Microfluid Nanofluid
DOI 10.1007/s10404-011-0821-6

RESEARCH PAPER

A full-Eulerian solid level set method for simulation of fluid–structure interactions

Ping He · Rui Qiao

Received: 4 April 2011 / Accepted: 15 May 2011
© Springer-Verlag 2011

Abstract We present a full Eulerian method, termed solid level set (SLS) method, for modeling of a class of fluid–structure interactions (FSI) problems soft solid body can deform significantly but remains nearly incompressible. The SLS method is based on the unified momentum equation framework in which the solid–fluid interactions are modeled by introducing a solid body force term and a solid–fluid interfacial force term into the Navier–Stokes equation. The key idea of the SLS method is that the deformation of the solid body is no longer tracked using a Lagrangian mesh. Instead, the solid body is tracked by introducing a reference coordinate for describing the reference state of the solid body and by introducing three dynamic level set functions on the Cartesian coordinate and one static level set functions on the reference coordinate. The SLS method is easy to implement and addresses several challenges in the simulation of FSIs in which a fixed Cartesian mesh is used for fluid flow and a Lagrangian mesh is used for tracking the solid deformation. The effectiveness of the SLS method is demonstrated by studying two FSI problems. The method is suitable for studying a wide range of problems in microfluidics, e.g., manipulation of cells in confined space and ink-jet printing of biological samples.

Keywords Fluid–structure interactions · Microfluidics · Fixed grid methods · Level set methods · Droplet dynamics

P. He · R. Qiao (✉)
Department of Mechanical Engineering, Clemson University, Clemson, SC 29634, USA
e-mail: rqiao@clemson.edu

1 Introduction

Fluid–structure interactions (FSI) play an important role in many fields such as swimming of small biological entities, cardiovascular circulation, and manipulation of biological entities using microfluidics. In the past decades, many numerical methods have been developed to study this problem. Of the many methods developed, fixed grid methods such as the immersed boundary method (IBM) (Peskin 1972) and the immersed interface method (IIM) (Leveque and Li 1994) are very popular. In these methods, the fluids are described on fixed Cartesian meshes and the elastic structures (typically fibers and membranes) are described by Lagrangian meshes. The elastic structures are evolved through convection by fluid flow, and the coupling between the structure and fluid motion is accomplished by incorporating a singular surface force originated from the deformation of solid structure into the Navier–Stokes (N–S) equations. The advantages of these methods include simplicity and versatility. In particular, (1) the coupling between fluid and complex structure is straightforward; (2) re-meshing during solution of fluid flow can often be avoided; and (3) efficient linear solvers are easy to be adopted with fixed grids. Although many of these methods and other novel numerical schemes (Guendelman et al. 2005) have been used to study fiber or membrane-like structures, fixed grid methods which can simulate interactions between fluids and flexible solid bodies have also been developed. For example, Zhao, Freund, and Moser (ZFM) (Zhao et al. 2008) recently proposed an efficient and accurate method that solves FSI problems involving flexible bodies using a unified equation of motion for both the solids and the fluids. The key idea was to formulate a combined momentum equation in the entire simulation domain, and the difference between the solids and fluids

Published online: 27 May 2011

lies in the governing laws of the stresses in them. The solid structure is discretized into a Lagrangian mesh to provide the tracking of solids for computations of solid strains and stresses, which are subsequently transferred to the fluid solver as body forces and surface forces using several methods. The combined momentum equation was solved on the fixed mesh, and the computed fluid velocity was used to convect the solid body. Two key challenges were encountered and successfully resolved in the ZFM method. First, the solid stresses computed using finite element method (FEM) are discontinuous across the neighboring elements, and using these stresses directly in the momentum equation reduces numerical accuracy and potentially leads to instability. Second, to march the solid structure, the flow velocity at the position of Lagrangian mesh must be interpolated from that on the fixed fluid mesh. Since the interpolated velocity field is not necessarily divergence-free and the solid constitutive law itself does not prevent local volume change, solid structure advected simply using the interpolated velocity could lead to volume change, which eventually causes instability as evident from numerical simulations. To resolve the first issue, the Zienkiewicz–Zhu (ZZ) patch (Zienkiewicz and Zhu 1992) method is used to recover a continuous solid stress field following the FEM stress calculation. To resolve the second issue, a correction displacement is added to the displacement computed from fluid convection to enforce the incompressibility condition. With these issues addressed, the ZFM's method has been used to solve several highly non-trivial FSI problems.

The ZFM method represents a significant step in the development of FSI simulation method. Nevertheless, several issues still remain. First, although the ZZ-patch and displacement correction techniques help ensure the accuracy and stability of the ZFM method, they considerably increase its computational complexity. Second, since the incompressibility of solid bodies is enforced by computing a displacement correction that yields the minimal L_2 norm of $|\det \mathbf{A} - 1|$ ($\mathbf{A}$ is the deformation gradient tensor of the solid body), this condition may not be enforced exactly at every point inside the simulation domain. Third, re-meshing is still required when the solid body deforms significantly. Since generating high-quality FEM mesh for three-dimensional objects with complex shape is a challenging problem, such a requirement should preferably be avoided. To resolve these issues, one possible approach is to solve the coupled fluid and structure motion in a pure Eulerian framework. In such an approach, the solid object is not tracked using a Lagrangian mesh but traced on the Cartesian mesh over which the fluid flow is solved. Consequently, the ZZ-patch and displacement correction steps can be entirely avoided. The third issue can now be alleviated by locally refining the Cartesian mesh, which is more straightforward than re-meshing the FEM mesh. The major new challenge is how to compute the strain and stress in the solid body without using explicit Lagrangian points to track the deformation of a solid body with respect to its resting shape.

In this study, we develop a full Eulerian method, herein termed solid level set (SLS) method, for solving FSI problems. Only a fixed fluid mesh is used in the simulation, and the dynamics of fluids and solids are modeled by the combined momentum equation as in the ZFM method. The solid body is tracked by four level set functions: an interface signed distance function for the contour of the solid body at the resting shape, an interface signed distance function for the contour of the solid body at any given time, a reference X-level set function and a reference Y-level set function for the interior of the solid body. X- and Y- level set functions are marched using a special method, and are used to compute the solid deformation and stresses. The method otherwise follows the same framework as that of the ZFM method. The rest of the article is organized as follows: in Sect. 2, we present the physical and mathematical model for the FSI problem studied here; in Sect. 3, we present the numerical algorithm for the SLS method; in Sect. 4, we demonstrate the effectiveness of the SLS method through two FSI problems; finally, conclusions are presented in Sect. 5.

