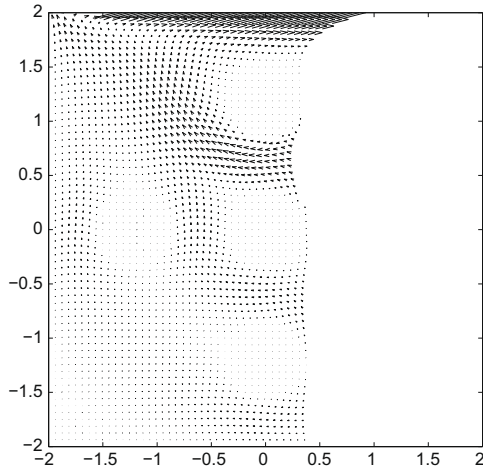
We have also tested a moving boundary example for the flow past an in-line oscillating cylinder as described in [21]. The prescribed velocity of the cylinder is $u = 0$, $v = 0.14 \sin(2\pi f_c t)$, where the cylinder oscillating frequency f_c is about two times of the natural vortex shedding frequency f_q . The average drag coefficient is $C_D = 1.693$; and the lift coefficient is $C_L = 1.010$, which are very close to the results given in [21].

Example 3. A lid-driven cavity flow with multiple obstacles

We show our numerical experiments for a lid-driven cavity flow with multiple obstacles and different geometries. Similar numerical results can be found in the literature [21]. Since the focus of this paper is on the new numerical method, we thus present numerical results for a few problems without too much elaborations about their physical meanings.

In Fig. 8, we show several plots of numerical simulations. The flow is driven by the horizontal velocity along the boundary $y = d$ where $u = 1$ and $v = 0$. No-slip boundary conditions are prescribed along other sides of the boundary. The mesh size is 256×256 . In Fig. 8(a), we show the steady velocity field of the flow with five circular cylinders. We obtain qualitatively similar flow behavior as shown in [21] for $\mu = 0.01$. Fig. 8(b) and (c) shows the steady velocity field of the flow with five rose-shaped obstacles at different locations. Fig. 8(d) shows the vorticity plot of the flow corresponding to the one in Fig. 8(c).





4. Conclusions and acknowledgment

In this paper, we propose a new augmented method for solving Navier–Stokes equations in irregular domains. This new approach significantly improves the condition number of the linear system of equations for the augmented variable compared with some other methods. The efficiency of the new method has been illustrated by running several interesting examples. The method is very efficient for fixed obstacles and long time simulations, since we can form the coefficient matrix in the beginning and perform the matrix decomposition outside of the time loop. For a moving boundary, or a short time simulation with a fixed boundary, the GMRES iteration would be more efficient.

The author would like to acknowledge the following supports. The first and third authors are partially supported by US-ARO Grant 49308-MA, and US-AFSOR Grant FA9550-06-1-0241. The third author is also partially supported by US-NSF Grant DMS0412654, and USA NSF-NIH Grant #0201094. The second author is partially supported by National Science Council of Taiwan under Grant NSC-95-2115-M-009-010-MY2 and MoE-ATU project.

References

- [1] J.B. Bell, P. Colella, H.M. Glaz, A second-order projection method for the incompressible Navier–Stokes equations, *J. Comput. Phys.* 85 (1989) 257–283.
- [2] D.L. Brown, R. Cortez, M.L. Minion, Accurate projection methods for the incompressible Navier–Stokes equations, *J. Comput. Phys.* 168 (2001) 464.
- [3] D. Calhoun, A Cartesian grid method for solving the streamfunction–vorticity equation in irregular regions, *J. Comput. Phys.* 176 (2002) 231–275.
- [4] E. Weinan, J. Liu, Projection method: I. Convergence and numerical boundary-layers, *SIAM J. Numer. Anal.* 32 (1995) 1017–1057.
- [5] E. Guendelman, A. Selle, F. Losasso, R. Fedkiw, Coupling water and smoke to thin deformable and rigid shells, *SIGGRAPH, ACM TOG* 24 (2005) 973–981.
- [6] J. Hunter, Z. Li, H. Zhao, Autophobic spreading of drops, *J. Comput. Phys.* 183 (2002) 335–366.
- [7] J. Van Kan, A second-order accurate pressure-correction scheme for viscous incompressible flow, *SIAM J. Sci. Comput.* 7 (1986) 870–891.
- [8] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier–Stokes equations, *J. Comput. Phys.* 59 (1985) 308–323.

- [9] M.-C. Lai, C.S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, *J. Comput. Phys.* 160 (2000) 705–719.
- [10] D.V. Le, B.C. Khoo, J. Peraire, An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries, *J. Comput. Phys.* 220 (2006) 109–138.
- [11] Z. Li, A fast iterative algorithm for elliptic interface problems, *SIAM J. Numer. Anal.* 35 (1998) 230–254.
- [12] Z. Li, K. Ito, The immersed interface method – numerical solutions of PDEs involving interfaces and irregular domains, *SIAM Front. Ser. Appl. Math.* (2006). FR33.
- [13] Z. Li, K. Ito, M.-C. Lai, An augmented approach for Stokes equations with a discontinuous viscosity and singular forces, *Comput. Fluids* 36 (2007) 622–635.
- [14] Z. Li, M.-C. Lai, The immersed interface method for the Navier–Stokes equations with singular forces, *J. Comput. Phys.* 171 (2001) 822–842.
- [15] Z. Li, H. Zhao, H. Gao, A numerical study of electro-migration voiding by evolving level set functions on a fixed cartesian grid, *J. Comput. Phys.* 152 (1999) 281–304.
- [16] M.N. Linnick, H.F. Fasel, A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains, *J. Comput. Phys.* 204 (2005) 157–192.
- [17] F. Losasso, G. Irving, E. Guendelman, R. Fedkiw, Melting and burning solids into liquids and gases, *IEEE TVCG* 12 (2006) 343–352.
- [18] S. Marella, S. Krishnan, H. Liu, H.S. Udaykumar, Sharp interface Cartesian grid method I: an easily implemented technique for 3D moving boundary computations, *J. Comput. Phys.* 210 (2005) 1–31.
- [19] R. Mittal, G. Iaccarino, Immersed boundary methods, *Ann. Rev. Fluid Mech.* 37 (2005) 239–261.
- [20] D. Russell, Z.J. Wang, A Cartesian grid method for modeling multiple moving irregular objects in 2D incompressible viscous flow, *J. Comput. Phys.* 191 (2003) 177–205.
- [21] S.-W. Su, M.-C. Lai, C.-A. Lin, An immersed boundary technique for simulating complex flows with rigid boundary, *Comput. Fluids* 36 (2007) 313–324.
- [22] S. Xu, Z.J. Wang, An immersed interface method for simulating the interaction of a fluid with moving boundaries, *J. Comput. Phys.* 216 (2006) 454–493.
- [23] T. Ye, R. Mittal, H.S. Udaykumar, W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, *J. Comput. Phys.* 156 (1999) 209–240.