Computers & Fluids 39 (2010) 1033–1040

Contents lists available at ScienceDirect

Computers & Fluids

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

An augmented method for free boundary problems with moving contact lines

Zhilin Li [a,*], Ming-Chih Lai [b], Guowei He [c], Hongkai Zhao [d]

[a] Center for Research in Scientific Computation & Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205, United States
[b] Center of Mathematical Modeling and Scientific Computing & Department of Applied Mathematics, National Chiao Tung University, 1001, Ta Hsueh Road, Hsinchu 30050, Taiwan
[c] LNM, Institute of Mechanics, Chinese Academy of Sciences, Beijing 10080, China
[d] Department of Mathematics, University of California, Irvine, CA 92697, United States

ARTICLE INFO

Article history:
Received 11 April 2009
Received in revised form 10 January 2010
Accepted 18 January 2010
Available online 25 January 2010

Keywords:
Moving contact line
Free boundary problem
Triple junction
One-phase flow
Navier-Stokes equations
Embedding technique
Immersed interface method
Irregular domain
Augmented method

ABSTRACT

An augmented immersed interface method (IIM) is proposed for simulating one-phase moving contact line problems in which a liquid drop spreads or recoils on a solid substrate. While the present two-dimensional mathematical model is a free boundary problem, in our new numerical method, the fluid domain enclosed by the free boundary is embedded into a rectangular one so that the problem can be solved by a regular Cartesian grid method. We introduce an augmented variable along the free boundary so that the stress balancing boundary condition is satisfied. A hybrid time discretization is used in the projection method for better stability. The resultant Helmholtz/Poisson equations with interfaces then are solved by the IIM in an efficient way. Several numerical tests including an accuracy check, and the spreading and recoiling processes of a liquid drop are presented in detail.

1. Introduction

In this paper, we propose an augmented immersed interface method for free boundary problems with moving contact lines in two space dimensions. This problem arises from the spreading of liquid drop on a solid substrate. We wish to investigate the wetting effects of the drop on the solid surface [9]. The contact line is defined as the intersection of the fluid interface with the solid surface. When the contact line is in its equilibrium state (not moving), the contact angle, the angle between the fluid interface and the solid boundary, is called the static contact angle that depends on the surface tension coefficients between the different phases (Laplace–Young relation). In the presence of moving contact lines, it is well-known that the no-slip boundary condition at the moving contact line will lead to a non-integrable force singularity [12,6]. Several different techniques including continuum hydrodynamical or molecular dynamical treatments have been proposed to remove such singularity, see for example, a recent review article by Qian et al. [25]. In this paper, our intention is not to provide a new mathematical model or analysis, but is to develop an efficient numerical method for the mathematical model developed by Ren and Weinan [26] for the moving contact line problem.

The one-phase free boundary problem is a simplified model for a two-phase (liquid–gas) problem with neglecting the influence of one phase (gas), see Fig. 1 for an illustration. In this paper, we particularly consider a liquid drop on a solid surface and study its wetting behavior. As usual, the governing equations are described by the incompressible Navier-Stokes equations in a time dependent single fluid domain $\Omega(t)$, together with the stress force balancing conditions on the free surface $\Gamma_F(t)$, and the slip boundary conditions on the solid surface $\Gamma_S(t)$, that is,

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t}+(\mathbf{u}\cdot\nabla)\mathbf{u}\right)+\nabla p=\mu\Delta\mathbf{u}+\mathbf{G}, \qquad \text{in } \Omega(t), \tag{1.1}$$

$$\nabla\cdot\mathbf{u}=0, \qquad \text{in } \Omega(t), \tag{1.2}$$

$$-p+\mathbf{n}^T\cdot\mu(\nabla\mathbf{u}+\nabla\mathbf{u}^T)\cdot\mathbf{n}=\gamma\kappa-p_{air}, \qquad \text{on } \Gamma_F(t), \tag{1.3}$$

$$\boldsymbol{\tau}^T\cdot\mu(\nabla\mathbf{u}+\nabla\mathbf{u}^T)\cdot\mathbf{n}=0, \qquad \text{on } \Gamma_F(t), \tag{1.4}$$

$$v=0, \quad \beta u=\mu\frac{\partial u}{\partial y}, \qquad \text{on } \Gamma_S(t). \tag{1.5}$$

Here, $\mathbf{u}=(u,v)$ is the fluid velocity, p the pressure, and $\mathbf{G}$ is the external force that can include the gravity. The constants ρ and μ are the fluid density and viscosity, respectively.

* Corresponding author.
E-mail address: zhilin@unity.ncsu.edu (Z. Li).

0045-7930/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compfluid.2010.01.013

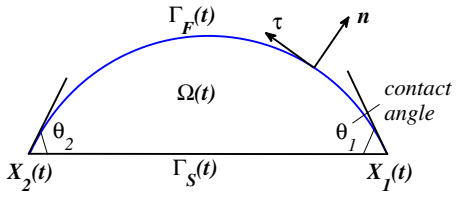

Fig. 1. A diagram of a one-phase flow with two moving contact lines $X_1(t)$ and $X_2(t)$, and two contact angles $\theta_1(t)$ and $\theta_2(t)$. The circular portion represents the free boundary $\Gamma_F(t)$ where the stress force balancing conditions are imposed. The bottom line represents the solid boundary $\Gamma_S(t)$ where the Navier slip boundary conditions are imposed.

Eqs. (1.3) and (1.4) are the stress force balancing conditions on the free boundary $\Gamma_F(t)$, where γ is the surface tension, κ the curvature of the free boundary (negative sign in Fig. 1), p_{air} the pressure outside of the drop, and $\mathbf{n}$ and τ are the unit normal and tangent vectors at the free boundary. The derivation for above stress balance conditions can be found, for example, in [14]. In additional to the stress force conditions, we should impose the kinematic condition on the free boundary, that is, the normal velocity of the fluid at the free boundary should be equal to the normal velocity of the interface. In this paper, we shall use the level set method [21] to evolve the motion of the free boundary so that the kinematic boundary condition is imposed naturally. The detail for the free boundary representation by level set should be discussed in later section.

Along the fluid–solid boundary $\Gamma_S(t)$, as mentioned before, the no-slip boundary condition at the moving contact lines leads to a non-integrable force singularity. Therefore, we allow the solid boundary to be partial slip and impose the Navier slip boundary condition, Eq. (1.5). The first condition in (1.5) is the no penetration boundary condition. The second condition is the slip with friction indicating that the tangential velocity of the fluid is proportional to their tangential stress. Here, as in [26], β is the friction coefficient. Both conditions in Eq. (1.5) can also be easily derived from their general expression as shown in [8]. Notice that, in other literature such as in [8,23], one can define $\lambda = \mu/\beta$ as the slip length indicating the fictitious distance to the solid surface where the fluid tangential velocity is extrapolated to be zero.

