Journal of Computational Physics **169,** 427–462 (2001)
doi:10.1006/jcph.2000.6592, available online at http://www.idealibrary.com on IDEAL®

Direct Numerical Simulations of Fluid–Solid Systems Using the Arbitrary Lagrangian–Eulerian Technique

Howard H. Hu,* N. A. Patankar,† and M. Y. Zhu*

*Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, Pennsylvania 19104-6315; and †Department of Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, Minnesota 55455

E-mail: hhu@seas.upenn.edu, patankar@aem.umn.edu, mzhu@seas.upenn.edu

Received March 11, 2000; revised July 17, 2000

Since the initial publication of Hu *et al.* (1992, *Theor. Comput. Fluid Dyn.* **3**, 285), the numerical method developed for direct simulations of fluid–solid systems using the arbitrary Lagrangian–Eulerian (ALE) technique has undergone continuous modifications. Some of the modifications were described in H. H. Hu (1996, *Int. J. Multiphase Flow* **22**, 335). In this paper, we will present the most up-to-date implementation of the method and the results of several benchmark test problems.

Key Words: arbitrary Lagrangian–Eulerian technique; finite-element methods; fluid–solid system; particulate flows; particle collision.

1. INTRODUCTION

Numerical simulations of fluid–solid two-phase flow systems can be classified into different categories. The most common approach is to use the continuum theory that views the solid and the fluid as interpenetrating mixtures, each being governed by conservation laws, either postulated or derived by averaging (see, for example, Ishii [39], Zhang and Prosperetti [69], Gidaspow [23], Fan and Zhu [15], and Drew and Passman [13]). This Eulerian continuum approach results in field equations for the flow properties for all phases in the system. It also leads to unknown terms representing the interactions between the phases. These terms must be modeled to close the description of the system. The nature of the detailed interactions between the solid and the fluid cannot be understood from the application of mixture theories alone. However, once these interaction terms are determined, the Eulerian continuum approach is most efficient and has been widely used in multiphase flow simulations.

0021-9991/01 $35.00

A second approach in multiphase flow simulations is Lagrangian particle tracking, or Lagrangian numerical simulation (LNS). This approach provides a direct description of the particulate flow by tracking the motion of individual particles. In LNS, the fluid satisfies the continuum equations that are solved on a fixed field in the usual Eulerian way. The particle motion is governed by Newton's second law for rigid particles with empirical forms of the hydrodynamic forces. When the particle concentration is low, models with one-way coupling are often used. In these, the motion of the particles is determined by the fluid flow, but the particle motion does not influence the fluid flow. In models with two-way coupling, a momentum exchange term could be introduced into the fluid equations to take into account the effects of the particle motion on the fluid flow (see McLaughlin [52]). Andrews and O'Rourke [3] and Snider *et al.* [66] introduce a scheme that considers the particle phase both as a continuum and as a discrete phase. In this way, they can track the motion of the particles and at the same time model the interparticle stress. This results in a computational method for multiphase flows that can handle particulate loading ranging from dense to dilute and particles of different sizes and materials. As in the Eulerian continuum approach, the LNS requires the empirical forms of the hydrodynamic forces acting on the solid particles. These forces are normally determined from certain dilute conditions that do not account, for particle–particle or particle–boundary interactions.

The clusters and anisotropic microstructures observed in fluid–solid systems are the results of solid particle migrations produced by particle–particle and particle–wall interactions. These local rearrangement mechanisms are mediated by things such as hydrodynamic forces and moments acting on the solid particles, wake interactions, and vortex shedding. The third type of approach to simulating the motion of both the fluid and the solid particles is termed the direct numerical simulation (DNS). In DNS, the hydrodynamic forces acting on the solid particles are directly computed from the fluid flow, and the motion of the fluid flow and solid particles are fully coupled. The DNS of the exact particle motion in a fluid may be the only theoretical tool capable of studying the nonlinear and geometrically complicated phenomena of particle–particle and particle–wall interactions.

In DNS, it is possible to simplify the flow description considerably by ignoring the viscous effects completely (inviscid potential flow) or by ignoring the fluid inertia completely (Stokes flow). Potential flow simulations (see, Sangani and Didwania [61] and Sangani and Prosperetti [62]) do lead to cross-stream alignment of particles in fluidized systems, but the wakes and the other nonlinear mechanisms for the fundamental arrangement of particles in a fluidized suspension are absent. Brady and co-workers (see Brady and Bossis [9] and Brady [8]) have developed numerical techniques (Stokesian dynamics) for simulating the motion of a large number of particles in Stokes flows. These simulations are appropriate for colloidal suspensions in the limit of zero particle Reynolds number.

For simulations of fluid–solid systems at finite Reynolds numbers, a number of numerical methods have been developed in recent years. The first method is termed the ALE (arbitrary Lagrangian–Eulerian) particle mover. The ALE particle mover uses a technique based on a combined formulation of the fluid and particle momentum equations, together with an arbitrary Lagrangian–Eulerian (ALE) moving, unstructured, finite-element mesh technique to deal with the movement of the particles. It was first developed by Hu and co-workers [28, 30]. The method has been used to solve particle motions in both Newtonian and viscoelastic fluids under two-dimensional and three-dimensional flow geometries. It also handles particles of different sizes, shapes, and materials. Hu *et al.* [30] first simulated two-dimensional sedimentation of circular and elliptic cylinders confined in a channel.

Feng *et al.* [16, 17] studied the motion and interaction of circular and elliptical particles in sedimenting, Couette, and Poiseuille flows of a Newtonian fluid. Huang *et al.* [33] examined the turning couples on an elliptic particle settling in a channel. Hu [27] studied the rotation of a circular cylinder settling close to a solid wall. Feng *et al.* [18] analyzed the mechanisms for the lifting of flying capsules in pipelines. Later Hu [28] reported the results of two-dimensional direct numerical simulation of the motion of a large number of circular particles in a Newtonian fluid at particle Reynolds numbers around 100. Feng *et al.* [19] also studied the sedimentation of circular particles in an Oldroyd-B fluid. Later, Huang *et al.* [32] examined the motion of particles in Couette and Poiseuille flows of viscoelastic and shear-thinning fluids. Huang *et al.* [34] investigated the effects of viscoelasticity and shear thinning on the stable orientation of ellipses falling in a viscoelastic fluid. Patankar [55] investigated the rheology of suspensions of particles in both Newtonian and viscoelastic fluids. Zhu [70] studied extensively the migration and interaction of spheres in various three-dimensional flows.

Another method for solving problems with moving boundaries uses space–time finite-element methods (see Hughes and Hulbert [36], Tezduyar *et al.* [67, 68], and Hansbo [26]). In the space–time approach, along with the spatial coordinates, the temporal coordinate is discretized using finite-element methods. The deformation of the spatial domain with time is reflected simply in the deformation of the mesh in the temporal coordinate. A space–time finite-element scheme for solving fluid–particle systems was developed by Johnson [40] and Johnson and Tezduyar [41, 42]. Using this technique, Johnson and Tezduyar [43] are able to simulate the sedimentation of 1,000 spheres in a Newtonian fluid at a Reynolds number of 10. The advantage of the space–time finite-element method is its generality. One can frame the ALE finite-element scheme as a special case of the space–time method, as discussed by Hansbo [26] and Behr and Tezduyar [4].

The third numerical method used to simulate fluid–solid systems is termed the DLM (distributed-Lagrange-multiplier) particle mover. The basic idea of the DLM particle mover is to extend a problem on a time-dependent geometrically complex domain to a stationary, larger, but simpler domain (the "fictitious domain"). On this fictitious domain, the constraints of rigid-body motion of the particles are enforced using a distributed Lagrange multiplier, which represents the additional body force needed to maintain the rigid-body motion inside the particle, much like the pressure in incompressible fluid flows is used to maintain the constraint of incompressibility. The DLM particle mover was recently introduced by Glowinski *et al.* [24] and has been extended to handle viscoelastic fluids (Singh *et al.* [65]). It has been used to simulate the sedimentation and fluidization of over 1,000 spheres in a Newtonian fluid.