2 Physical and mathematical model of FSIs

To demonstrate our proposed method, we consider a class of FSI problem sketched in Fig. 1. Specifically, the fluids are incompressible Newtonian multi-/single-fluid(s) with different viscosities and densities. The solids are incompressible neo-Hookean elastic bodies with a pseudo-viscosity.

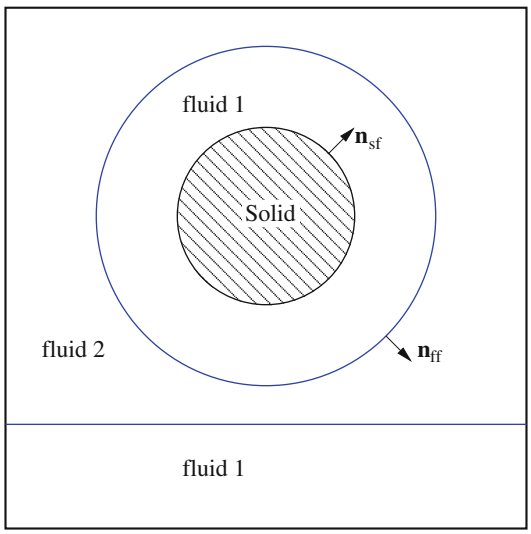

Fig. 1 Physical model of a multi-fluids FSI system

The density and pseudo-viscosity of the solids are the same as the fluids in the system (if several fluids are present in the system, the solid density/viscosity will be the same as the fluids in which the solid body is immersed).

The fluid flow is governed by the N–S equations for incompressible Newtonian fluids,

$$\nabla \cdot \mathbf{u} = 0 \tag{1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla \cdot \mu_{\mathrm{f}} \nabla \mathbf{u} + \mathbf{g}, \tag{2}$$

where $\mathbf{u}$ is the vector of velocity, ρ is the fluid/solid density, μ_{f} is the fluid viscosity or the solid pseudo-viscosity, $\mathbf{g}$ is the gravity, and t is time. Neo-Hookean law is used to model the solid, but other constitutive models can also be easily adopted. The motion of solid body is governed by

$$\frac{\partial^2 \mathbf{x}}{\partial t^2} = \mathbf{B} \tag{3}$$

$$\text{in which} \quad \mathbf{B} = \frac{1}{\rho}\nabla \cdot \boldsymbol{\tau}_{\mathrm{elas}} \tag{4}$$

$$\boldsymbol{\tau}_{\mathrm{elas}} = \mu_{\mathrm{s}}\left(\mathbf{A} \cdot \mathbf{A}^T - \mathbf{I}\right) \tag{5}$$

$$\mathbf{A} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}, \tag{6}$$

where $\mathbf{X}$ is the reference coordinates for any point in the solid body at the resting shape; $\mathbf{x}$ is the current coordinates for the corresponding point; $\mathbf{A}$ is the deformation gradient tensor; μ_{s} is the elastic modulus for neo-Hookean law; $\mathbf{I}$ is the identity matrix; $\boldsymbol{\tau}_{\mathrm{elas}}$ is the elastic stress tensor; and $\mathbf{B}$ is the solid body force.

The FSI problem sketched involves fluid–fluid interfacial interactions and FSIs. To solve the entire problem in a unified framework, the fluid–fluid interactions will be modeled using the continuum surface force (CSF) method (Brackbill et al. 1992), and the FSIs will be modeled using the method developed by ZFM (Zhao et al. 2008). Consequently, the fluid and solid dynamics are governed by

$$\nabla \cdot \mathbf{u} = 0 \tag{7}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla \cdot \mu_{\mathrm{f}} \nabla \mathbf{u} + \mathbf{g} + \mathbf{F}_{\mathrm{ff}} + \chi \mathbf{B} + \mathbf{F}_{\mathrm{sf}} \tag{8}$$

where the solid indicator function χ is 1 for region inside any solid body and 0 otherwise. The fluid–fluid interactions in the multi-fluid systems are described using a surface force,

$$\mathbf{F}_{\mathrm{ff}} = -\frac{1}{\rho}\sigma\kappa\mathbf{n}_{\mathrm{ff}}\delta_{\mathrm{ff}}, \tag{9}$$

in which, σ is the surface tension between two types of fluids, κ is the curvature of a fluid–fluid interface, $\mathbf{n}_{\mathrm{ff}}$ is the normal direction of the fluid–fluid interface, and δ_{ff} is the Dirac delta function to describe the fluid–fluid interface. The FSI is also described as a surface force using a δ-function,

$$\mathbf{F}_{\mathrm{sf}} = -\frac{1}{\rho}\boldsymbol{\tau}_{\mathrm{elas}} \cdot \mathbf{n}_{\mathrm{sf}}\delta_{\mathrm{sf}}. \tag{10}$$

Equations 4–10 constitute the governing equations for the FSI problem sketched in Fig. 1 to be solved numerically.

3 Numerical algorithm

The FSI problem described above involves FSIs and interactions between different types of fluids. The problem is solved on staggered Cartesian grids (Zhao et al. 2008), in which density, viscosity, and velocities are defined at the four corners of each grid cell, and the pressure is defined at the center of the grid cell. Marching the system from time step n to $n+1$ essentially involves four sub-steps: (1) resolving fluid–fluid interfaces and computing fluid–fluid interactions terms in Eq. 9, (2) resolving solid structures and their deformation, and computing $\mathbf{B}$ and $\mathbf{F}_{\mathrm{sf}}$ given by Eqs. 4–6, 10 to account for the FSIs, (3) marching fluid–fluid interfaces and solid structure to time step $n+1$ using the fluid velocity at the time step n, and finally, (4) computing the fluid velocity and pressure inside the entire domain at the time step $n+1$ by solving Eqs. 7, 8. Below we provide details of each sub-step with a focus on sub-steps 2 and 3.

3.1 Resolving fluid–fluid interface and computing fluid–fluid interactions

We use the level set method to keep track of the fluid–fluid interfaces (Osher 1993; Sethian 1991; Sussman et al. 1994; Osher and Fedkiw 2002). Briefly, a level set function ϕ is defined on an Eulerian mesh and used to describe a certain region in the system (e.g., one type of fluid in a multi-fluids system) and its interface with other regions, e.g.,

$$\phi(\mathbf{x}) \begin{cases} <0, & \mathbf{x} \text{ is inside the region} \\ =0, & \mathbf{x} \text{ is on the interface} \\ >0, & \mathbf{x} \text{ is outside the region} \end{cases} \tag{11}$$

Based on Eq. 11, the fluid–fluid interface can be defined as the 0-level contour of the level set function ϕ. Level set functions are usually defined as a signed distance function, i.e., its value at a given point is taken as the closest distance of this point to the interface, and its sign is negative (positive) if the point is inside (outside) the region.