In additional to imposing the Navier slip boundary condition on the solid wall to remove force singularity, the contact line dynamics must also be prescribed. In the presence of contact angle hysteresis, one can prescribe the contact angle depending on the sign of contact line speed [31,22,8,32]. In the above models, however, the advancing and receding contact angles must be given. Recently, Ren and Weinan derived an effective boundary condition at the contact line from the force balance argument [26]. As mentioned by the authors, the main driving force for the slip is the unbalanced Young's force which results from the deviation of the contact angle from its static value. Therefore, in this work, we simply apply their effective condition to the contact line directly. That is, at the moving contact lines $x_1(t)$ and $x_2(t)$, we have

$$\alpha u = \gamma(\cos\theta^* - \cos\theta), \tag{1.6}$$

where α is the effective friction coefficient, θ^* the static (equilibrium) contact angle that depends on the surface tensions of the free and solid boundaries, and θ is the dynamic contact angle between the moving free boundary and the solid boundary as shown in Fig. 1.

While a two-phase model, in which the density and viscosity are discontinuous across the interface, is more practical, a direct discretization of the Navier-Stokes equations may be ill-conditioned and difficult to solve. The condition number of the discrete system is proportional to the jumps of the physical parameters. Under certain circumstances, a one-phase model is adequate and can be solved more efficiently than the two-phase model. More discussions about the circumstances will be discussed later.

The solvability and stability of the problem under certain circumstances are discussed in [10,24,30,29] including simplified one-dimensional numerical simulations. Most numerical simulations are for the two-phase model. In [17], a volume of fluid (VOF) method was developed. In [8], an Arbitrary Lagrangian Eulerian (ALE) finite element method using a body fitted mesh was studied. The Immersed Boundary (IB) method have been proposed in [11,26], in which the boundary condition at the contact lines are enforced through an external point force at those lines. These point-sources are then distributed to the nearby grid points via a discrete delta function. In [13], a simplified model using the Laplace equation to obtain the contact angle is studied and a level set method was developed for the numerical simulations.

In this paper, we propose an augmented immersed interface method for the one-phase free boundary problem based on Cartesian grids. Thus, there is almost no cost in the grid generation even though the free boundary is moving. We first embed the domain into a rectangular one with the solid surface being the bottom side so that we can solve the problem on a rectangular domain using a Cartesian grid. We then use a modified version of the projection method [3,15,16] to solve the Navier-Stokes equations for the motion of the fluid. They key step is to use the jump in the normal derivative of the velocity as an augmented variable so that we can solve the fluid equations efficiently. We will explain our numerical method in detail in the next section; followed by results of numerical experiments and analysis. We draw some conclusions in the last section.

2. The numerical method

Our numerical method for the fluid equations 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, [3,15,16] and many others. The projection method that we used in our scheme is the one described in [5] which is based on the pressure increment formulation of [3,15].

As mentioned before, instead of solving the fluid equations in an irregular time dependent domain $\Omega(t)$, we embed the fluid domain of the droplet into a rectangular computational domain $R = [a, b] \times [c, d]$ which is sufficiently large to enclose the time dependent droplet with the bottom side coinciding with the solid surface. Since the computational domain is a regular rectangle instead of an irregular one bounded by the free boundary, the governing equations now can be solved in a Cartesian grid. The apparent advantage of using Cartesian grid is that there are several fast direct solvers for fluid equations available in literature. Here, the spatial spacings are chosen as $h_x = (b-a)/M$ and $h_y = (d-c)/N$, where M and N are the number of grid points used in the x and y directions, respectively. For simplicity, we use a standard uniform mesh such that $h = h_x = h_y$.

As a common practice, from one time level to the next, we use a splitting approach for the free boundary problem. In this approach, we first fix the moving boundary (denoted by Γ_F^k below) and solve the fluid equations on the irregular domain to get the velocity. We then use the computed velocity to evolve the free boundary using the level set method. We describe the numerical scheme from time t^k to t^{k+1} below. At time step t^k, given an approximation of the velocity $\mathbf{u}^k$, the pressure p^k, the free boundary position Γ_F^k, and an initial guess of augmented variable $\mathbf{q}^{k+1}$, we carry out the following steps:

Step 1: Prediction step

$$\rho\frac{\mathbf{u}^*-\mathbf{u}^k}{\Delta t}=\begin{cases}-\nabla p^k-\rho(\mathbf{u}\cdot\nabla\mathbf{u})^{k+\frac{1}{2}}+\dfrac{\mu}{2}(\Delta\mathbf{u}^*+\Delta\mathbf{u}^k)+\mathbf{G}^{k+\frac{1}{2}}, & \mathbf{x}\in(\Omega\cap Z^R)\\ -\nabla p^k-\rho(\mathbf{u}\cdot\nabla\mathbf{u})^k+\mu\Delta\mathbf{u}^*+\mathbf{G}^k, & \mathbf{x}\in(\Omega\cap Z^I)\\ \dfrac{\mu}{2}(\Delta\mathbf{u}^*+\Delta\mathbf{u}^k), & \mathbf{x}\in(\Omega^c\cap Z^R)\\ \mu\Delta\mathbf{u}^*, & \mathbf{x}\in(\Omega^c\cap Z^I)\end{cases}\tag{2.7}$$

$$[\mathbf{u}^*]_{\Gamma_F^k}=0,\quad \left[\frac{\partial\mathbf{u}^*}{\partial n}\right]_{\Gamma_F^k}=\mathbf{q}^{k+1},\tag{2.8}$$

$$\frac{\partial\mathbf{u}^*}{\partial n}=0\ \text{ at } x=a,x=b,\text{ and } y=d,\ v^*=0 \text{ and } \beta u^*=\mu\frac{\partial u^*}{\partial y}\text{ at } y=c.\tag{2.9}$$

Step 2: Projection step

$$\begin{cases}\Delta\phi^{k+1}=\dfrac{\nabla\cdot\mathbf{u}^*}{\Delta t}, & \mathbf{x}\in R,\\ [\phi^{k+1}]_{\Gamma_F^k}=0, \quad \left[\dfrac{\partial\phi^{k+1}}{\partial n}\right]_{\Gamma_F^k}=0, & \left.\dfrac{\partial\phi^{k+1}}{\partial\mathbf{n}}\right|_{\partial R}=0,\end{cases}\tag{2.10}$$

$$\mathbf{u}^{k+1}=\mathbf{u}^*-\Delta t\nabla\phi^{k+1},\qquad \mathbf{x}\in R,\tag{2.11}$$

$$\nabla p^{k+1}=\nabla p^k+\nabla\phi^{k+1},\qquad \mathbf{x}\in\Omega\tag{2.12}$$

Step 3: Computing the residual of the boundary condition (1.3) and (1.4).

$$E_1^{k+1}=-p^{k+1}+\mathbf{n}^T\cdot\mu(\nabla\mathbf{u}^{k+1}+\nabla(\mathbf{u}^{k+1})^T)\cdot\mathbf{n}-(\gamma\kappa-p_{air}),\ \text{ on } \Gamma_F^k\tag{2.13}$$

$$E_2^{k+1}=\boldsymbol{\tau}^T\cdot\mu(\nabla\mathbf{u}^{k+1}+\nabla(\mathbf{u}^{k+1})^T)\cdot\mathbf{n}=0,\quad \text{on}\quad \Gamma_F^k.\tag{2.14}$$

The first two steps consist of the traditional steps of the projection method for the Navier-Stokes solver. The augmented variable $\mathbf{q}^{k+1}$ defined only along the free boundary Γ_F^k is introduced in the prediction step and should be determined suitably such that the free boundary conditions (1.3) and (1.4) are satisfied at time level t^{k+1}. If the boundary conditions are satisfied, that is, $E_1^{k+1}=0$ and $E_2^{k+1}=0$, then we have computed the velocity $\mathbf{u}^{k+1}$, otherwise, we need to find better $\mathbf{q}^{k+1}$ until the residual $\|E_1^{k+1}\|$ and $\|E_2^{k+1}\|$ are smaller enough. This can be done iteratively, or by solving a linear system of equations. This is explained in detail below and in Section 2.3. Notice that, the normal vector $\mathbf{n}$, the tangent vector $\boldsymbol{\tau}$ and the curvature κ are all evaluated using the free boundary position at the time level t^k.