In recent years, the lattice Boltzmann method (LBM) has been developed into an alternative and promising numerical scheme for simulating fluid flows. Unlike the conventional numerical schemes based on discretizations of macroscopic continuum equations, LBM is based on microscopic models and mesoscopic kinetic equations. The fundamental idea of the LBM is to construct simplified kinetic models that incorporate the essential physics of the microscopic or mesoscopic processes so that the macroscopic-averaged properties obey the desired macroscopic equations; see the recent review article by Chen and Doolen [12]. The LBM has been adapted to simulate the motion of solid particles in a Newtonian fluid. Most of the work in this area was done by Ladd [46–48], Behrend [5], Aidun *et al.* [1, 2], and Qi [59]. Their schemes are based on a fully explicit scheme, where the hydrodynamic forces and moments acting on solid particles are first calculated from lattice Boltzmann

simulation, and the motion of the solid particles is then determined from these forces and moments using Newton's second law. The LBM simulations can be easily performed on parallel computers. The computational cost for simulating particle motion scales linearly with the number of the particles. Using the lattice Boltzmann technique, Ladd [48] simulated up to 32,000 three-dimensional spheres suspended in a fluid.

The four numerical techniques mentioned above are not the only ones available for direct numerical simulations of multiphase flows at finite particle Reynolds numbers. For example, the front-tracking/finite-difference method developed by Tryggvason's group is very powerful in simulating the motion of a large number of deformable drops and bubbles (see, for example, Esmaeeli and Tryggvason [14]).

Since the initial publication of Hu *et al.* [30], the ALE particle mover has undergone continuous modifications. Some of them were documented in Hu [28]. In this paper, we will present the most up-to-date implementation of the method and the results of several benchmark test problems. The governing equations describing the motion of both the fluid and the solid are laid down in Section 2. A simple fully explicit scheme is presented in Section 3 and its stability is also discussed. Section 4 derives the combined formulation for the fluid–solid system. An ALE mesh movement scheme is presented in Section 5. The temporal discretization using a second-order finite-difference scheme and the spatial discretization using a finite-element scheme of the governing equations are described in Sections 6 and 7, respectively. The topics of automatic finite-element mesh generation and remeshing criteria are discussed in Section 8. Two schemes of flow field projection from one mesh onto another mesh are described in Section 9. The Section 10 outlines the procedure for the explicit–implicit solution scheme for fluid–solid systems. The models of particle collision are then discussed in Section 11. Finally, the results of five benchmark test problems are presented and compared with ones from the literature.

2. GOVERNING EQUATIONS

In a system of solid rigid particles suspended in a fluid, the motion of the fluid and that of the solid particles are fully coupled. The motion of the particles is determined by the hydrodynamic forces and torques imposed on them by the surrounding fluid. However, the fluid motion is strongly influenced or sometimes even driven by the particle motion—for example, in the case of sedimentation. In direct numerical simulations, we want to calculate the motion of both the fluid and the individual solid particles, without using empirical correlations for the hydrodynamic forces acting on the particles. In most applications, the Reynolds number of the flow based on the particle size is usually not small; thus the inertias of the fluid and the solid have to be included in the model. In this section, we shall lay down the governing equations describing the motion of both the fluid and the solid.

We shall consider the motion of N rigid solid objects (particles) in an incompressible fluid. Denote $\Omega_0(t)$ as the domain occupied by the fluid at a given time instant $t \in [0, T]$, and denote $\Omega_i(t)$ as the domain occupied by the ith particle ($i = 1, 2, \ldots, N$). The boundaries of $\Omega_0(t)$ and $\Omega_i(t)$ are denoted as $\partial\Omega_0(t)$ and $\partial\Omega_i(t)$, respectively.

The governing equations for the fluid motion in $\Omega_0(t)$ are the conservation of mass,

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

and the conservation of momentum,

$$\rho_f \frac{D\mathbf{u}}{Dt} = \rho_f \mathbf{f} + \nabla \cdot \sigma, \tag{2}$$

where $\mathbf{u}$ is the velocity vector; ρ_f is the density of the fluid; $\mathbf{f}$ is the body force per unit mass, which could be the gravitational acceleration; the material derivative of the velocity is given by

$$\frac{D\mathbf{u}}{Dt} = \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u}; \tag{3}$$

and σ is the stress tensor. For a Newtonian fluid, the stress tensor is given by the simple constitutive relation

$$\sigma = -p\mathbf{1} + 2\eta \mathbf{D}[\mathbf{u}] \quad \text{and} \quad \mathbf{D}[\mathbf{u}] = \frac{1}{2}[\nabla \mathbf{u} + (\nabla \mathbf{u})^T], \tag{4}$$

with p being the pressure and η being the fluid viscosity. For a viscoelastic fluid, the stress tensor may be expressed as

$$\sigma = -p\mathbf{I} + 2\eta_2 \mathbf{D}[\mathbf{u}] + \tau_p, \tag{5}$$

where τ_p is the "polymer contribution" to the stress and may be governed by a constitutive equation such as an Oldroyd-B fluid model [6],

$$\lambda\left(\frac{D\tau_p}{Dt} - \nabla \mathbf{u}^T \cdot \tau_p - \tau_p \cdot \nabla \mathbf{u}\right) + \tau_p = 2\eta_1 \mathbf{D}[\mathbf{u}]. \tag{6}$$

The parameter λ is the fluid relaxation time; $\eta = \eta_1 + \eta_2$ is the fluid viscosity. The Newtonian fluid can be considered a special case with $\eta_2 = \eta$ and $\tau_p = \mathbf{0}$. In general, the viscosity and relaxation time of the fluid are functions of the local shear rate of the flow; for example, viscosity laws such as Bird–Carreau, power-law, Bingham, and Herschel–Bulkley may apply.

The rigid particles satisfy Newton's second law for the translational motion,

$$m_i \frac{d\mathbf{V}_i}{dt} = \mathbf{G}_i + \mathbf{F}_i = \mathbf{G}_i - \int_{\partial\Omega_i(t)} \sigma \cdot \mathbf{n}\, dS, \tag{7}$$

and the Euler equations for the rotation,

$$\frac{d}{dt}(\mathbf{I}_i \omega_i) = \mathbf{I}_i \frac{d\omega_i}{dt} + \omega_i \times \mathbf{I}_i \omega_i = \mathbf{T}_i = -\int_{\partial\Omega_i(t)} (\mathbf{x} - \mathbf{X}_i) \times (\sigma \cdot \mathbf{n})\, dS, \tag{8}$$

where the index i (=1, 2, ..., N) represents different solid particles; m_i is the mass, and $\mathbf{I}_i$ is the moment of inertia matrix of the ith particle; $\mathbf{V}_i$ and ω_i are the translational and angular velocities of the particle, respectively; $\mathbf{G}_i$ is the body force exerted by external fields such as the gravity; and the hydrodynamic force $\mathbf{F}_i$ and moment $\mathbf{T}_i$ acting on the particle are obtained by integrating the fluid stress over the particle surface, as noted in (7) and (8), with $\mathbf{n}$ being the unit normal vector on the surface of the particle pointing into the particle. The centroid, $\mathbf{X}_i$, and the orientation (for example, the three Euler angles), Θ_i, of the particle are updated according to

$$\frac{d\mathbf{X}_i}{dt} = \mathbf{V}_i, \tag{9}$$

and

$$\frac{d\Theta_i}{dt} = \omega_i, \tag{10}$$

respectively.

The boundary of the fluid domain $\Omega_0(t)$ can be decomposed into three nonoverlapping sections: $(\partial\Omega)_u$, $(\partial\Omega)_\sigma$, and $\cup\partial\Omega_i$. On these boundary sections three types of boundary conditions are imposed,

$$\mathbf{u} = \mathbf{u}_g, \qquad \text{on } (\partial\Omega)_u \tag{11}$$

$$\sigma \cdot \mathbf{n} = 0, \qquad \text{on } (\partial\Omega)_\sigma \tag{12}$$

$$\mathbf{u} = \mathbf{V}_i + \omega_i \times (\mathbf{x} - \mathbf{X}_i), \qquad \text{for } \mathbf{x} \in \partial\Omega_i(t), \tag{13}$$

where $\mathbf{u}_g$ is the prescribed velocity. Expression (13) represents the no-slip condition on the particle surface. The boundary conditions for the elastic stress are normally imposed on the inflow boundary $(\partial\Omega)_{\text{in}}$,

$$\tau_p = \tau_{\text{in}}, \qquad \text{on } (\partial\Omega)_{\text{in}} \tag{14}$$

where the stress, τ_{in}, is prescribed.