The advection, viscous terms, and $\mathbf{F}_{\mathrm{ff}}$ in Eq. 8 are evaluated as follows. The advection term is computed using the Godunov method (Colella 1985; Sussman et al.

1999). At positions away from the fluid–fluid interface, the viscous terms in Eq. 8 are computed straightforwardly using central differences. At positions near the fluid–fluid interface, the viscous term is computed by smearing off the density and viscosity of fluids in a narrow band area from the fluid–fluid interface. Specifically, for any fluid property λ (e.g., ρ or μ), its value at a position near the fluid–fluid interface can be computed with the help of the level set function by

$$\lambda = \frac{\sum_{i=1}^{n} \lambda_i H(-\phi_i)}{\sum_{i=1}^{n} H(-\phi_i)} \tag{12}$$

where n is the number of fluid types, and H is the numerical Heaviside step function given by

$$H(\phi) = \begin{cases} 0, & \phi < -\varepsilon \\ \frac{1}{2} + \frac{\phi}{2\varepsilon} + \frac{1}{2\pi}\sin\left(\frac{\pi\phi}{\varepsilon}\right), & -\varepsilon \le \phi \le \varepsilon \\ 1, & \phi > \varepsilon \end{cases} \tag{13}$$

where ε is the half width of the narrow band across the interface, over which the fluid properties are smeared off. To compute the fluid–fluid interfacial force in Eq. 9, a numerical δ-function is defined to transfer the fluid–fluid interfacial force as a body force to fluids near the interfaces using the CSF method (Brackbill et al. 1992)

$$\delta(\phi) = \begin{cases} 0, & \phi < -\varepsilon \\ \frac{1}{2\varepsilon} + \frac{1}{2\varepsilon}\cos\left(\frac{\pi\phi}{\varepsilon}\right), & -\varepsilon \le \phi \le \varepsilon \\ 0, & \phi > \varepsilon \end{cases} \tag{14}$$

The normal vector and the surface curvature in Eqs. 9, 10 can be computed using the level set function (Osher 1993; Sethian 1991; Sussman et al. 1994; Osher and Fedkiw 2002)

$$\kappa = \nabla^2\phi, \quad \mathbf{n} = \nabla\phi. \tag{15}$$

3.2 Resolving solid structure and computing **B** and $\mathbf{F}_{sf}$

To keep track of the solid body and its deformation and stress, in addition to the Cartesian coordinates (x, y), we introduce reference coordinates (X, Y) to provide the coordinates of the material points of the solid body at a reference state in two dimensional spaces (or in asymmetric configurations). In this study, the reference state is taken as the state of system at $t = 0$, and thus the reference coordinates coincide with the fixed Cartesian coordinates at $t = 0$. We define four level set functions to describe the solid body (see Fig. 2):

1. $X(x, y)$ and $Y(x, y)$: these level sets are initialized by setting $X(x, y, t = 0) = x$ and $Y(x, y, t = 0) = y$, and marched in time using the fluid velocity (to be detailed in Sect. 3.3). Therefore, for any material point inside the system, these functions provide a mapping from its current Cartesian coordinates (x, y) to its coordinates in the reference state.
2. $\phi_s^0(X,Y)$: a signed distance level set function defined on the reference coordinates whose 0-level contour represents the interface between the solid body and its surrounding fluids.
3. $\phi_s(x, y, t)$: a signed distance level set function defined on the Cartesian coordinates whose 0-level contour represents the interface between the solid body and its surrounding fluids.

The first two level set functions, X- and Y-level sets, together enable us to track the movement of material points

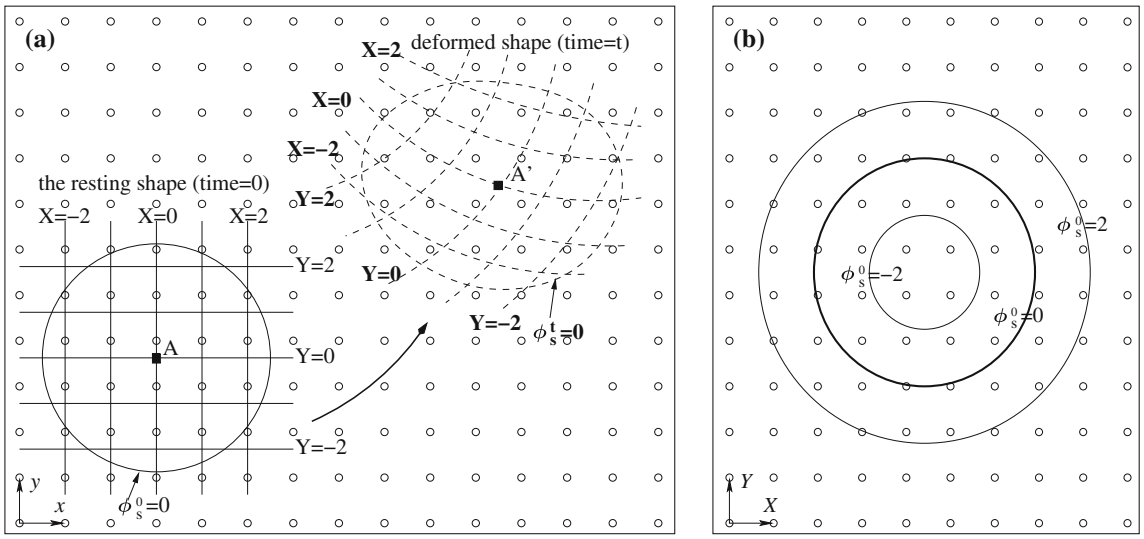

Fig. 2 Illustrations of the level set functions used to track the solid body: **a** An illustration of SLS functions in the resting shape (*solid lines*) and a deformed shape (*dashed lines*). The *circles* are the fixed Eulerian grid points on which ϕ_s^t, X- and Y-level set functions are defined. All these functions are defined on the Cartesian coordinates (x, y). **b** The solid interface level set function ϕ_s^0, which is a signed distance function defined on the reference coordinates (X, Y). This function remains the same during the simulation

inside the solid body. Figure 2 illustrates how a material point is tracked using these functions. For example, assume that a material point inside the solid body (denoted here by a filled square) is located at point A at time $t = 0$. The position of this material point at time $t = \Delta t$, A', can be obtained by noting that it is located at the intersection of the 0-level contours of the X- and Y-level set functions at both $t = 0$ and $t = \Delta t$. Note that the X- and Y-level set functions are not signed distance functions. These two level set functions can be defined in the entire simulation domain, but their exact values, which are used to track the movement of material points, are needed only inside the solid body.