In the algorithm above, $\Omega\cap Z^R$ and $\Omega^c\cap Z^R$ are *regular* grid points inside and outside of the domain Ω, respectively, while $\Omega\cap Z^I$ and $\Omega^c\cap Z^I$ are *irregular* grid points in the corresponding domains. The classification of the *regular* or *irregular* grid points can be found in Section 2.2. The non-linear term $(\mathbf{u}\cdot\nabla\mathbf{u})^{k+\frac{1}{2}}$ is approximated by

$$(\mathbf{u}\cdot\nabla\mathbf{u})^{k+\frac{1}{2}}=\frac{3}{2}(\mathbf{u}^k\cdot\nabla)\mathbf{u}^k-\frac{1}{2}(\mathbf{u}^{k-1}\cdot\nabla)\mathbf{u}^{k-1},\tag{2.15}$$

at regular grid points. We use a first order discretization for $(\mathbf{u}\cdot\nabla\mathbf{u})^k$ at irregular grid points as well to avoid cross differentiation for the explicit terms.

The pressure term in the prediction step is treated explicitly and we are only interested in the solution in the domain Ω, so we do not need to compute ∇p outside the domain. As we can see, the resultant equation in the prediction step in Ω is an elliptic equation, so it is natural to embed the domain Ω into a computational rectangular domain R by setting another elliptic equation (heat equation here) in $\Omega^c=R-\Omega$. In such way, we can discretize the whole equations and form a linear system based on Cartesian grids.

Given the augmented variable $\mathbf{q}$, we can see that in Step 1 and 2, there involve solving elliptic interface problem with singular sources on a Cartesian domain R with different boundary conditions. The detail on how to solve those equations using immersed interface method can be found in Section 2.4. To solve the unknown $\mathbf{q}^{k+1}$, we can choose $\mathbf{q}=\mathbf{e}_i$ to get coefficient matrix for $\mathbf{q}^{k+1}$, see Section 2.3. In other words, to get the coefficient matrix, we need to run the algorithm N_b times at one time level, where N_b is the dimension of $\mathbf{Q}^{k+1}$, the discrete approximation of $\mathbf{q}^{k+1}$. Alternatively, we can use the GMRES iterative method that only requires the matrix–vector multiplication. Each matrix–vector multiplication requires to solve the Navier-Stokes equation once, and to interpolate the boundary conditions once. More details will be followed later.

In this paper, we use the backward Euler scheme near/on the free boundary while use the Crank-Nicholson scheme away from the boundary in the prediction step. The reason is for the stability consideration. The scheme is at least second order accurate in space and first order in time. When $\Delta t\sim O(h_x,h_y)$, the scheme seems to be still second order accurate since we can use one order lower scheme along the boundary, see for example, [2,4,19]. The stability reason is to avoid using the explicit Laplacian term $\Delta\mathbf{u}^k$ for the moving boundary. Even if $\mathbf{u}^k$ is second order accurate, the approximation of the explicit term $\Delta\mathbf{u}^k$ may be inaccurate near/on the boundary because the errors are often not smooth. Such a treatment is very useful if the boundary is moving. The disadvantage is that a fast Helmholtz solver based on FFT cannot be applied directly. Here, we use the structured multi-grid solver DMGD9V [7]. Note that, it is possible to use a second-order time-stepping scheme such as the BDF scheme. The difficulty is the extra bookkeeping effort needed to deal with the irregular grid points in three different time levels rather than the two levels.

Note that a fully implicit approach (backward Euler)

$$\rho\frac{\mathbf{u}^*-\mathbf{u}^k}{\Delta t}=\begin{cases}-\nabla p^k-\rho(\mathbf{u}\cdot\nabla\mathbf{u})^k+\mu\Delta\mathbf{u}^*+\mathbf{G}^{k+1}, & \mathbf{x}\in(\Omega)\\ \mu\Delta\mathbf{u}^*, & \mathbf{x}\in(R\setminus\Omega)\end{cases}\tag{2.16}$$

is simpler and may be appropriate if we are only interested in the equilibrium solution. The method is first order in time and second order in space and has better stability. Another advantage is that, fast Helmholtz/Poisson solvers based FFT can be applied. This approach has also been tested and worked well.

2.1. The implementation of the contact line condition

As discussed in [26], on the solid boundary and away from the contact line, the Navier slip boundary condition (1.5) is imposed. However, on the solid boundary but at the contact line, the effective condition Eq. (1.6) must be imposed. There are several ways to implement the moving contact line condition (1.6) numerically. The simplest approach is to enforce the condition smoothly by using

$$u(x)=\frac{\gamma}{\alpha}(\cos\theta^*-\cos\theta)e^{-(x-x_i)^2/w^2},\tag{2.17}$$

where x_i is the position of the contact line on the solid boundary, and $w\sim O(h_x)$ is the effective width of the transition region, see [26]. In summary, we have Navier slip boundary condition at almost all grid points along the solid boundary except for the two neighboring grid points next to the contact line where the condition (2.17) is imposed. Notice that, the above numerical boundary condition is implemented for u^* on the solid boundary in the projection step in which the contact angle θ and the contact line position x_i are both obtained explicitly (that is, from previous time step).

The second approach is to add an external point force $\mathbf{f}_i\delta(x-x_i)\delta(y)$ at the contact line, say $(x_i,0)$, $i=1,2$, for a pre-determined force strength $\mathbf{f}_i$ arising from the unbalanced Young's force,

then use a discrete delta function to distribute the force term to the nearby grid points of the contact lines as in the Peskin's Immersed Boundary method, see [11,26].