The initial conditions for the flow field and the particle variables are

$$\mathbf{u} = \mathbf{u}_0 \quad \text{and} \quad \tau_p = \tau_0, \qquad \text{in } \Omega_0(0), \tag{15}$$

and

$$\mathbf{X}_i(0) = \mathbf{X}_i^o, \quad \Theta_i(0) = \Theta_i^o, \quad \mathbf{V}_i(0) = \mathbf{V}_i^o, \quad \text{and} \quad \omega_i(0) = \omega_i^o, \quad \text{for } i = 1, 2, \ldots, N, \tag{16}$$

where the initial velocity, $\mathbf{u}_0$, should be divergence-free.

3. FULLY EXPLICIT SCHEME AND ITS STABILITY

A simple approach to numerically simulating fluid–particle motion is to decouple the motion of the fluid and solid at each time step. A fully explicit scheme is described below.

SCHEME 1. Fully Explicit Scheme.

Initialization: $t_0 = 0$, $n = 0$ (index for time step).
  Initialize $\mathbf{u}(\mathbf{x}, t_0)$ and $\mathbf{V}_i(t_0)$, $\omega_i(t_0)$ for $i = 1, 2, \ldots, N$.
Do $n = 1, 2, \ldots, M$ (total number of time steps)
  Select time step Δt_n: $t_n = t_{n-1} + \Delta t_n$.
  Using the particle velocities $\mathbf{V}_i(t_{n-1})$ and $\omega_i(t_{n-1})$ as the boundary conditions, solve for the flow field $\mathbf{u}(\mathbf{x}, t_n)$ and $p(\mathbf{x}, t_n)$ by a numerical method (traditional CFD).
  Using the flow field $\mathbf{u}(\mathbf{x}, t_n)$ and $p(\mathbf{x}, t_n)$, calculate the hydrodynamic forces $\mathbf{F}_i(t_n)$ and moments $\mathbf{T}_i(t_n)$ acting on the particles.
  Using the forces $\mathbf{F}_i(t_n)$ and moments $\mathbf{T}_i(t_n)$, update the particle velocities $\mathbf{V}_i(t_n)$ and $\omega_i(t_n)$.
  Using the particle velocities $\mathbf{V}_i(t_n)$ and $\omega_i(t_n)$, update particle positions and orientations $\mathbf{X}_i(t_n)$ and $\Theta_i(t_n)$.
End Do

The above scheme is fully explicit as the particle positions and velocities are explicitly updated at each time step. The scheme is simple and easy to implement. Unfortunately the scheme could be unstable under certain circumstances (see Hu *et al.* [30]). Let us consider the initial stage of the motion of a particle accelerating from rest in an infinite medium of quiescent fluid. In the early stages of the motion, the particle velocity is very small, with the drag on the particle being mainly from the virtual (or added) mass force, which is caused by the acceleration of the mass of the fluid surrounding the particle. The translational motion of the particle takes the form

$$m\frac{d\mathbf{V}}{dt} = \mathbf{G} + \mathbf{F} \approx \mathbf{G} - m_{\mathrm{v}}\frac{d\mathbf{V}}{dt}, \tag{17}$$

where $\mathbf{G}$ is a constant driving force such as the weight of the particle, $\mathbf{F}$ is the hydrodynamic drag acting on the particle (mainly due to the virtual mass force in this situation), m is the mass of the particle, and m_{v} is the virtual mass of the fluid. Using the fully explicit scheme described above, we can calculate the hydrodynamic force acting on the particle based on the particle velocity in the previous time step, and (17) can be written as

$$m\frac{d\mathbf{V}}{dt}(t_n) = \mathbf{G} - m_{\mathrm{v}}\frac{d\mathbf{V}}{dt}(t_{n-1}), \tag{18}$$

or

$$\begin{aligned}
\frac{d\mathbf{V}}{dt}(t_n) &= \frac{\mathbf{G}}{m} - \frac{m_{\mathrm{v}}}{m}\frac{d\mathbf{V}}{dt}(t_{n-1}) \\
&= \frac{\mathbf{G}}{m} - \frac{m_{\mathrm{v}}}{m}\left[\frac{\mathbf{G}}{m} - \frac{m_{\mathrm{v}}}{m}\frac{d\mathbf{V}}{dt}(t_{n-2})\right] \\
&= \frac{\mathbf{G}}{m}\left[1 + \left(-\frac{m_{\mathrm{v}}}{m}\right) + \left(-\frac{m_{\mathrm{v}}}{m}\right)^2 + \cdots + \left(-\frac{m_{\mathrm{v}}}{m}\right)^{n-1}\right] + \left(-\frac{m_{\mathrm{v}}}{m}\right)^n \frac{d\mathbf{V}}{dt}(t_0) \\
&= \frac{1 - \left(-\frac{m_{\mathrm{v}}}{m}\right)^n}{m + m_{\mathrm{v}}}\mathbf{G} + \left(-\frac{m_{\mathrm{v}}}{m}\right)^n \frac{d\mathbf{V}}{dt}(t_0).
\end{aligned} \tag{19}$$

Obviously, as computation proceeds, the particle velocity oscillates with increasingly large amplitudes when the added mass of the fluid is larger than the mass of the particle, $m_{\mathrm{v}} \geq m$. Therefore, this scheme is not stable. The actual value of the virtual mass, m_{v}, depends on the particle shape and the flow geometry. For a spherical particle in an infinite medium, the virtual mass is equal to one-half the mass of the fluid displaced by the particle. However, if the same sphere is moving through a tightly fitted tube, the value of the virtual mass is much higher, since the sphere needs to accelerate more fluid both ahead and behind the sphere. Therefore, the condition $m_{\mathrm{v}} \geq m$ could be true for certain fluid–particle systems, for example, those involving the motion of light particles in a fluid or the motion of particles in some confined geometries.

To avoid the stability problem of the fully explicit scheme, a coupled scheme for solving the flow field and particle velocities in a given time step is needed. For example, the solution of the flow field and the forces and moments acting on the particles could be determined iteratively with the velocities of the particles at the same time instant (an implicit scheme), or a predictor–corrector scheme could be used to update the particle velocity. However, one can treat the fluid and the solid particles as one system and generate a combined formulation for this system.

4. COMBINED FLUID–SOLID FORMULATION

In fluid–particle systems, owing to the complex, irregular nature of the domain occupied by the fluid, finite-element techniques are particularly powerful for discretizing the governing fluid equations. In order to use the finite-element method, we first seek a weak formulation that incorporates both the fluid and particle equations of motion, Eqs. (2), (7), and (8).

Let us introduce the function space $\mathbf{V}$, given by

$$\mathbb{V} = \left\{ \begin{array}{l} \mathbf{U} = (\mathbf{u}, \mathbf{V}_i, \ldots, \mathbf{V}_N, \omega_1, \ldots, \omega_N) \mid \mathbf{u} \in H^1(\Omega_0)^3, \mathbf{V}_i \in \Re^3, \omega_i \in \Re^3, \\ \mathbf{u} = \mathbf{V}_i + \omega_i \times (\mathbf{x} - \mathbf{X}_i) \text{ on } \partial\Omega_i(t), \text{ and } \mathbf{u} = \mathbf{u}_g \text{ on } (\partial\Omega)_u \text{ for } i = 1, 2, \ldots, N \end{array} \right\}, \tag{20}$$

where $H^1(\Omega_0)^3$ corresponds to the space for the 3-D velocity field in the fluid, and $\Re^3$ stands for the space for the particle velocities (three translational and three angular velocity components per particle). The space $\mathbb{V}$ is a natural space for the velocity of the fluid–solid mixture. The space for the pressure is chosen as $L^2(\Omega_0)$ with a zero value at a fixed point in the domain and is denoted as

$$\mathbb{Q}(\Omega_0) = \{p \mid p \in L^2(\Omega_0),\ p(\mathbf{x}_0) = 0\}. \tag{21}$$

Similarly, the space for the elastic polymer stress tensor is selected to be $L^2(\Omega_0)^6$, which represents six independent components, and is denoted as

$$\mathbb{T} = \{\tau \mid \tau \in L^2(\Omega_0)^6,\ \tau = \tau_{\text{in}} \text{ on } (\partial\Omega)_{\text{in}}\}. \tag{22}$$