The third and fourth level set functions, ϕ_s^0 and ϕ_s, are used to describe the fluid–solid interface. $\phi_s^0(X,Y)$ is defined on the reference coordinate, and once initialized, it will not change during the simulation. Since it does not need to be updated during the simulation, the grid on which it is described, ΔX, can be very fine to provide accurate description of the solid–fluid interface without increasing the computational cost. In this study, ΔX is chosen to be 1/4 of the Cartesian grid size. The level set functions for describing the solid–fluid interface in the Cartesian coordinate, $\phi_s(x, y, t)$, is derived based on the X- and Y-level set functions using

$$\phi_s(x,y) = \phi_s^0(X(x,y), Y(x,y)). \tag{16}$$

Specifically, at any time t, for an arbitrary grid point (x_0,y_0) on the fixed Cartesian mesh, its position on the reference coordinates (X_0,Y_0) can be obtained using

$$X_0 = X(x_0,y_0), \quad Y_0 = Y(x_0,y_0). \tag{17}$$

Subsequently, obtain

$$\phi_s(x_0,y_0,t) = \phi_s^0(X_0,Y_0) \tag{18}$$

In practice, Eq. 18 is implemented through interpolation of ϕ_s^0 available on the grid points defined in the reference coordinate. In this study, bi-linear interpolation is used to compute ϕ_s at grid point inside and near the solid body from ϕ_s^0 available on the grid points defined in the reference coordinate. To ensure good accuracy during interpolation, a fine grid is used for describing $\phi_s^0(X,Y)$ as mentioned earlier. The $\phi_s(x_0, y_0, t)$ constructed from this procedure represents the surface and interior of the solid body. However, it is not always a signed distance function. Consequently, re-initialization is necessary to rebuild the signed distance property for several grid points across the interface on both side of the solid surface so that the numerical δ-function given by Eq. 14 can be computed. The re-initialization will be described in Sect. 3.3.

Using the above four level set functions, we can track the movement of material points in the solid body and the evolution of the solid–fluid interfaces. Since the reference coordinates X and Y of any material point are explicit functions of its current coordinate, the inverse of the deformation gradient tensor $\mathbf{A}$ can be computed straightforwardly using finite difference

$$\mathbf{A}^{-1} = \begin{bmatrix} \frac{\partial X}{\partial x} & \frac{\partial X}{\partial y} \\ \frac{\partial Y}{\partial x} & \frac{\partial Y}{\partial y} \end{bmatrix}. \tag{19}$$

In this study, a second order central difference scheme was used to compute the derivatives in Eq. 19. Once $\mathbf{A}^{-1}$ is computed, it can be inverted and substituted into Eq. 5 to obtain the solid stress tensor τ_{elas} and subsequently the solid body force term $\mathbf{B}$ in Eq. 4. Using the solid stress tensor computed above and the level set function $\phi_s(x, y, t)$, the solid surface force can be computed using

$$\mathbf{F}_{sf} = -\frac{1}{\rho}\tau_{elas} \cdot \nabla\phi_s \delta(\phi_s). \tag{20}$$

3.3 Updating fluid–fluid interfaces and solid structures

To update the fluid–fluid interfaces, the level set function describing the interfaces between different fluids is marched in time using the velocity computed at time step n

$$\frac{\partial\phi}{\partial t} + \mathbf{u} \cdot \nabla\phi = 0. \tag{21}$$

Equation 21 was solved using a 5th-order WENO scheme (Jiang and Peng 2000). Every two steps, a re-initialization step is performed to enforce the signed distance property of the level set without changing the location of the interface through (Peng et al. 1999)

$$\frac{\partial\phi}{\partial\tau} + S(\phi_0)(|\nabla\phi| - 1) = 0, \tag{22}$$

$$\text{where} \quad S(\phi) = \frac{\phi}{\sqrt{\phi^2 + |\nabla\phi|^2(\Delta x)^2}}. \tag{23}$$

The X- and Y-level set functions used for tracking the material points in solid body are also marched in time using the fluid velocity and the 5th-order WENO scheme. However, using the fluid velocity obtained in Sect. 4 directly to march these level sets often leads to significant distortion of the level set contours in region outside of the solid body. While such distortion is physical, unless very fine grid is used, it gradually propagates into the solid body during the simulation to cause fictitious distortion of the solid body. Such fictitious distortion can compromise the accuracy of stress calculation based on these level set functions. To address this issue, we note that although the X- and Y-level set functions are defined in the entire domain, they are useful only inside the solid body and thus must be marched using the local fluid velocity. Outside the

solid body, they can in principle be marched using any velocity. In particular, one may choose the velocity to reduce the fictitious distortion inside the solid body described above. Here, we adopt a simple method to generate the fluid velocity field for marching these level sets. Inside the solid body, the velocity field used to march the level sets is the same as that computed at time step n. In the region near the solid surface (in this study, this region is defined as within 20 grid spacing from the solid surface), the velocity field is obtained by extrapolation of the velocity field inside the solid body. The method for extrapolating the velocity is summarized in Algorithm 1.

After the X- and Y-level set functions are marched in time using the velocity computed in Algorithm 1. We use the same method to extrapolate the X- and Y-level set functions in the region near the solid surface. As pointed out earlier, X- and Y-level set functions are not signed distance functions and thus are not re-initialized during the simulation. The level set function $\phi_s(x, y, t)$ describing the interface of the solid body and fluids at the new time step is constructed using new X- and Y-level functions and $\phi_s^0(X,Y)$ as described in Sect. 3.2. The level set function $\phi_s(x, y, t)$ is re-initialized using Eq. 22.

3.4 Solving fluid velocity and pressure

The unified momentum equation (Eq. 8) is solved using a two-step fractional method (Zang et al. 1994; Marella et al. 2005). In the first step, called a predictor step, an intermediate velocity $\mathbf{u}^*$ is computed based on the results at time step n without the pressure gradient term

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + [\mathbf{u}\cdot\nabla\mathbf{u}]^{n+\frac{1}{2}} = \frac{1}{2\rho^{n+\frac{1}{2}}}\nabla\cdot\mu_f^{n+\frac{1}{2}}\nabla\mathbf{u}^* + \frac{1}{2\rho^{n+\frac{1}{2}}}\nabla\cdot\mu_f^{n+\frac{1}{2}}\nabla\mathbf{u}^n + \mathcal{F}^{n+\frac{1}{2}} \tag{24}$$

$$\text{in which} \quad \mathcal{F} = \mathbf{F}_{ff} + \chi\mathbf{B} + \mathbf{F}_{sf} + \mathbf{g}. \tag{25}$$

The density $\rho^{n+\frac{1}{2}}$ and viscosity $\mu^{n+\frac{1}{2}}$ are variable in the domain due to the smearing-off methodology across an interface. They are evaluated using Eq. 12 and level set