2.2. The level set representation and the orthogonal projections

We use the level set method [23,28] to evolve the free boundary for convenience since some of the codes used for the results in [13] can be handily applied. We believe that the front tracking method would work equally well in two space dimensions. One difficulty is to deal with the boundary condition of the free boundary at the contact lines. In the level set method, the free boundary is represented by the zero level set of a two-dimensional Lipschitz continuous function $\varphi(x,y,t)$, for example, the signed distance function. The level set function then is evolved according to the Hamilton–Jacobi equation

$$\varphi_t + \mathbf{u}\cdot\nabla\varphi = 0. \tag{2.18}$$

In discretization, the level set function is defined at grid points. One advantage of the level set representation is that we can classify grid points easily. In reference to the standard central five-point stencil, a grid point $\mathbf{x}_{ij} = (x_i, y_j)$ is *regular* if $\varphi_{ij}^{min}\varphi_{ij}^{max} > 0$, where

$$\varphi_{ij}^{max} = \max\left\{\varphi_{i-1,j}, \varphi_{ij}, \varphi_{i+1,j}, \varphi_{i,j-1}, \varphi_{i,j+1}\right\}, \tag{2.19}$$

$$\varphi_{ij}^{min} = \min\left\{\varphi_{i-1,j}, \varphi_{ij}, \varphi_{i+1,j}, \varphi_{i,j-1}, \varphi_{i,j+1}\right\}. \tag{2.20}$$

Otherwise it is called an *irregular* grid point which means that the free boundary cuts through the finite difference stencil.

In discretization, the augmented variable is defined at the orthogonal projection of the irregular grid points from the domain Ω^+, that is the outside of the fluid domain. Let be $\mathbf{x}_{ij}$ be such a grid point, the orthogonal projection can be approximated by

$$\mathbf{x}_{ij}^* = \mathbf{x}_{ij} + \alpha\nabla\varphi, \tag{2.21}$$

where α is an approximate signed distance between $\mathbf{x}_{ij}$ and the free boundary, which is obtained by solving the quadratic equation,

$$\varphi(\mathbf{x}) + (\nabla\varphi(\mathbf{x})\cdot\mathbf{p})\alpha + \frac{1}{2}(\mathbf{p}^T He(\varphi(\mathbf{x}))\mathbf{p})\alpha^2 = 0, \tag{2.22}$$

where $\mathbf{p} = \nabla\varphi$ and

$$\mathbf{p}^T He(\varphi)\mathbf{p} = \varphi_x^2\varphi_{xx} + 2\varphi_x\varphi_y\varphi_{xy} + \varphi_y^2\varphi_{yy}, \tag{2.23}$$

in Cartesian coordinates. The partial derivatives $\nabla\varphi(\mathbf{x})$, the Hessian matrix $He(\varphi)$ are computed at the grid point $\mathbf{x}_{ij}$. The computed projections have third order of accuracy if $\nabla\varphi(\mathbf{x})$ and $He(\varphi)$ are computed using the standard centered five-point finite difference formulas.

The contact angle between the interface and the x-axis can be easily obtained from $\mathbf{n} = \nabla\varphi/|\nabla\varphi|$. It is well-known that we should keep the level set function as a good approximation to the signed distance function through a re-initialization process. A simple way is to solve the following Hamilton–Jacobi equation

$$\varphi_{\bar{t}} + \mathrm{sgn}(\varphi)(|\nabla\varphi| - 1) = 0. \tag{2.24}$$

through an artificial time $\bar{t}$, where $\mathrm{sgn}(\varphi)$ is the sign function of φ. Both Eqs. (2.18) and (2.24) are solved by the third order WENO scheme.

Note that since the free boundary cuts the bottom of the boundary, a simple treatment of the boundary for the level set function as used for closed interface cannot be used here. When we solve the level set function (2.18), we apply the boundary condition $v = 0$ and thus the level set equation on $y = c$ becomes a one-dimensional Hamilton–Jacobi equation that is solved by the WENO scheme. For the re-initialization process along $y = c$, after we updated the level set function for the interior points, we use the WENO scheme again in the x-direction, and approximate the partial derivative in y-direction using

$$\frac{\partial\varphi}{\partial y}(:,0) = \frac{\varphi_{:,2} - \varphi_{:,1}}{h_y} \tag{2.25}$$

in which $\varphi_{:,2}$ and $\varphi_{:,1}$ have already been updated as they are defined at interior grid points. In Fig. 2, we show a contour plot of the level set function after the proposed re-initialization process. The contour lines have almost the same distance, or $|\nabla\varphi| \sim 1$ as we desired.

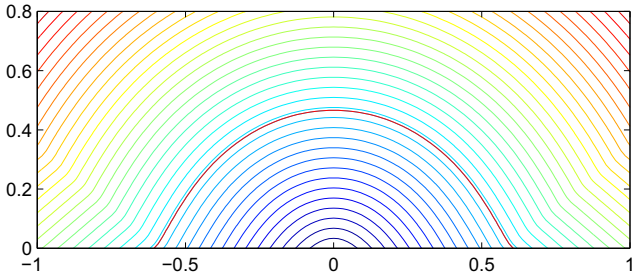

Fig. 2. A snap shot of a free boundary and the level set function after the proposed re-initialization process. The solid red line in the middle is the free boundary. We can see that the level set function is a good approximation of the signed distance function after the re-initialization. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

2.3. The augmented method and the Schur complement system

Since it is sufficient to consider the solution from one time level t^k level to the next t^{k+1}, we will ignore the time index for convenience if there is no confusion. Below we describe how to get the linear system of equations for the augmented variable $\mathbf{Q}$. Let $\mathscr{U}$ be the vector whose components are $\mathbf{U}_{ij}$ and P_{ij}, the approximate solution to the problem at one particular time step. Thus $\mathscr{U}$'s dimension is $O(3MN)$, where M and N are the number of grid lines in the x and y direction respectively. Let $\mathbf{Q}$ be the vector of the discrete values of $\mathbf{q}$ at the orthogonal projections of the irregular grid points from the Ω^+ side. Thus $\mathbf{Q}$'s dimension is $O(N)$ assuming $M \sim N$. Then the discrete solution of (2.9)–(2.11) and (2.12) given $\mathbf{Q}$ can be written as

$$A\mathscr{U} + B\mathbf{Q} = \mathbf{F}_1 \tag{2.26}$$

for some vector $\mathbf{F}_1$ and sparse matrices A and B. It requires solving three Helmholtz[1]/Poisson equations with different source terms and jump conditions to get $\mathscr{U}$.