To derive the weak formulation of the combined fluid and particle equations of motion, we consider a test function (the variation of $\mathbf{U}$),

$$\tilde{\mathbf{U}} = (\tilde{\mathbf{u}}, \tilde{\mathbf{V}}_1, \ldots, \tilde{\mathbf{V}}_N, \tilde{\omega}_1, \ldots, \tilde{\omega}_N) \in \mathbb{V}_0, \tag{23}$$

where the variational space $\mathbb{V}_0$ is the same as $\mathbb{V}$, except that $\mathbf{u} = 0$ on $(\partial\Omega)_u$. We shall define the variational space $\mathbb{T}_0$ to be the same as $\mathbb{T}$, except that $\tau = 0$ on $(\partial\Omega)_{\text{in}}$. Multiplying (2) by the test function for the fluid velocity, $\tilde{\mathbf{u}}$, and integrating over the fluid domain at a time instant t, we have

$$\int_{\Omega_0} \rho_f \left(\frac{D\mathbf{u}}{Dt} - \mathbf{f} \right) \cdot \tilde{\mathbf{u}}\, d\Omega + \int_{\Omega_0} (\sigma : \nabla\tilde{\mathbf{u}})\, d\Omega - \sum_{1 \le i \le N} \int_{\partial\Omega_i} (\sigma \cdot \mathbf{n}) \cdot \tilde{\mathbf{u}}\, dS = 0. \tag{24}$$

The test function for the fluid velocity on the particle surface will be replaced with the test functions of the particle velocities according to the no-slip condition (13). Furthermore, using the equations of motion for the particles (7) and (8), we obtain

$$\begin{aligned} -\int_{\partial\Omega_i} (\sigma \cdot \mathbf{n})\, \tilde{\mathbf{u}}\, dS &= -\int_{\partial\Omega_i} (\sigma \cdot \mathbf{n}) \cdot [\tilde{\mathbf{V}}_i + \tilde{\omega}_i \times (\mathbf{x} - \mathbf{X}_i)]\, dS \\ &= -\tilde{\mathbf{V}}_i \cdot \int_{\partial\Omega_i} (\sigma \cdot \mathbf{n})\, dS - \tilde{\omega}_i \cdot \int_{\partial\Omega_i} (\mathbf{x} - \mathbf{X}_i) \times (\sigma \cdot \mathbf{n})\, dS \\ &= \tilde{\mathbf{V}}_i \cdot \left(m_i \frac{d\mathbf{V}_i}{dt} - \mathbf{G}_i \right) + \tilde{\omega}_i \cdot \frac{d(\mathbf{I}_i \omega_i)}{dt}. \end{aligned} \tag{25}$$

Substituting (25) into (24), we find the combined fluid–particle momentum equation

$$\int_{\Omega_0} \rho_f \left(\frac{D\mathbf{u}}{Dt} - \mathbf{f} \right) \cdot \tilde{\mathbf{u}} \, d\Omega \quad + \int_{\Omega_0} \sigma : \mathbf{D}[\tilde{\mathbf{u}}] \, d\Omega + \sum_{1 \le i \le N} \tilde{\mathbf{V}}_i \cdot \left(m_i \frac{d\mathbf{V}_i}{dt} - \mathbf{G}_i \right)$$
$$+ \sum_{1 \le i \le N} \tilde{\omega}_i \cdot \frac{d(\mathbf{I}_i \omega_i)}{dt} = 0. \tag{26}$$

The stress tensor in (26) can be replaced with (5). The weak formulations for the mass conservation (1) and the constitutive equation (6) can also be similarly obtained by multiplying their corresponding test functions and integrating over the fluid domain.

In summary, the weak formulation for the combined equations of the fluid–particle system is as follows:

Find $(\mathbf{U}, p, \tau_p) \in \mathbb{V} \times \mathbb{Q} \times \mathbb{T}$, such that

$$\int_{\Omega_0} \rho_f \left(\frac{D\mathbf{u}}{Dt} - \mathbf{f} \right) \cdot \tilde{\mathbf{u}} \, d\Omega + 2 \int_{\Omega_0} \eta_2 \mathbf{D}[\mathbf{u}] : \mathbf{D}[\tilde{\mathbf{u}}] \, d\Omega - \int_{\Omega_0} p(\nabla \cdot \tilde{\mathbf{u}}) \, d\Omega + \int_{\Omega_0} \tau_p : \mathbf{D}[\tilde{u}] \, d\Omega$$
$$+ \sum_{1 \le i \le N} \tilde{\mathbf{V}}_i \cdot \left(m_i \frac{d\mathbf{V}_i}{dt} - \mathbf{G}_i \right) + \sum_{1 \le i \le N} \tilde{\omega}_i \cdot \frac{d(\mathbf{I}_i \omega_i)}{dt} = 0 \qquad \forall \tilde{\mathbf{U}} \in \mathbb{V}_0 \tag{27}$$

$$\int_{\Omega_0} \tilde{p} \nabla \cdot \mathbf{u} \, d\Omega = 0 \qquad \forall \tilde{p} \in \mathbb{Q} \tag{28}$$

$$\int_{\Omega_0} \tilde{\tau} : \left[\lambda \left(\frac{D\tau_p}{Dt} - \nabla \mathbf{u}^T \cdot \tau_p - \tau_p \cdot \nabla \mathbf{u} \right) + \tau_p - 2\eta_1 \mathbf{D}[\mathbf{u}] \right] d\Omega = 0 \qquad \forall \tilde{\tau} \in \mathbb{T}_0. \tag{29}$$

It is noted that in the combined momentum equation (27) for the fluid–particle system, the hydrodynamic forces and moments acting on the particles do not explicitly appear in the formulation. This fact comes out naturally, since these forces are internal when the fluid and the solid particles are considered as one system. The advantage of this combined formulation is that the hydrodynamic forces and moments need not be explicitly computed. More importantly, the scheme based on this formulation is not subject to the numerical instability which can arise when the equations of fluid and particle motion are integrated with explicitly computed hydrodynamic forces and moments, as discussed in the previous section.

5. ARBITRARY LAGRANGIAN–EULERIAN (ALE) MESH MOVEMENT

As we are expecting a large number of solid particles moving freely in the fluid, the domain occupied by the fluid is irregular and changes with time. To handle the movement of the domain, an arbitrary Lagrangian–Eulerian (ALE) technique can be used. In this section we describe this technique and discuss the methods for controlling the mesh movement. A general kinematic theory for the ALE technique was originally introduced by Hughes *et al.* [37].

In an ALE formulation, the material time derivative (3) of the velocity at a given point $\mathbf{x}$ in the fluid domain and at a time instant t is written as

$$\frac{D}{Dt}\mathbf{u}(\mathbf{x}, t) = \frac{\delta \mathbf{u}}{\delta t} + [(\mathbf{u} - \hat{\mathbf{u}}) \cdot \nabla]\mathbf{u}, \tag{30}$$

where

$$\frac{\delta}{\delta t}\mathbf{u}(\mathbf{x}, t) = \frac{\partial}{\partial t}\mathbf{u}(\mathbf{x}(\chi, t), t)|_{\chi \text{ fixed}} \tag{31}$$

is the referential time derivative keeping the coordinates, χ, in the referential domain constant. The function $\mathbf{x}(\chi, t)$ can be viewed as a mapping from the fixed referential domain to the spatial domain $\Omega_0(t)$ where the fluid mechanics problem is posed. The velocity $\hat{\mathbf{u}}(\mathbf{x}, t)$ is the velocity of the domain (or the mesh velocity) and is defined as

$$\frac{d}{dt}\mathbf{x}(\chi, t) = \hat{\mathbf{u}}. \tag{32}$$

When the referential domain coincides with the spatial domain at the current time, $\chi = \mathbf{x}$, we have $\hat{\mathbf{u}} = 0$, and the referential time derivative (31) reduces to the local Eulerian time derivative. When the mesh velocity coincides with the velocity of the material particles, $\hat{\mathbf{u}} = \mathbf{u}$, the referential time derivative (31) recovers the Lagrangian (or material) time derivative.

In general, the domain (or mesh) velocity in (32) is only constrained at the boundary of the domain. It has to follow the motion of the particles and the motion of the confining flow geometry. In the interior of the domain, the mesh velocity is largely arbitrary.