Algorithm 1 Extrapolation of velocity in region near solid–fluid interface

Input: function $f(x,y)$ defined on fixed grid $N_x \times N_y$ and solid level set function $\phi_s(x,y)$
Output: extrapolated $f(x,y)$ for the grid points outside the solid domain
foreach *grid points* (i,j) **do**
  if $\phi_s(x_i,y_j) \le 0$ **then**
    set extrapolated flag $bExtrap(i,j)$=1;
  end
  else
    set extrapolated flag $bExtrap(i,j)$=0;
  end
  set to-be-extrapolated flag $bToBeExtrap(i,j)$=0;
end
set extrapolated points number n=1;
while $n \neq 0$ **do**
  n:=0;
  foreach *grid points* (i,j) **do**
    if *any of its existing neighbors* $(i-1,j-1)$, $(i,j-1)$, $(i+1,j-1)$, $(i-1,j)$, $(i+1,j)$, $(i-1,j+1)$, $(i,j+1)$, $(i+1,j+1)$ *has flag bExtrap=1* **then**
      set to-be-extrapolated flag $bToBeExtrap(i,j)$=1;
      foreach *grid points* (s,t) *from* $(i-4,j-4)$ *to* $(i+4,j+4)$ *within the computational domain* **do**
        if $bExtrap(s,t)$==*1* **then**
          store (s,t) in grid point set $\mathcal{G}$ for extrapolation;
        end
      end
      assume that the $f(x,y)$ is linear near (i,j), i.e. $f = C_1x + C_2y + C_3$;
      compute C_k (k=1,2,3) using all (x_s,y_t) and $f(x_s,y_t)$ which $(s,t) \in \mathcal{G}$ by linear least square method;
      compute $f(i,j)$ using C_k (k=1,2,3);
      n:=n+1;
    end
  end
  foreach *grid points* (i,j) **do**
    if $bToBeExtrap(i,j)$==*1* **then**
      $bToBeExtrap(i,j)$=0;
      $bExtrap(i,j)$=1;
    end
  end
end

functions at $n+\frac{1}{2}$ time step. A level set function ϕ at time step $n+\frac{1}{2}$ is computed using the arithmetic mean value

$$\phi^{n+\frac{1}{2}} = \frac{\phi^{n+1} + \phi^n}{2}. \tag{26}$$

Note that ϕ^{n+1} is computed in Sect. 3.3. The advection term $\mathbf{u}\cdot\nabla\mathbf{u}$ at time step $n+\frac{1}{2}$ is calculated using the Godunov method [for details see reference Colella (1985), Sussman et al. (1999)]. Note that this term is not related to time step Δt. In the second step, a corrector step, the velocity is corrected according to the pressure gradient

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\frac{1}{\rho^{n+\frac{1}{2}}}\nabla p^{n+\frac{1}{2}} \tag{27}$$

in which, the pressure $p^{n+\frac{1}{2}}$ is calculated using a pressure Poisson equation derived from the continuity equation (Eq. 7).

$$\nabla\cdot\frac{1}{\rho^{n+\frac{1}{2}}}\nabla p^{n+\frac{1}{2}} = \frac{\nabla\cdot\mathbf{u}^*}{\Delta t}. \tag{28}$$

The time step Δt is determined by CFL conditions (Zhao et al. 2008; Sussman et al. 1999)

$$\Delta t = \frac{1}{2}\min\left(\frac{\Delta x}{\mathbf{u}^n}, \sqrt{\frac{(\rho_1+\rho_2)\Delta x^3}{8\pi\sigma}}, \frac{\Delta x}{3}\sqrt{\frac{\rho_s}{\mu_s}}, \sqrt{\frac{2\Delta x}{\mathcal{F}^n}}\right) \tag{29}$$

in which, ρ_1 and ρ_2 are the density of fluid 1 and fluid 2 in Fig. 1, respectively; ρ_s is the solid density and gravitational force $\mathbf{g}$ is included in $\mathcal{F}$ (Eq. 25).

The SLS method is computationally efficient. Compared to the ZFM method, the extra computational cost includes marching the *X*/*Y*-level sets, constructing and re-initializing the surface level set $\phi_s(x, y, t)$, and computing the velocity field used to march the *X*/*Y*-level sets outside solid body by extrapolation. The computational cost of these operations is generally smaller compared to other steps in the SLS algorithm (e.g., solving Eq. 28). Consequently, the added computational cost is moderate. This was confirmed in the the numerical examples in Sect. 4.2.

4 Numerical results

We developed a code based on the algorithm described in the previous sections. The level set portion of our code made use of the open source level set library developed by Chu and Prodanovic (2011). To demonstrate the effectiveness of the code and algorithm, we have studied the fluid flow and solid structure evolution in a lid-driven cavity with an elastic wall and in the coalescence of droplet loaded with a soft elastic sphere with a pool of fluids. The first example is studied in the two-dimensional space and the second example is studied in an asymmetrical configuration.

4.1 A lid-driven cavity with an elastic wall

We choose to study this problem since it has been studied separately by Dunne (2006) and Zhao et al. (2008). A 2×2 cavity is divided into two parts: lower 0.5 as a neo-Hookean wall and the upper part as a Newtonian fluid. The flow and elastic properties are

$$\rho_f = \rho_s = 1, \quad \mu_f = 0.2, \quad \mu_s = 0.2. \tag{30}$$

Initially, the fluid and solid are both stationary; and the elastic wall is in its resting shape as a rectangle. At time $t = 0$, the upper lid begins to move to drive the flow, the velocity profile of the upper lid is

$$u = 0.5\begin{cases} \sin^2(\pi x/0.6), & 0 \le x \le 0.3, \\ 1, & 0.3 < x < 1.7, \\ \sin^2(\pi(x-2)/0.6), & 1.7 \le x \le 2 \end{cases} \tag{31}$$

Note that in this simulation, to compare the results with previous researchers, the convection term in N–S equations was turned off. The domain is discretized into a uniform 101×101 grid. The solid interface function $\phi_s^0(X, Y)$ is defined on a finer grid in which $\Delta X = \Delta Y = 0.005$. At $t = 69.6$, the maximum velocity $|u|$ inside the wall is smaller than 3×10^{-5}. Figure 3 shows the results using the method presented in Sect. 3 compared with that given by (Zhao et al. 2008). We observe that the surface of the wall matches well with that predicted by Zhao et al. The inner contours of the solid wall match well with the deformed mesh in Zhao et al.'s results as well. The volume loss happens in many simulations using level sets in the present case, the relative volume change of the elastic wall is less than 0.08%.