Once we know the solution $\mathscr{U}$ given $\mathbf{Q}$, we can interpolate $\mathbf{U}$ and P to get $\nabla\mathbf{U}^-$ and P^- to approximate the one-phase boundary conditions at those points where the discrete values $\mathbf{Q}$ are defined. We denote $\nabla\mathbf{U}^-$ and P^- as the limiting values of $\nabla\mathbf{U}$ and P from the Ω^- side. The interpolation scheme depends on $\mathscr{U}, \mathbf{Q}$ *linearly*. Therefore we can write

$$E\mathscr{U} + T\mathbf{Q} - \mathbf{F}_2 = \mathbf{0}, \tag{2.27}$$

where E and T are two sparse matrices, and $\mathbf{F}_2$ is a vector. We need to choose such a vector $\mathbf{Q}$ that the free boundary conditions (1.3) and (1.4) are satisfied along the free boundary $\partial\Omega$. If we put the two matrix–vector Eqs. (2.26) and (2.27) together we get

$$\begin{bmatrix} A & B \\ E & T \end{bmatrix}\begin{bmatrix} \mathscr{U} \\ \mathbf{Q} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \end{bmatrix}. \tag{2.28}$$

Note that $\mathbf{Q}$ is defined only on a set of points $\{\mathbf{X}_l\}$ on the free boundary while $\mathscr{U}$ is defined at grid points. The Schur complement for $\mathbf{Q}$ is

[1] They are actually generalized Helmholtz equations $\Delta u - k^2 u = f$. For simplicity, we simply call them the Helmholtz equations in this paper if there is no confusion.

$$(T - EA^{-1}B)\mathbf{Q} = \mathbf{F}_2 - EA^{-1}\mathbf{F}_1 = \bar{\mathbf{F}}. \tag{2.29}$$

If we can solve the system above to get $\mathbf{Q}$, then we can get $\mathscr{U}$ easily. Because the dimension of $\mathbf{Q}$ is much smaller than that of $\mathscr{U}$, we expect to get a reasonably fast algorithm if we can solve (2.29) efficiently. We refer the readers to [19] for other details of an augmented immersed interface method.

Since we know how to get the matrix–vector multiplication, we can get the Schur complement matrix by taking $\mathbf{Q} = \mathbf{e}_i$, where $\mathbf{e}_i$ is the ith base vector. Then we can use the Gaussian elimination method to solve the Schur complement system. Or alternatively, we can use the GMRES iterative method [27] to solve the Schur complement system. The GMRES method only requires the matrix–vector multiplication. Since the free boundary is moving, the Schur complement matrix is changing with time, in most of our tests, it is more efficient to use the GMRES iterative method than that of forming the matrix, then applying the *LU* decomposition, see Fig. 4b. In our tests, we do not apply preconditioning techniques because we do not form the matrix. It is still an interesting numerical issue on how to develop efficient preconditioning techniques with only the knowledge of the matrix–vector multiplication. More details about the augmented method can be found in [19,21].

2.3.1. The least squares interpolation for the free boundary condition

One crucial step of the algorithm is to interpolate the free boundary conditions (1.3) and (1.4) given an approximate solution $\mathbf{U}_{ij}$ and P_{ij}. At an orthogonal projection $\mathbf{x}^*_{ij}$ corresponding to an irregular grid point $\mathbf{x}_{ij}$, we need to interpolate $\mathbf{U}_{ij}$ to get $\partial\mathbf{u}^-/\partial n$ and $\partial\mathbf{u}^-/\partial\tau$. This is computed using a least squares interpolation, for example,

$$\frac{\partial u^-}{\partial n} = \sum_{k=0}^{n_s-1} \gamma_k U_{i+i_k,j+j_k} - C_{ij} \tag{2.30}$$

where i_k and j_k are integers taken from $0, \pm1, \pm2, \ldots; n_s$ is the number of neighboring grid points involved, often taken between 9 and 16, C_{ij} is a correction term to offset the jump in the normal derivative of u. The coefficients γ_k are chosen so that the interpolation scheme is second order accurate. The linear system of equations γ_k is obtained by expanding $u(\mathbf{x}_{ij})$ at $\mathbf{x}^*_{ij}$ and then matching up to second order partial derivative terms. If $u(\mathbf{x}_{ij})$ involved is from outside of Ω, then it is replaced by the extended value

$$\begin{aligned} u(\mathbf{x}_{ij}) &= u^+ + u^+_\xi \xi + u^+_\eta \eta + \tfrac{1}{2}\left(u^+_{\xi\xi}\xi^2 + 2u^+_{\xi\eta}\xi\eta + u^+_{\eta\eta}\eta^2\right) + O(h^3) \\ &= u^- + u^-_\xi \xi + u^-_\eta \eta + \tfrac{1}{2}\left(u^-_{\xi\xi}\xi^2 + 2u^-_{\xi\eta}\xi\eta + u^-_{\eta\eta}\eta^2\right) \\ &\quad + [u_\xi]\xi + [u_\eta]\eta + \tfrac{1}{2}\left([u_{\xi\xi}]\xi^2 + 2[u_{\xi\eta}]\xi\eta + [u_{\eta\eta}]\eta^2\right) + O(h^3) \end{aligned}$$

where (ξ, η) is the local coordinate system in the normal and tangential directions, and all the values are defined at $\mathbf{x}^*_{ij}$. Since we know the jump conditions $[u]$ and $[u_n] = [u_\xi]$, we can obtain all other jump conditions, see [19]. Note that we need to take $n_s \geqslant 6$ to get a consistent linear system. Thus the linear system of equations for the coefficients γ_k is under-determined, which can be solved by the singular value decomposition (SVD).

2.4. Solving an elliptic interface problem with singular sources

Our proposed algorithm consists of solving two generalized Helmholtz equations for the intermediate velocity $\mathbf{u}^*$ in (2.7), and one Poisson equation for the pressure increment ϕ^{k+1} in (2.10) 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 [18–20]. 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

$$\begin{aligned} &\psi_{xx} + \psi_{yy} - \lambda\psi = f, \qquad (x,y) \in R, \\ &[\psi]_{\partial\Omega} = 0, \qquad \left[\frac{\partial\psi}{\partial n}\right]_{\partial\Omega} = q. \end{aligned} \tag{2.31}$$

In our application, we have $\lambda = 2/(\mu\Delta t)$ for the prediction step, and $\lambda = 0$ for ϕ^{k+1} in (2.10). 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

$$\frac{\psi_{i+1,j} + \psi_{i-1,j} - 2\psi_{i,j}}{h_x^2} + \frac{\psi_{i,j+1} + \psi_{i,j-1} - 2\psi_{i,j}}{h_y^2} - \lambda\psi_{ij} = f(x_i, y_j) + C_{ij}, \tag{2.32}$$

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 [20] for the formula of C_{ij}.

We use the structured multi-grid solver DMGD9V [7] to solve the discrete system for $\mathbf{u}^*$. While using the fast Poisson solver from [1] for ϕ. This is because of the hybrid discretization for $\mathbf{u}^*$ that changes the coefficient in the Helmholtz equations for $\mathbf{u}^*$, and different boundary conditions for u at the bottom side.

3. Numerical experiments

In this section, we present some numerical results for the free boundary problem with moving contact lines. All the computations were performed at the North Carolina State University using either notebook or desktop computers. Most simulations are done within minutes to a couple of hours depending on the mesh, surface tension, the initial geometry, and the static contact angle. We use a level set method to evolve the free boundary for convenience since some of the codes used for the results in [13] can be used. We believe that the front tracking method would work equally well in two space dimensions. In most of simulations, we take the density $\rho = 1$, the viscosity $\mu = 2$, and the surface tension $\gamma = 0.5$. The other parameters are the slip coefficient $\beta = 2$, the effective friction coefficient $\alpha = 0.5$, and the smoothing length $w = h$, unless stated otherwise. The time step size is take as

$$\Delta t^k = \min\left\{\frac{h}{2}, \frac{h}{2\mathbf{U}^k_{max}}\right\}, \tag{3.33}$$

where $h = \max\{h_x, h_y\}$ and $U^k_{max} = \max_{ij}|\mathbf{U}^k_{ij}|$ unless specified otherwise.