If the deformation of the domain is prescribed, or somewhat predictable, the mesh velocity in the interior can be expressed simply as algebraic functions of the motion of the nodes at the boundary, such as the ones used in Huerta and Liu [35] and Nomura and Hughes [54].

For more complicated motion of particles in a fluid, the mesh motion in the interior of the fluid can be assumed to satisfy an elliptic partial differential equation, such as Laplace's equation, to guarantee its smooth variation,

$$\nabla \cdot (k^{\mathrm{e}} \nabla \hat{\mathbf{u}}) = 0 \qquad \text{in } \Omega_0(t), \tag{33}$$

where k^{e} is a function introduced to control the deformation of the domain such that the region away from the particles absorbs most of the deformation, while the region next to the particles is relatively stiff and retains its shape better. Here, we choose k^{e} to be the inverse of the local element volume. This mesh movement scheme was used by Hu [28]. It should be noted that the components of the mesh velocity $\hat{\mathbf{u}}$ are not coupled and can be solved separately. The boundary conditions that the mesh velocity must satisfy are

$$\hat{\mathbf{u}} = \mathbf{V}_i + \omega_i \times (\mathbf{x} - \mathbf{X}_i), \qquad \text{for } \mathbf{x} \in \partial\Omega_i(t),\ i = 1, 2, \ldots, N \tag{34}$$

and

$$\hat{\mathbf{u}} = 0 \qquad \text{on } (\partial\Omega)_u \cup (\partial\Omega)_\sigma. \tag{35}$$

It is possible to use different boundary conditions for certain flow problems. With circular or spherical particles, the mesh velocity can be allowed to slip on the particle surface; thus the nodes on a particle surface move with the particle with its translational velocity but do not need to rotate with the particle.

Similarly, if the particles undergo acceleration, an acceleration field, $\hat{\mathbf{a}}(\mathbf{x}, t)$, of the domain can be defined as

$$\nabla \cdot (k^{\mathrm{e}} \nabla \hat{\mathbf{a}}) = 0 \qquad \text{in } \Omega_0(t) \tag{36}$$

with the boundary conditions given by

$$\hat{\mathbf{a}} = \dot{\mathbf{V}}_i + \dot{\omega}_i \times (\mathbf{x} - \mathbf{X}_i) - \omega_i \times \mathbf{V}_i, \qquad \text{for } \mathbf{x} \in \partial\Omega_i(t),\ i = 1, 2, \ldots, N \tag{37}$$

and

$$\hat{\mathbf{a}} = 0 \qquad \text{on } (\partial\Omega)_u \cup (\partial\Omega)_\sigma, \tag{38}$$

where $\dot{\mathbf{V}}_i = d\mathbf{V}_i/dt$ and $\dot{\omega}_i = d\omega_i/dt$. This mesh acceleration field is useful when a higher order scheme is needed to discretize Eq. (32) for the mesh movement.

A similar mesh movement scheme was described by Johnson [40]. In his implementation, the domain is modeled as a linear elastic solid, and thus the equations of linear elasticity were used to solve for the mesh velocity in the interior of the domain based on the given boundary deformation. Thus the components of the mesh velocity are coupled in the scheme and have to be solved together. Johnson [40] also used a variable stiffness coefficient to control the mesh deformation so that most of the mesh deformation is absorbed by the larger elements in the mesh and the small elements are stiffer and retain their shape better.

The weak formulations for the equations of the mesh velocity (33) and the mesh acceleration (36) can be written as

Find $\hat{\mathbf{u}} \in \mathbb{V}_{\text{mesh}\,1}$ and $\hat{\mathbf{a}} \in \mathbb{V}_{\text{mesh}\,2}$, such that

$$\int_{\Omega_0} (k^{\text{e}} \nabla\hat{\mathbf{u}} \cdot \nabla\tilde{\hat{\mathbf{u}}})\, d\Omega = 0 \qquad \forall\tilde{\hat{\mathbf{u}}} \in \mathbb{V}_{\text{mesh}\,0} \tag{39}$$

and

$$\int_{\Omega_0} (k^{\text{e}} \nabla\hat{\mathbf{a}} \cdot \nabla\tilde{\hat{\mathbf{a}}})\, d\Omega = 0 \qquad \forall\tilde{\hat{\mathbf{a}}} \in \mathbb{V}_{\text{mesh}\,0}, \tag{40}$$

where the function spaces are defined as

$$\mathbb{V}_{\text{mesh}\,1} = \{\hat{\mathbf{u}} \in H^1(\Omega_0)^3;\ \hat{\mathbf{u}} = \mathbf{V}_i + \omega_i \times (\mathbf{x} - \mathbf{X}_i) \text{ on } \partial\Omega_i;\ \hat{\mathbf{u}} = 0 \text{ on } (\partial\Omega)_u \cup (\partial\Omega)_\sigma\}, \tag{41}$$

$$\mathbb{V}_{\text{mesh}\,2} = \{\hat{\mathbf{a}} \in H^1(\Omega_0)^3;\ \hat{\mathbf{a}} = \dot{\mathbf{V}}_i + \dot{\omega}_i \times (\mathbf{x} - \mathbf{X}_i) - \omega_i \times \mathbf{V}_i \text{ on } \partial\Omega_i;\ \hat{\mathbf{a}} = 0 \text{ on } (\partial\Omega)_u \cup (\partial\Omega)_\sigma\}, \tag{42}$$

and

$$\mathbb{V}_{\text{mesh}\,0} = \{\hat{\mathbf{u}} \in H^1(\Omega_0)^3;\ \hat{\mathbf{u}} = 0 \text{ on } \partial\Omega_0\}. \tag{43}$$

6. TEMPORAL DISCRETIZATION—FINITE-DIFFERENCE SCHEME

Owing to the special nature of the temporal coordinate, the time derivatives in the system of equations are usually discretized by simpler finite-difference methods. In this section, we introduce a finite-difference scheme to replace the time derivatives in the combined fluid–particle system of (27)–(29). We shall consider all the terms in the equations (27)–(29) at

a given instant $t = t_{n+1}$ (fully implicit discretization). First, the referential time derivative in (31) can be discretized as

$$\frac{\delta \mathbf{u}}{\delta t}(\mathbf{x}, t_{n+1}) \approx \alpha \frac{\mathbf{u}(\mathbf{x}, t_{n+1}) - \mathbf{u}(\bar{\mathbf{x}}, t_n)}{\Delta t} - \beta \frac{\delta \mathbf{u}}{\delta t}(\bar{\mathbf{x}}, t_n), \tag{44}$$

where $\Delta t = t_{n+1} - t_n$ is the time step, and the mesh nodes are moved according to an integrated version of (32),

$$\mathbf{x} = \bar{\mathbf{x}} + \hat{\mathbf{u}}(\bar{\mathbf{x}}, t_n)\Delta t + \hat{\mathbf{a}}(\bar{\mathbf{x}}, t_n)\frac{\Delta t^2}{2}. \tag{45}$$

The approximation in (44) is first-order accurate in time when ($\alpha = 1, \beta = 0$). It can be improved to second-order accurate in time by selecting ($\alpha = 2, \beta = 1$), which is a variation of the well-known Crank–Nicolson scheme.