To assess the convergence rate of the SLS method, we computed the fluid velocities on a horizontal line $(x: 0 \to 2, y = 1.25)$ at $t = 20$ using different grid spacings, and compared them to the results obtained using a 400×400 grid to estimate error. Figure 4 shows that the error in *x*-direction velocity is $O(\Delta x)^{1.6}$ in an L_∞ norm, and $O(\Delta x)^{1.7}$ in an L_2 norm, and very similar results are obtained for the velocity in the *y*-direction. These results are consistent with the facts that the SLS method shares the same framework for computing FSI with the ZFM method and the ZFM method generally has a spatial convergence rate between first order and second order (Zhao et al. 2008). We also studied the convergence of the SLS method in the time domain and a first-order temporal convergence rate, consistent with the time marching scheme shown in Sect. 4, was obtained. The results are not shown here for brevity.

4.2 Interaction of solid laden droplet with a pool

To further test the effectiveness of the SLS method, in this problem, a water droplet loaded with a neo-Hooken elastic

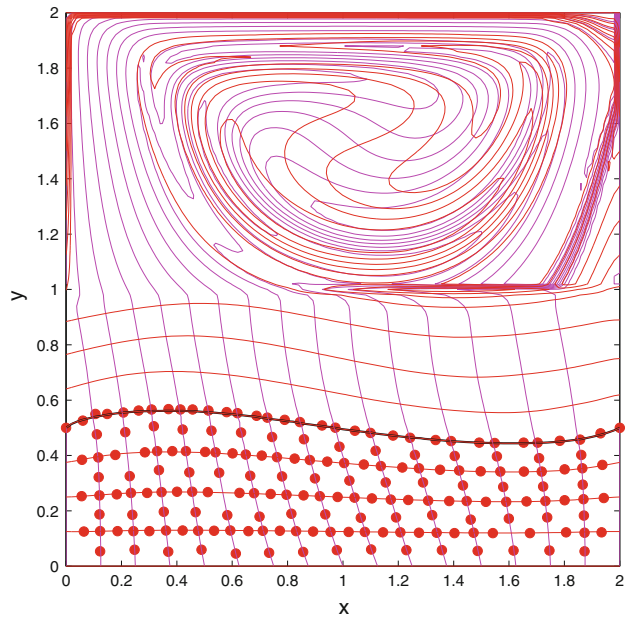

Fig. 3 Steady state deformation of the elastic wall in a lid-driven cavity: the *dots* show the results of Zhao et al. (2008). *X*-/*Y*-functions are shown as two sets of contours. Note that the contours of *X*-/*Y*-level set functions are significantly distorted by the fluid circulation in the upper portion of the cavity. The field extension algorithm to the fluid velocity and *X*-/*Y*-level sets described in Sect. 3 is necessary to avoid the fictitious distortion of the elastic wall

sphere (shear modulus: 10 kPa) is placed on top of a deep pool of water as shown in Fig. 5. The space above the pool is occupied by air. The material properties of the fluids are taken as

$$\rho_{\text{water}} = 998\,\text{kg/m}^3, \quad \rho_{\text{air}} = 1.226\,\text{kg/m}^3$$
$$\mu_{\text{water}} = 1\text{cP}, \quad \mu_{\text{air}} = 0.0178\,\text{cP}, \quad \sigma_{\text{water-air}} = 73\,\text{mN/m} \tag{32}$$

At $t = 0$, the velocity of all fluids is zero, and the solid sphere is located at the center of the droplet and is in its resting shape. At $t > 0$, the droplet is driven downward violently by surface tension to merge with the pool fluids. In the absence of solid sphere, the droplet coalesces with the pool fluids and small daughter droplet is ejected from the central jet formed during the coalescence process, and this phenomenon is often termed partial absorption in the literature (Blanchette and Bigioni 2006, 2009). When a soft solid sphere is loaded inside the droplet, the coalescence process is significantly changed by the solid body. Meanwhile, the solid sphere can deform significantly due to the imbalance of pressure on different parts of its surface. The strong coupling between the fluid flow and solid body evolution makes this a challenging problem and is thus an ideal test problem for examining the effectiveness of the proposed method.

We studied the above problem under the axisymmetric assumption. The entire system (cf. Fig. 5 for geometrical parameters) is discretized into a uniform grid with spacing in the radial and vertical direction both equal to 0.4 μm. Our simulations with similar problems indicated that a grid spacing smaller than 0.5 μm, i.e., sixty grid points across the un-deformed cell, is sufficient to obtain grid-independent results. The half width of the narrow band across the interface, over which the fluid properties are smeared off, is 0.8 μm. The lower and upper boundaries of the simulation box are resolved using no-slip boundary condition. A velocity extrapolation boundary condition (Pan and Law 2007) was applied on the right boundary ($r = 80$ μm).

We solve this problem using both the ZFM method and the SLS method. The ZFM method is 37% faster than the SLS method. The slightly lower computational efficiency of the SLS method is mostly due to the fact that the SLSs functions are tracked inside the entire simulation domain, while the solid body is tracked using the Lagrangian mesh overlapped on part of the Cartesian grid in the ZFM method. One can reduce the computational cost involved in maintaining and marching the SLSs by exploiting the fact that these level sets are useful only inside the solid body *and* adjacent to the solid body surface. Consequently, they

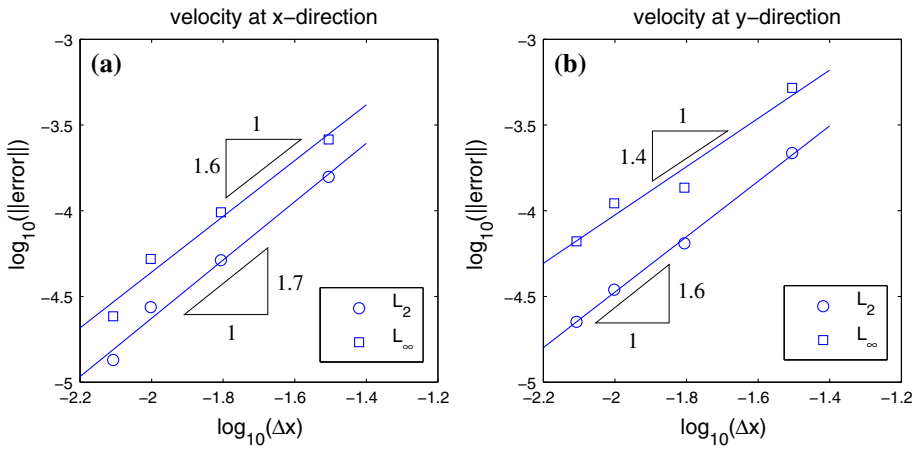

Fig. 4 Convergence rate of the fluid velocity in *x*-direction (**a**) and *y*-direction (**b**) on a horizontal line ($x : 0 \to 2, y = 1.25$) in the cavity driven flow problem