Example 1 (*Validation of the method against an exact solution*). As a first numerical test for our scheme, we consider an example in a stationary irregular domain in which the exact solution is known analytically. This is an example without the effect of moving contact lines which we simply use it as an accuracy check. The analytic solution is

$$u(x,y,t) = \begin{cases} \sin(t)\left(\dfrac{y}{r} - 2y\right), & \text{if } r \geqslant 1/2 \\ \sin(t)\left(r^2 - \frac{1}{4}\right)y, & \text{otherwise,} \end{cases} \tag{3.34}$$

$$v(x,y,t) = \begin{cases} \sin(t)\left(-\dfrac{x}{r} + 2x\right), & \text{if } r \geqslant 1/2 \\ -\sin(t)\left(r^2 - \frac{1}{4}\right)x, & \text{otherwise,} \end{cases} \tag{3.35}$$

$$p(x,y,t) = \sin(t)(\sin x + \sin y), \tag{3.36}$$

where $r = \sqrt{x^2 + y^2}$. The domain of the interest is the domain bounded by $r \leqslant 1/2$ and $y \geqslant 0$. We extend the solution to the rect-

Table 1
A grid refinement analysis against the exact solution at a final time $T = 0.5$. $\|E_u\|_\infty$ is the sum of the maximal error in the velocity component u and v, *order* is the approximated convergence order computed from the two consecutive errors, *cond* is the condition number of the coefficient matrix for the unknown jump in the normal derivative of the velocity at the first step.

$M \times N$	$\|E_u\|_\infty$	$order_u$	$\|E_p\|_\infty$	$order_p$	*cond*
32×16	9.4533×10^{-3}		7.2203×10^{-3}		11.90
64×32	2.2300×10^{-3}	2.0838	1.8997×10^{-3}	1.9278	27.42
128×64	6.0532×10^{-4}	1.8813	6.0351×10^{-4}	1.6528	29.66
256×128	9.7791×10^{-5}	2.6290	2.8304×10^{-4}	1.0923	190.0

angular domain $R = [-1, 1] \times [0, 1]$ so that we have a consistent initial condition and a problem with the exact jump condition in the normal derivative of the velocity. The source term **G** is derived directly from the exact solution.

In Table 1, we show the grid refinement analysis to check the order of the accuracy of our method. We use the exact non-homogeneous normal derivative boundary condition at ∂R. Since we are interested in the computed solutions in the domain Ω, we set

$$\|E_u\|_\infty = \max_{r_{ij} \leqslant 1/2,\, y \geqslant 0} \left\{ |U_{ij}^k - u(x_i, y_j, T)| \right\} + \max_{r_{ij} \leqslant 1/2,\, y \geqslant 0} \left\{ |V_{ij}^k - v(x_i, y_j, T)| \right\}$$

$$\|E_p\|_\infty = \max_{r_{ij} \leqslant 1/2,\, y \geqslant 0} \left\{ |P_{ij}^k - p(x_i, y_j, T)| \right\}$$

to be the error in the velocity at time T. The numbers *order* are the approximated order of accuracy from the two consecutive errors for the velocity. Second order accuracy is clearly seen for the velocity. The pressure is at least first order accurate. The number *cond* is the condition number of the coefficient matrix for the unknown jump in the normal derivative of the velocity at the first step.

Example 2 (*Drop spreading and recoiling*). We started with the same example that was used in [26]. The initial drop is a semicircle centered at the origin with radius $r = 0.5$, and $\mu = 2$. Thus the initial contact angle between the boundary and the x-axis is $\pi/2$. The surface tension is taken as $\gamma = 0.5$. In Fig. 3a, we show the initial droplet and its shape at $t = 3.9060$. The static contact angle is $\theta^* = \pi/4$ so the drop would spread as we can see from the simulation. The simulation was computed using a 128 by 64 grid. We use homogeneous Neumann boundary condition on ∂R except for the bottom side where we use the Navier BC for u and $v = 0$. We set $u = 0, v = 0$, and $p = 0$, initially. We keep the boundary fixed and solve the problem to get consistent initial data before letting the free boundary to move. The condition number of the Schur complement matrix is well under 200. In Fig. 3b, we show a recoiling case by plotting the initial free boundary, $r = 0.4$ and $y \geqslant 0$, and its shape at $t = 0.9586$ without the gravity force. The static angle now is $\theta^* = 3\pi/4$. In this case, the drop contracts at the solid surface. Also for this case, the condition number of the Schur complement matrix is larger than that of the spreading case.

In Fig. 4a, we show the time evolution of the contact angle and the contact line speed in the spreading case which the static contact angle is $\pi/4$. The contact angle and line speed do not decrease exactly in a monotonic way to the equilibrium state due the interaction of the fluid and the surface tension.

In Fig. 4b, we show the number of iterations of the GMRES iteration at each time level for the spreading case. The mesh size is 128 by 64. The size of the Schur complement system is about 178 by 178 for the augmented variable. The convergence tolerance is 10^{-5}. The number of iterations of the GMRES fluctuates but seldom exceeds its full dimensions. It is faster than that when we form the matrix of the Schur complement and then use the *LU* decomposition to solve the system of equations. However, while the condition number is small to modest, the number of iterations of the GMRES is not close to a constant and can be quite large if the tolerance is very small. This is because the matrix is far from a normal matrix ($AA^T = A^TA$). It is a challenging problem to develop an efficient linear solver for the Schur complement system of equations knowing only the matrix–vector multiplications.