Therefore, the material time derivative (30) can be written as

$$\begin{aligned}\frac{D\mathbf{u}}{Dt}(\mathbf{x}, t_{n+1}) \approx \alpha \frac{\mathbf{u}(\mathbf{x}, t_{n+1}) - \mathbf{u}(\bar{\mathbf{x}}, t_n)}{\Delta t} - \beta \frac{\delta \mathbf{u}}{\delta t}(\bar{\mathbf{x}}, t_n) \\ + [(\mathbf{u}(\mathbf{x}, t_{n+1}) - \hat{\mathbf{u}}(\mathbf{x}, t_{n+1})) \cdot \nabla]\, \mathbf{u}(\mathbf{x}, t_{n+1}).\end{aligned} \tag{46}$$

Similarly, the time derivatives of the particle velocities in (7) and (8) can be discretized as

$$\frac{d}{dt}\mathbf{V}_i(t_{n+1}) \approx \alpha \frac{\mathbf{V}_i(t_{n+1}) - \mathbf{V}_i(t_n)}{\Delta t} - \beta \frac{d}{dt}\mathbf{V}_i(t_n) \tag{47}$$

and

$$\frac{d}{dt}(\mathbf{I}_i\omega_i)(t_{n+1}) \approx \alpha \frac{(\mathbf{I}_i\omega_i)(t_{n+1}) - (\mathbf{I}_i\omega_i)(t_n)}{\Delta t} - \beta \frac{d}{dt}(\mathbf{I}_i\omega_i)(t_n). \tag{48}$$

However, the equations for the particle positions and orientations (9) and (10) are updated using an explicit finite-difference scheme,

$$\mathbf{X}_i(t_{n+1}) = \mathbf{X}_i(t_n) + \Delta t \mathbf{V}_i(t_n) + \frac{(\Delta t)^2}{2}\dot{\mathbf{V}}_i(t_n) \tag{49}$$

and

$$\Theta_i(t_{n+1}) = \Theta_i(t_n) + \Delta t \omega_i(t_n) + \frac{(\Delta t)^2}{2}\dot{\omega}_i(t_n). \tag{50}$$

As mentioned above, in the weak formulations of (27), (28), and (29), the spatial domain and all the functions in the integrals are evaluated at a given time instant $t = t_{n+1}$ or frozen at this time instant. The time derivatives in (27) and (29) are kept inside the integral and are replaced by expressions such as (46), which gives

$$\begin{aligned}&\int_{\Omega_0} \rho_{\mathrm{f}} \left(\frac{\alpha}{\Delta t}\mathbf{u} + (\mathbf{u} - \hat{\mathbf{u}}) \cdot \nabla \mathbf{u} \right) \cdot \tilde{\mathbf{u}}\, d\Omega + 2\int_{\Omega_0} \eta_2 \mathbf{D}[\mathbf{u}] : \mathbf{D}[\tilde{\mathbf{u}}]\, d\Omega - \int_{\Omega_0} p(\nabla \cdot \tilde{\mathbf{u}})\, d\Omega \\ &+ \int_{\Omega_0} \tau_{\mathrm{p}} : \mathbf{D}[\tilde{\mathbf{u}}]\, d\Omega + \frac{\alpha}{\Delta t} \sum_{1 \le i \le N} m_i \mathbf{V}_i \cdot \tilde{\mathbf{V}}_i + \frac{\alpha}{\Delta t} \sum_{1 \le i \le N} (\mathbf{I}_i\omega_i) \cdot \tilde{\omega}_i\end{aligned}$$

$$= \int_{\Omega_0} \rho_f \left(\frac{\alpha}{\Delta t}\mathbf{u}(\bar{\mathbf{x}}, t_n) + \beta \frac{\delta \mathbf{u}}{\delta t}(\bar{\mathbf{x}}, t_n) + \mathbf{f} \right) \cdot \tilde{\mathbf{u}}\, d\Omega$$
$$+ \sum_{1 \le i \le N} \left(\frac{\alpha}{\Delta t} m_i \mathbf{V}_i(t_n) + \beta m_i \frac{d}{dt}\mathbf{V}_i(t_n) + \mathbf{G}_i \right) \cdot \tilde{\mathbf{V}}_i$$
$$+ \sum_{1 \le i \le N} \left(\frac{\alpha}{\Delta t}(\mathbf{I}_i \omega_i)(t_n) + \beta \frac{d}{dt}(\mathbf{I}_i \omega_i)(t_n) \right) \cdot \tilde{\omega}_i \tag{51}$$

and

$$\int_{\Omega_0} \tilde{\tau} : \left(\lambda \frac{\alpha}{\Delta t}\tau_p + \lambda(\mathbf{u} - \hat{\mathbf{u}}) \cdot \nabla \tau_p - \lambda(\nabla \mathbf{u}^T \cdot \tau_p + \tau_p \cdot \nabla \mathbf{u}) + \tau_p - 2\eta_1 \mathbf{D}[\mathbf{u}] \right) d\Omega$$
$$= \int_{\Omega_0} \lambda \left(\frac{\alpha}{\Delta t}\tau_p(\bar{\mathbf{x}}, t_n) + \beta \frac{\delta \tau_p}{\delta t}(\bar{\mathbf{x}}, t_n) \right) : \tilde{\tau}\, d\Omega. \tag{52}$$

Since the domain of integration and all the functions in the integrals, unless specified otherwise, are all evaluated at the current time instant t_{n+1}, the temporal discretization in (51) and (52) is fully implicit and unconditionally stable. The functions inside the integrals on the right-hand sides of (51) and (52) are known (they are the computed solution of the previous time step). Although the domain on which these functions are defined, which is the old domain $\Omega^n = \Omega_0(t_n)$, is not the same as the domain of the integration, which is the new domain $\Omega^{n+1} = \Omega_0(t_{n+1})$, the integration can be perceived as the integration over the fixed referential domain. The location of the grid in the new domain, $\mathbf{x}$, and its correspondence in the old domain, $\bar{\mathbf{x}}$, is the same in the referential domain. Expression (45) provides the mapping between the old and the new domains.

One should be careful in using (45) to update the nodes on a particle surface, especially using the first-order scheme. If one simply uses the velocity due to the rigid-body motion, the body shape will become more and more distorted, as depicted in Fig. 1 for the case of a rotating rectangular particle. The numerically updated position of its corner A will be located at A″ instead of the desired position A′, after the particle rotates 90°. This is purely a numerical artifact. To keep the shape of the rigid body during the simulation, the nodes on the particle surface should be simply reset to the surface at each time step.

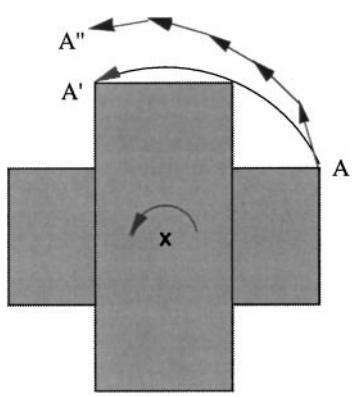

FIG. 1. Distortion of the particle shape due to the improper update of the nodes on the particle surface.

7. SPATIAL DISCRETIZATION—GALERKIN FINITE-ELEMENT SCHEME

In general, the spatial discretization of a partial differential equation can be accomplished by a number of numerical methods. However, in our fluid–particle systems, because of the complex, irregular nature of the domain occupied by the fluid, finite-element methods are particularly attractive. In this section, we discuss the approximation of the weak formulations of (28), (51), and (52) by a Galerkin finite-element formulation and the proper choices of the interpolation functions for the fluid velocity, pressure, and stress.

The fluid domain Ω_0 is first approximated by a finite-element triangulation T_h, where h is the typical mesh size. To fit the surface of the particles, curved P2 quadratic elements are more appropriate. For two-dimensional problems, these elements are triangles with 6 basis functions that are second-order polynomials defined on 3 vertices and 3 mid-nodes on each side of the triangle, as shown in Fig. 2. For three-dimensional problems, these elements are tetrahedrons with 10 basis functions that are second-order polynomials defined on 4 vertices and 6 mid-nodes on each edge of the tetrahedron. The curved quadratic line segments in these elements approximate the local surface curvature of the particle, as indicated in Fig. 2.

Subsequently, the function spaces, $\mathbb{V}$, $\mathbb{Q}$, $\mathbb{T}$, are approximated by their corresponding finite-dimensional counterparts defined on the triangulation T_h. In this particular implementation, we use a mixed type of finite-element, where different interpolation functions are chosen for the different unknown variables. The discrete solution for the fluid velocity is approximated by piecewise quadratic functions and is assumed to be continuous all over the domain (P2). Thus in a finite-element, the velocity is locally interpolated with its values on all 6 nodes in two dimensions or 10 nodes in three dimensions. The discrete solution for the pressure is piecewise linear and continuous (P1). The discrete solution for the components of the stress tensor is also piecewise linear and continuous (P1). In a finite-element, they are locally interpolated only with their values on the vertices. This P1/P2 element for the pressure and velocity is known to satisfy the LBB condition.