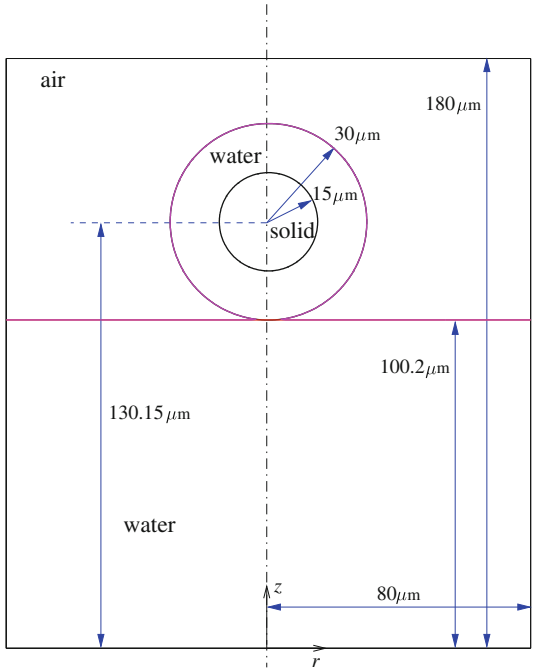

Fig. 5 A schematic of the droplet-sphere-pool system studied in the text

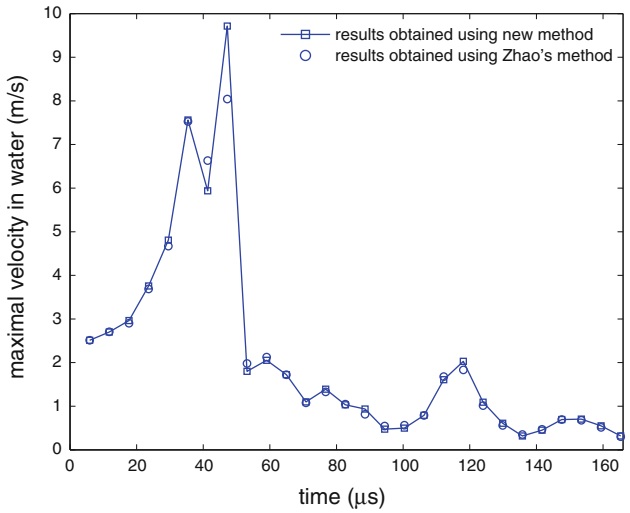

Fig. 6 Evolution of maximal velocity magnitude in water

can be handled using the narrow band level set scheme or its extensions (Adalsteinsson and Sethian 1995), which will reduce the computational cost considerably.

Figure 6 compares the maximum magnitude of water velocity inside the system obtained from simulations using the two different methods. We observe that although the initial fluid velocity is zero, surface tension induces very strong flow inside the water but such flow eventually diminishes due to viscous dissipation. Figure 6 shows that the two methods give nearly the same results in most time period. The volume change of the solid cell relative to its original value is −0.3%. Figure 7 compares the evolution of the water–air interfaces and the surface of the solid sphere during the droplet–pool coalescence. First, the solid sphere is pinched in the radial direction while being driven downwards by the capillary pressure ($t < 23.38\mu_s$). Driven by elastic forces, the solid sphere subsequently recovers to nearly its resting shape ($23.38\mu_s < t < 32.82\mu_s$). However, the collapse of the central column of water droplet back to the pool leads to compression of the top portion of the solid sphere and the further downward migration of the solid sphere ($32.82\mu_s < t < 42.27\mu_s$). The solid sphere subsequently recovers toward its resting shape and move downward due to recirculation of water inside the pool ($t > 42.27\mu_s$, the water recirculation is not shown in Fig. 7 for clarity). The agreement of the water–air interface and solid body shape predicted by the ZFM and SLS methods is very good. In particular, the shape of the solid sphere are essentially the same. The only difference of the solid sphere evolution is that the solid sphere migrates slightly shallower into the pool at $t = 67.07\mu_s$ when the SLS method is used. The small difference at this late time can be due to the fact that the velocities used to march the solid body and the techniques used to march the solid body are slightly different in the two methods. In the ZFM method, the velocity at any Lagrangian mesh point is interpolated from the fixed Cartesian mesh, and the solid body is convected using interpolated velocity. In the SLS method, the velocity used to march the solid body is not interpolated, and Eq. 21 was used to march the SLSs. In the ZFM method, the accuracy of marching solid body is limited by the accuracy of velocity interpolation step. In the SLS method, the accuracy of marching solid body is limited by the accuracy of the level set marching. We expect the SLS method to be slightly less accurate compared to the ZFM method when the same fixed Cartesian grid is used because artificial smearing off of level set is a known issue in level set methods. However, given the very small difference observed in Fig. 7, the accuracy of SLS method is very good. To further improve the SLS method, one may adopt other more accurate methods for constructing or marching level sets (Enright et al. 2002, 2005), but this is beyond the scope of this study.

5 Conclusions

We developed a full Eulerian scheme for solving the interactions between fluid flow and solid bodies. The scheme is based on a unified framework for solving fluid–solid body interactions proposed recently, in which the solid body is convected by fluids and the fluid/solid dynamics are modeled using a unified momentum equation

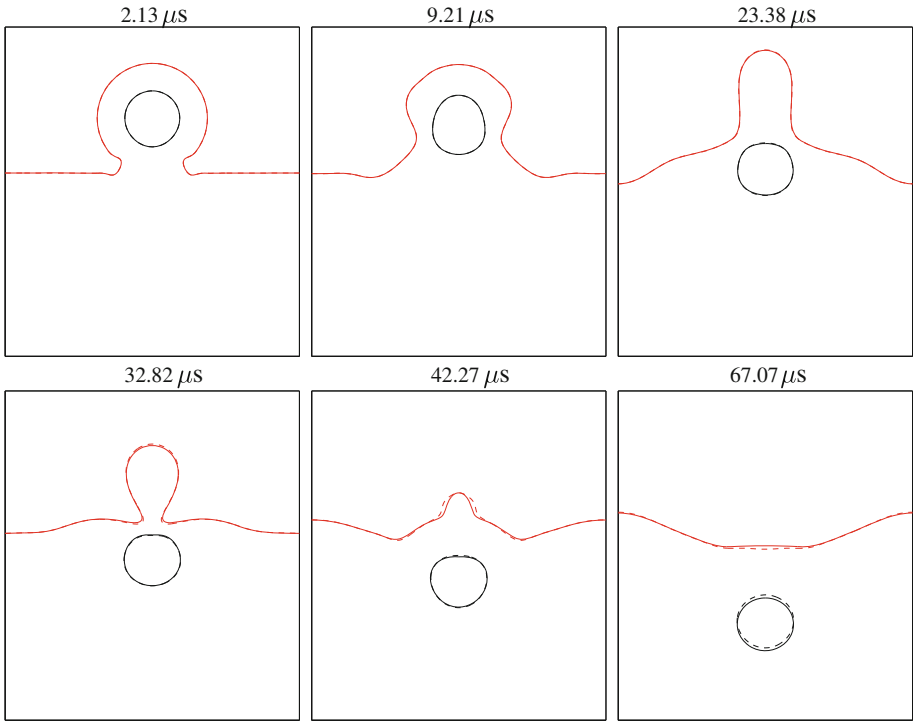

Fig. 7 Evolution of the water–air interfaces and the solid sphere during the coalescence of a water droplet laden with an elastic sphere with a pool of water. *Solid lines* are the surface contours computed using the ZFM's algorithm. The *dashed lines* are those obtained using the SLS method described in Sect. 3