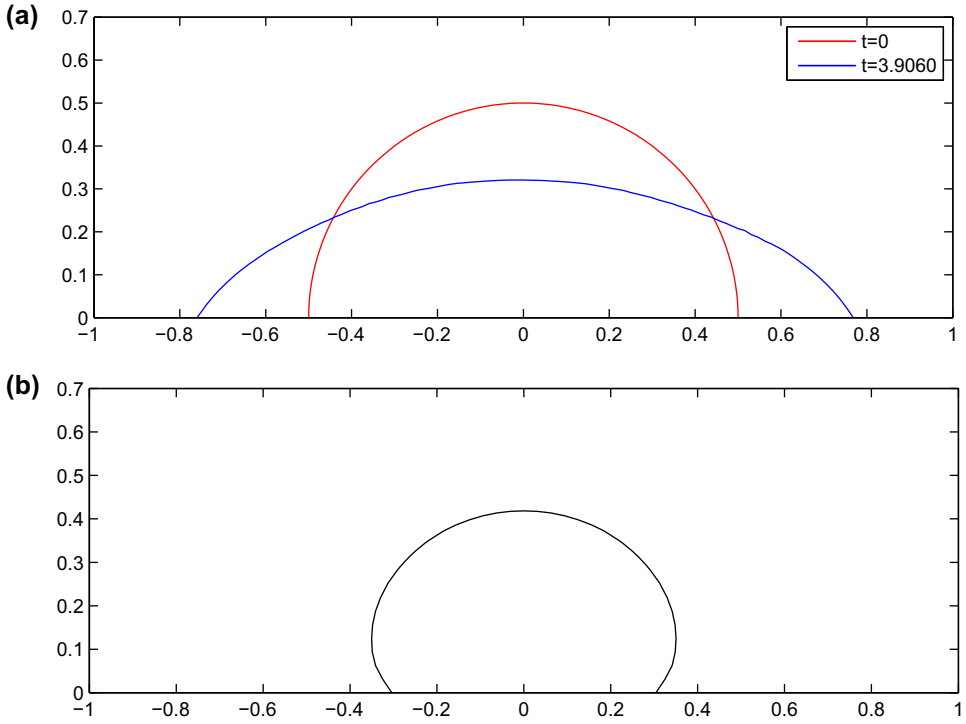

Fig. 3. (a) A drop spreads along the contact line with $\theta^* = \pi/4$. The initial free boundary is $r = 0.5$ above the x-axis. The red-line plot is the initial free boundary. (b) A drop recoils along the contact line with $\theta^* = 3\pi/4$ with $t = 0.9586$. The initial free boundary is $r = 0.4$ above the x-axis. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

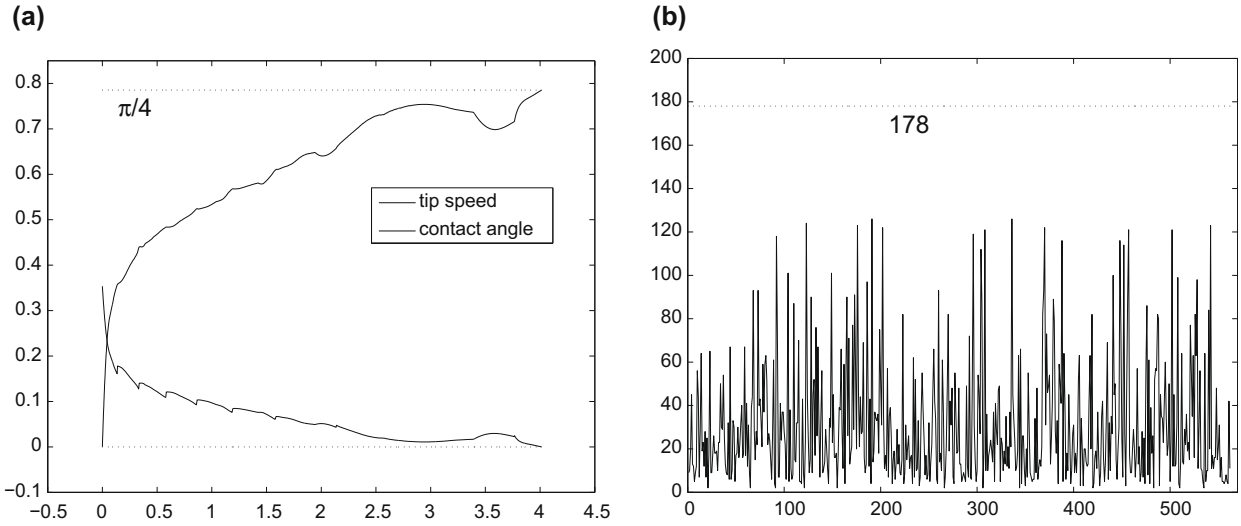

Fig. 4. (a) A history of the contact angle and the contact line speed as functions of time. (b) The number of GMRES iterations for the Schur complement system whose size is about 178 by 178 for a 128 by 64 grid.

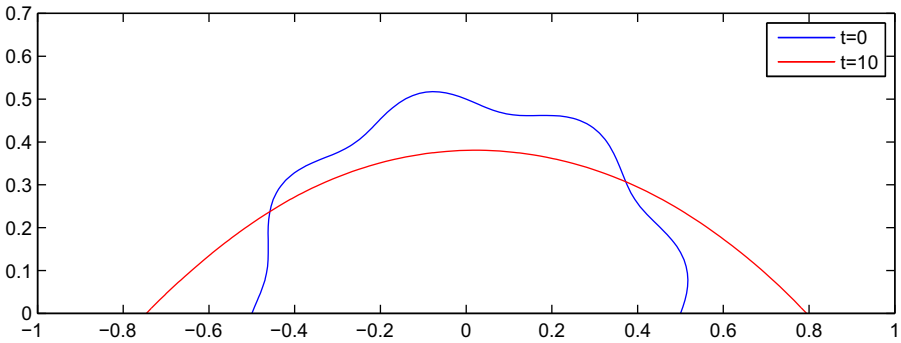

Fig. 5. A drop with perturbation spreads along the contact line.

Example 3 (*A perturbed surface*). In Fig. 5a, we show the free boundary at $t = 0$, and at a time $t = 10$. The initial boundary was a perturbed semi-circle $r = 0.5 + 0.05\sin(8\theta), 0 \leqslant \theta \leqslant \pi$. In this case, the surface tension smooths the free boundary while the contact line is moving. The static contact angle is $\theta^* = \pi/4$.

Example 4 (*The effect of the gravity*). In Fig. 6, we show the numerical experimental results of the simulation with and without gravity. The initial free boundary is the half circle $r = 0.4$ and $y \geqslant 0$ centered at the origin. Thus, the initial angle $\theta = \pi/2$. We assume that the static angle is $\theta^* = \pi/4$. Without the gravity, the motion is slower, see the plot of the green line . With the gravity, the motion is faster and flatter, see the plot of the blue line.

Example 5. This example is adapted from [17]. The initial free boundary is $r = 0.3$ and $y \geqslant 0$ centered at $(0, 0.2)$. The initial angle $\theta > \pi/2$. We assume again that the static angle is $\theta^* = \pi/4$. Fig. 7 shows some snap shots of the free boundary at different time. The GMRES method does take more number of iterations.

4. Conclusions

In this paper, we propose an augmented immersed interface method for solving one-phase free boundary problems with moving contact lines. The one-phase model is a simplification of two-phase model under some assumptions. The one-phase model

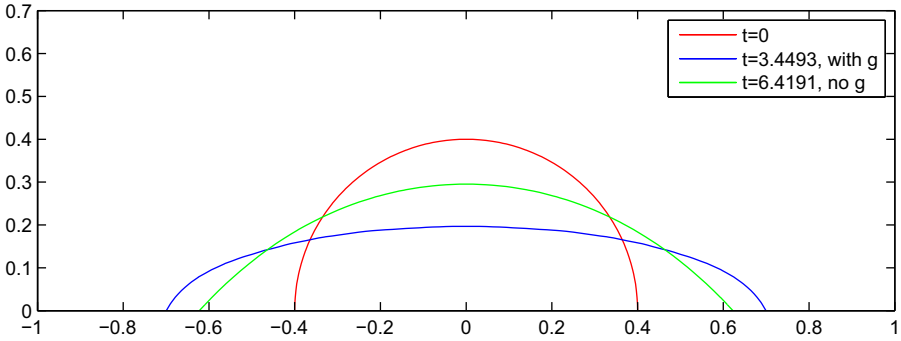

Fig. 6. A comparison of drop spreading with and without the gravity.