One of the advantages of choosing this type of mixed finite-element is that it reduces the cost of mesh generation in comparison with that for the finite-element method using equal-order (linear) interpolation functions for both the velocity and the pressure. For a given desired accuracy of the numerical solution, a coarser mesh with fewer elements would be needed with quadratic velocity interpolation (P2 element) than with linear interpolation function (P1 element). Such cost savings could be considerable for simulations of large

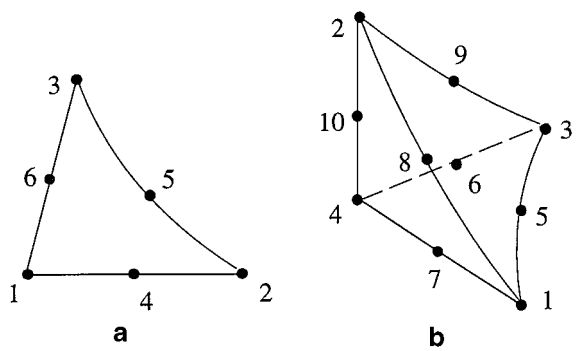

FIG. 2. Curved 6-node triangle and 10-node tetrahedron. The curved line/surface is next to the particle surface.

numbers of particles where the total number of elements is large and the frequency with which the mesh is regenerated (to be discussed in the next section) is also high.

There is another advantage of using the P2 element for the velocity field: When two particles are approaching each other or are moving with respect to the other, in the lubrication limit the velocity profile across the gap between the particles is parabolic. Therefore, the P2 element will capture the exact solution in the region between two particles near contact, even with only a single layer of elements. With the linear P1 elements, a single element across the gap between two particles near contact would create mesh locking and the failure of the numerical scheme. In such a situation, a few layers of elements are needed in the gap region between two moving particles, and a special finite-element mesh generator is needed to guarantee that [43].

On a given finite-element mesh and with the finite-element interpolation functions chosen above, the weak formulations (28), (51), and (52) would reduce to a nonlinear system of algebraic equations. This nonlinear system can be solved by a Newton–Raphson algorithm. In each step of the Newton–Raphson iteration, one solves a linear system of the form

$$\begin{pmatrix} \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{G}_1 & \mathbf{B}_1 \\ \mathbf{A}_3 & \mathbf{A} & \mathbf{G} & \mathbf{B} \\ \mathbf{Q}_1 & \mathbf{Q} & \mathbf{D} & 0 \\ \mathbf{B}_1^T & \mathbf{B}_T & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{U} \\ \mathbf{u} \\ \tau \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_U \\ \mathbf{f}_u \\ \mathbf{f}_\tau \\ \mathbf{f}_p \end{pmatrix}, \tag{53}$$

where all submatrices in the system are sparse. In general, the system is not symmetric. $\mathbf{U}$ is the vector combining all the translational and angular velocities of the solid particles; $\mathbf{u}$, τ, and $\mathbf{p}$ represent, respectively, the vectors collecting all the fluid velocity, stress, and pressure unknowns at grid points in the fluid. In (53) the fluid velocity unknowns on the particle surface are eliminated with the particle velocities using the relationship for the rigid-body motion. The vector on the right-hand side of (53) is the discretized form of the residual of the system for a given trial flow field during the nonlinear iteration.

The weak formulations for the mesh velocity (39) and the mesh acceleration (40) can also be approximated on the finite-dimensional function spaces based on the linear polynomials (P1 element). The final linear systems of algebraic equations are

$$\mathbf{H}\,\hat{\mathbf{u}} = \mathbf{f}_{\mathrm{m1}} \tag{54}$$

and

$$\mathbf{H}\,\hat{\mathbf{a}} = \mathbf{f}_{\mathrm{m2}}, \tag{55}$$

where $\mathbf{H}$ is symmetric and positive definite. In (54) and (55), $\hat{\mathbf{u}}$ and $\hat{\mathbf{a}}$ represent, respectively, the vectors collecting all the mesh velocity and the mesh acceleration unknowns at grid points in the fluid. The vectors on the right-hand side of (54) and (55) are due to the boundary conditions on the surface of the particles.

The algebraic systems (53) and (54) are coupled since the matrix $\mathbf{A}$ in (53) depends on the mesh velocity field $\hat{\mathbf{u}}$. Thus, (53) and (54) need to be solved iteratively at each time step.

The linear system of algebraic equations (53) can be solved with an iterative solver using a preconditioned generalized minimal residual (GMRES) scheme or biconjugate gradient stabilized algorithm (BICGSTAB) (see Saad [60]). These schemes are suitable for the

nonsymmetric matrix in the system (53). In simulations with a GMRES scheme, we set the size of the Krylov subspace to around 20 for good convergence. To make the iterative solver converge, use of a proper preconditioner is essential. The preconditioners, such as ILU(0) or ILU(t) (incomplete LU factorization without or with controlled fill-ins), when applied to the global system, are found to be quite robust and efficient. For more efficient implementations, one may take the advantage of the structure of the system in (53), and design different preconditioners for different parts of the equations within the system. The design of more efficient and reliable preconditioners, especially for parallel computation, is still the topic of active research [49, 60, 63].

The symmetric and positive definite systems of (54) and (55) can be solved iteratively with the conjugate gradient method. A preconditioner such as ILU(0) can be used to improve the convergence.

8. MESH GENERATION

In simulations of fluid–particle systems, complicated interactions of particles make the geometry of the domain occupied by the fluid complex and irregular. We therefore choose to use unstructured finite-element grids (meshes) to cover the computational domain.

The first task in simulating the motion of a fluid–particle system is to generate a finite-element grid based on the initial positions of the particles in the domain. We developed an automatic mesh generator for this purpose. The mesh generator first creates a uniform grid on all the particle surfaces and boundary sections. It then checks the distance between the boundary nodes belonging to different particles or boundary sections. If the distance is less than a certain value, for example, the gap between two corresponding particles, the mesh generator performs a refinement by inserting nodes locally. The purpose of this boundary grid refinement is to eventually generate a fine mesh in the regions where it is needed. With the complete boundary grid information, the mesh generator next generates the elements in the interior of the domain using the Delaunay–Voronoi methods (see, George [22]). Our 3D-volume mesh generator is built around the package GHS3D developed by George and Hecht in INRIA. Finally, the middle nodes are added on the edges of the mesh to form P2/P1 mixed elements used in our finite-element scheme.

In computing solid–liquid flows with a large number of solid particles, it is often necessary to use periodic boundary conditions in one or more directions. At the periodic boundaries, the particles frequently leave and enter the computation domain. The finite-element mesh generator described above automatically takes care of the periodic boundaries without introducing artificial cuts on these boundaries. The artificial cuts on the periodic boundaries may give rise to very unsatisfactory elements. Techniques for periodic finite-element mesh generation are discussed by Patankar and Hu [56], Johnson and Tezduyar [43], and Maury [51].

The mesh generator has a local refinement capability in regions formed by approaching particles or between a particle and the surrounding wall, as mentioned earlier. There is always at least one layer of elements in those regions, and the mesh size in those regions is designed to be of the order of the minimum gap size between the approaching particles. The local refinement in the gaps between particles is essential to capture the "particle collision" process that is to be discussed in Section 11.

Examples of finite-element meshes are presented in Figs. 3 and 4. Figure 3 displays a 2-D mesh with 100 circular disks in a periodic domain between two channel walls. In the

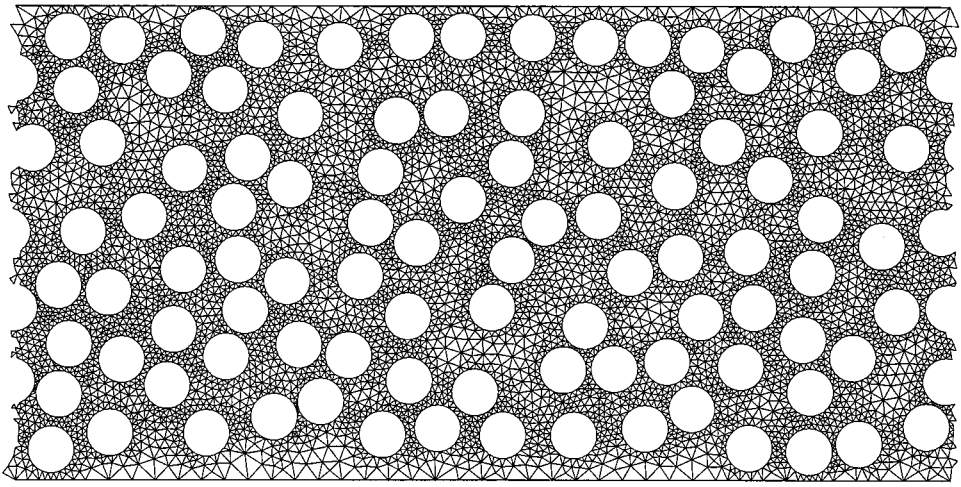

FIG. 3. Two-dimensional finite-element mesh in a channel flow with 100 circular disks.

figure, straight lines are used to connect three vertices of a triangle. However, the curved P2 triangles (with 6 nodes) are actually used in the simulation to fit the curved particle surfaces. Figure 4 shows the meshes on the surfaces of two spheres and on the surface of a circular tube.