that is valid throughout the entire simulation domain. The coupling between solid motion and fluid flow is modeled by introducing a solid body force and solid–fluid interfacial force into the unified momentum equation. In the SLS method, the solid body is captured on a fixed Cartesian grid instead of using a Lagrangian mesh. Specifically, we introduce reference coordinates (X, Y) to provide the coordinates of material points of the solid body at a reference state. Two X- and Y-level set functions are used to provide the mapping between the current Cartesian coordinates of a material point in the solid body to its coordinates in the reference coordinates. Two level set functions, $\phi_s^0(X,Y)$ and $\phi_s(x, y)$, are introduced for describing the interfaces between the solid body and fluids in the reference coordinates and in the Cartesian coordinates. The X- and Y-level set functions are convected by fluid velocity, but a special extrapolation scheme is designed to obtain the fluid velocity prevent the fictitious distortion of the solid body. We demonstrate the effectiveness of the SLS method by solving the coupled fluid/structure dynamics in a cavity driven flow featuring elastic walls and in the coalescence of a water droplet loaded with an elastic sphere with a pool of water.

The SLS method presented here can be extended to solve three dimensional problems straightforwardly since the basic idea of this method, i.e., using level sets to track the materials points inside solid body, is easily adopted for three dimensional systems drawing on the strength of the level set method in handling complex shapes in three dimensional spaces. The new method inherits the computational efficiency of the ZFM method. By eliminating the Lagrangian mesh for tracking solid bodies, the proposed method significantly reduces the complexity of FSI simulations. In addition, several challenges requiring special treatment in methods simultaneously using a fixed Cartesian mesh and a moving Lagrangian mesh, e.g., discontinuity of stress across Lagrangian mesh and artificial compression/dilation of solid body, are naturally resolved in this method. The method is particularly suitable for simulating the interactions between soft solid objects and fluids flows. This method is thus useful for studying a wide range of problems in microfluidics, e.g., manipulation of cells in confined channels and ink-jet printing of biological samples.

Acknowledgements The authors gratefully acknowledge support from NSF under grant No. CBET-0936235. The authors thank the Clemson-CCIT office for generous allocation of computer time on the Palmetto cluster.

References

Adalsteinsson D, Sethian JA (1995) A fast level set method for propagating interfaces. J Comput Phys 118:269–277

Blanchette P, Bigioni T (2006) Partial coalescence of drops at liquid interfaces. Nat Phys 2:254–257

Blanchette P, Bigioni T (2009) Dynamics of drop coalescence at fluid interfaces. J Fluid Mech 620:333–352

Brackbill J, Kothe D, Zemach C (1992) A continuum method for modeling surface-tension. J Comput Phys 100(2):335–354

Chu K, Prodanovic M (2011) Homepage of level set method library (lsmlib). http://ktchu.serendipityresearch.org/software/lsmlib/index.html. Accessed 22 Feb 2011

Colella P (1985) A direct Eulerian MUSCL scheme for gas-dynamics. SIAM J Sci Stat Comput 6(1):104–117

Dunne T (2006) An Eulerian approach to fluid–structure interaction and goal-oriented mesh adaptation. Int J Numer Methods Fluids 51(9–10):1017–1039

Enright D, Fedkiw R, Ferziger J, Mitchell I (2002) A hybrid particle level set method for improved interface capturing. J Comput Phys 183(1):83–116

Enright D, Losasso F, Fedkiw R (2005) A fast and accurate semi-Lagrangian particle level set method. Comput Struct 83(6–7): 479–490

Guendelman E, Selle A, Losasso F, Fedkiw R (2005) Coupling water and smoke to thin deformable and rigid shells. ACM Trans Gr 24:973–981

Jiang G, Peng D (2000) Weighted ENO schemes for Hamilton-Jacobi equations. SIAM J Sci Comput 21(6):2126–2143

Leveque R, Li Z (1994) The immersed interface method for elliptic-equations with discontinuous coefficients and singular sources. SIAM J Numer Anal 31(4):1019–1044

Marella S, Krishnan S, Liu H, Udaykumar H (2005) Sharp interface Cartesian grid method I: an easily implemented technique for 3D moving boundary computations. J Comput Phys 210(1):1–31

Osher S (1993) A level set formulation for the solution of the Dirichlet problem for Hamilton-Jacobi equations. SIAM J Math Anal 24(5):1145–1152

Osher S, Fedkiw R (2002) Level set methods and dynamic implicit surfaces. Springer, New York, pp 25–39

Pan K, Law C (2007) Dynamics of droplet-film collision. J Fluid Mech 587:1–22

Peng D, Merriman B, Osher S, Zhao H, Kang M (1999) A PDE-based fast local level set method. J Comput Phys 155(2):410–438

Peskin C (1972) Flow patterns around heart valves—numerical method. J Comput Phys 10(2):252–271

Sethian JA (1991) The collapse of a dumbbell moving under its mean curvature. In: Concus P, Finn R, Hoffman D (eds) Geometric analysis and computer graphics. Mathematical Sciences Research Institute Publications, Springer-Verlag, New York

Sussman M, Almgren A, Bell J, Colella P, Howell L, Welcome M (1999) An adaptive level set approach for incompressible two-phase flows. J Comput Phys 148(1):81–124

Sussman M, Smereka P, Osher S (1994) A level set approach for computing solutions to incompressible 2-phase flow. J Comput Phys 114(1):146–159

Zang Y, Street R, Koseff J (1994) A non-staggered grid, fractional step method for time-dependent incompressible Navier-Stokes equations in curvilinear coordinates. J Comput Phys 114(1): 18–33

Zhao H, Freund JB, Moser RD (2008) A fixed-mesh method for incompressible flow-structure systems with finite solid deformations. J Comput Phys 227(6):3114–3140

Zienkiewicz O, Zhu J (1992) The superconvergent patch recovery and a posteriori error estimates. Part 1. The recovery technique. Int J Numer Methods Eng 33(7):1331–1364