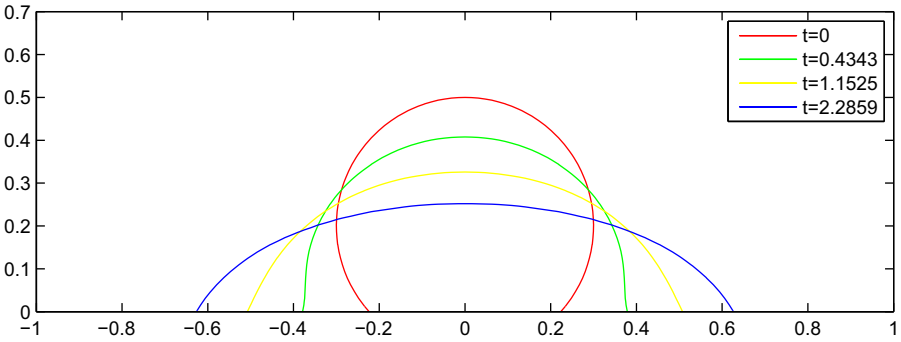

Fig. 7. Snap shots of the free boundary in which the initial angle $\theta > \pi/2$.

is valid if, the boundary conditions at the contact lines are nearly consistent. The method can track the interactions of the fluid equations and the moving contact lines. By introducing the augmented variable along the boundary, we can solve the problem in a rectangular domain. We use a hybrid time discretization with backward Euler at irregular grid points and Crank-Nicholson at regular grid points for the prediction step to have better stability. The linear system of equations is well-conditioned too.

Acknowledgement

We would like to thank Dr. Weinan E (Princeton) for bringing us this interesting problem. We would also like to thank Drs. Weiqing Ren (NYU), Kazufumi Ito and Michael Shearer (NCSU) for valuable discussions.

The author would like to acknowledge the following supports. The first author is partially supported by US-ARO Grants 56349-MA and 49308-MA, US-AFSOR Grant FA9550-09-1-0520, and NSF Grant 0911434. The second author is partially supported by National Science Council of Taiwan under Grant NSC-97-2628-M-009-007-MY3 and MoE-ATU project.

References

[1] Adams J, Swarztrauber P, Sweet R. Fishpack: efficient Fortran subprograms for the solution of separable elliptic partial differential equations. %3chttp://www.netlib.org/fishpack/%3e.
[2] Beale JT, Layton AT. On the accuracy of finite difference methods for elliptic problems with interfaces. Math Comput Sci 2006;1:91–119.
[3] Bell JB, Colella P, Glaz HM. A second-order projection method for the incompressible Navier-Stokes equations. J Comput Phys 1989;85:257–83.
[4] Beyer RP, LeVeque RJ. Analysis of a one-dimensional model for the immersed boundary method. SIAM J Numer Anal 1992;29:332–64.
[5] Brown DL, Cortez R, Minion ML. Accurate projection methods for the incompressible Navier-Stokes equations. J Comput Phys 2001;168:464.
[6] Dussan EBV, Davis SH. On the motion of a fluid–fluid interface along a solid surface. J Fluid Mech 1974;65:71.
[7] De Zeeuw D. Matrix-dependent prolongations and restrictions in a blackbox multigrid solver. J Comput Appl Math 1990;33:1–27.
[8] Ganesan S, Tobiska L. Modeling and simulation of moving contact line problems with wetting effects. Comput Visual Sci 2008. doi:10.1007/s00791008-0111-3.
[9] de Gennes P-G, Brochard-Wyart F, Quere D. Capillarity and wetting phenomena. Springer; 2003.
[10] Haley PJ, Miksis MJ. The effect of the contact line on droplet spreading. J Fluid Mech 1991;223:57–81.
[11] Huang H, Liang D, Wetton B. Computation of a moving drop/bubble on a solid surface using a front-tracking method. Commun Math Sci 2004;2:535–52.
[12] Huh C, Scriven LE. Hydrodynamics model of steady movement of a solid/liquid/fluid contact line. J Colloid Interface Sci 1971;35:85.
[13] Hunter J, Li Z, Zhao H. Autophobic spreading of drops. J Comput Phys 2002;183:335–66.
[14] Joseph DD, Renardy YY. Fundamentals of two-fluid dynamics. Springer-Verlag; 1993.
[15] Van Kan J. A second-order accurate pressure-correction scheme for viscous incompressible flow. SIAM J Sci Comput 1986;7:870–91.
[16] Kim J, Moin P. Application of a fractional-step method to incompressible Navier-Stokes equations. J Comput Phys 1985;59:308–23.
[17] Li J, Renardy M. Numerical simulation of moving contact line problems using a volume-of-fluid method. J Comput Phys 2001;171:243–63.
[18] Li Z. A fast iterative algorithm for elliptic interface problems. SIAM J Numer Anal 1998;35:230–54.
[19] Li Z, Ito K. The immersed interface method – numerical solutions of PDEs involving interfaces and irregular domains. SIAM Front Ser Appl Math 2006;FR33.
[20] Li Z, Lai M-C. The immersed interface method for the Navier-Stokes equations with singular forces. J Comput Phys 2001;171:822–42.
[21] Li Z, Wan X, Ito K, Lubkin SR. An augmented approach for the pressure boundary condition in a Stokes flow. Commun Comput Phys 2006;1:874–85.
[22] Liu H, Krishnan S, Marella S, Udaykumar HS. Sharp interface Cartesian grid method II: a technique for simulating droplet interactions with surfaces of arbitrary shape. J Comput Phys 2005;210:32–54.
[23] Osher S, Fedkiw R. Level set methods and dynamic implicit surfaces. New York: Springer; 2002.
[24] Pismen LM, Rubinstein BY, Bazhlekov I. Spreading of a wetting film under the action of van der waals forces. Phys Fluids 2000;12:480–3.
[25] Qian T, Wang X-P, Sheng P. Molecular hydrodynamics of the moving contact line in two-phase immiscible flows. Commun Comput Phys 2006;1(1):1–52.
[26] Ren W, WE. Boundary conditions for the moving contact line problem. Phys Fluids 2007;19.
[27] Saad Y. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J Sci Stat Comput 1986;7:56–869.
[28] Sethian JA. Level set methods and fast marching methods. 2nd ed. Cambridge University Press; 1999.
[29] Socolowsky J. The solvability of a free boundary problem for the stationary Navier-Stokes equations with a dynamic contact line source. Nonlinear Anal 1993;21:763–84.
[30] Spaid MA, Homsy GM. Stability of Newtonian and viscoelastic dynamic contact lines. Phys Fluids 1996;8:460–78.
[31] Spelt PDM. A level-set approach for simulations of flows with multiple moving contact lines with hysteresis. J Comput Phys 2005;207:389–404.
[32] Wu C, Lei S, Qian T, Wang X. Stick-slip motion of moving contact line on chemically patterned surfaces. Commun Comput Phys 2010;7:403–22.