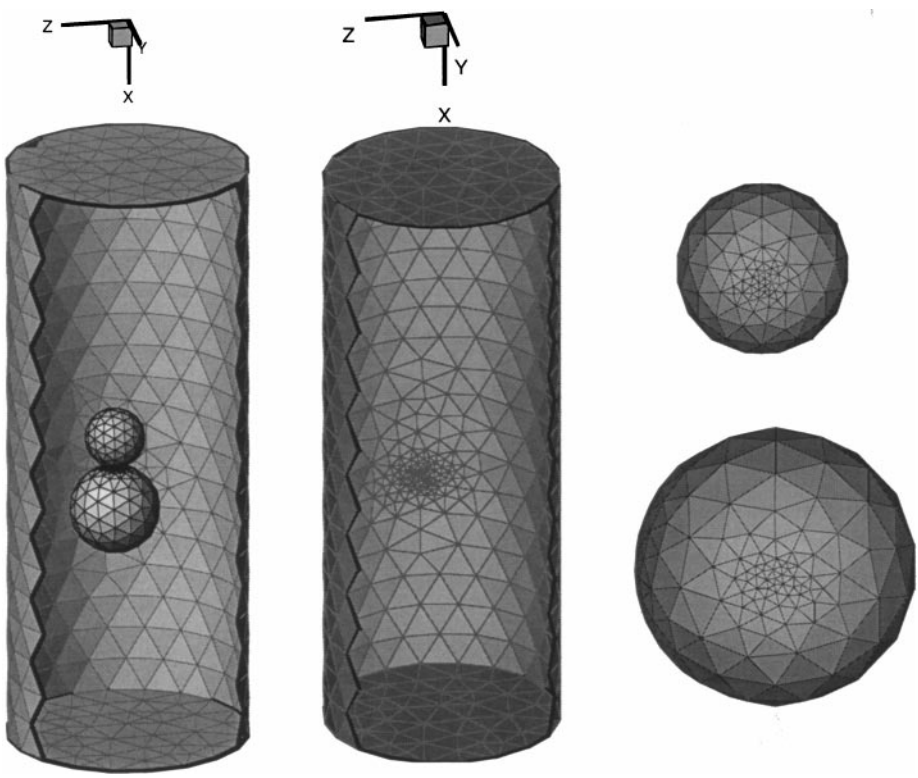

FIG. 4. Surface meshes on the two spheres and the cylindrical tube used in the study of sedimentation of spheres. The mesh is refined in the region of close contact.

During a typical simulation, we start the calculation by generating a finite-element mesh with the automatic mesh generator based on the initial particle positions in the domain. Using this mesh, we can generate and then solve the system of algebraic equations (53), (54). At the new time step, the old finite-element mesh will be moved using (45) according to the mesh velocity and mesh acceleration field obtained from the previous time step. This updated mesh is checked for the quality of its elements. If unacceptable element distortion is detected, a new finite-element mesh will be generated with the automatic mesh generator. The new mesh may not have any correspondence with the old mesh. The solution from the old mesh has to be projected onto the new mesh. Once the solution is projected, computations can proceed normally.

The quality of the mesh can be measured by checking the change in the element volume (and/or aspect ratio) in comparison with its value in the initial undeformed mesh. The changes of the element volume and aspect ratio are defined as

$$f_1^e = \left|\log\left(V^e/V_0^e\right)\right| \quad \text{and} \quad f_2^e = \left|\log\left(S^e/S_0^e\right)\right|, \tag{56}$$

where V^e and V_0^e are the volume of the eth element and its value in the initial undeformed mesh, respectively; S^e and S_0^e are the aspect ratio of the element and its value in the initial undeformed mesh, respectively. The aspect ratio is defined as

$$S^e = \frac{(l^e)^3}{V^e}, \tag{57}$$

where l^e is the maximum length of the sides of the element e. The global quality of the mesh is measured by the maximum mesh deformation,

$$f_1 = \max_{1\le e\le N_{\text{el}}} \left(f_1^e\right) \quad \text{and} \quad f_2 = \max_{1\le e\le N_{\text{el}}} \left(f_2^e\right), \tag{58}$$

where N_{el} is the total number of elements in the mesh. Usually, remeshing is considered when either one of these two parameters exceeds 1.39, which corresponds to the situation where element volume (or aspect ratio) is larger than four times or smaller than 1/4 of its original value.

9. PROJECTION SCHEME

At each time step, we explicitly update the particle positions and move the finite-element mesh using Eqs. (49), (50), and (45), based on the solution at the previous time step. If the updated mesh is too distorted we need to generate a new mesh, as described in the previous section. We then need to project the flow field defined on the old mesh onto the new mesh to continue the simulation. Projection errors will be introduced during the process and need to be minimized. There are a number of schemes to perform this projection. Two of them will be discussed in this section.

Let us assume that at the time step $t = t_{n+1}$, the mesh nodes moved from $\bar{\mathbf{x}}$ to $\mathbf{x}$ according to Eq. (45). Supposing that the updated mesh is found to be too distorted, a new mesh $\mathbf{y}$ is generated. The meshes $\mathbf{y}$ and $\mathbf{x}$ cover the same domain occupied by the fluid at $t = t_{n+1}$. Since we have calculated the flow field at the previous step, all the flow properties are known. Let us consider one of the flow properties, $\varphi(\bar{\mathbf{x}}, t_n)$. The objective of the projection

scheme is to find the value of the same flow property $\varphi'(\bar{\mathbf{y}}, t_n)$ on a different mesh $\bar{\mathbf{y}}$ that is traced back from the new mesh $\mathbf{y}$ according to

$$\mathbf{y} = \bar{\mathbf{y}} + \hat{\mathbf{u}}(\bar{\mathbf{y}}, t_n)\Delta t + \hat{\mathbf{a}}(\bar{\mathbf{y}}, t_n)\frac{\Delta t^2}{2}. \tag{59}$$

Since we do not have the velocity $\hat{\mathbf{u}}$ and the acceleration $\hat{\mathbf{a}}$ on the mesh $\bar{\mathbf{y}}$, calculating the direct projection from $\varphi(\bar{\mathbf{x}}, t_n)$ to $\varphi'(\bar{\mathbf{y}}, t_n)$ is difficult. However, since the mapping $\bar{\mathbf{x}}$ to $\mathbf{x}$ given by (45) is affine, the projection can be performed from the mesh $\mathbf{x}$ to the mesh $\mathbf{y}$, namely, the projection from $\varphi(\bar{\mathbf{x}}(\mathbf{x}), t_n) = \bar{\varphi}(\mathbf{x}, t_n)$ to $\varphi'(\bar{\mathbf{y}}(\mathbf{y}), t_n) = \bar{\varphi}'(\mathbf{y}, t_n)$. Another way to view this is to define the projection on the referential domain.

The first scheme is a direct local interpolation scheme. The local interpolation functions could be linear or quadratic depending on the functions being interpolated and the type of elements being used. To find the flow field information at each node in the new mesh, three steps are required. In the first step, one needs to locate the element in the old mesh where a given new node lies. An example of a search in a two-dimensional problem is depicted in Fig. 5. The search is based on evaluating the local (area) coordinates with respect to an element in the old mesh,

$$\begin{aligned} r &= \frac{(x - x_1)(y_3 - y_1) - (x_3 - x_1)(y - y_1)}{(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)} \\ s &= \frac{(x_2 - x_1)(y - y_1) - (x - x_1)(y_2 - y_1)}{(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)}, \end{aligned} \tag{60}$$

where $\mathbf{y} = (x, y)$ is the coordinates of the given node in the new mesh; $\mathbf{x}_a = (x_a, y_a)$ $(a = 1, 2, 3)$ are the coordinates of the three vertices of the element e encountered during the

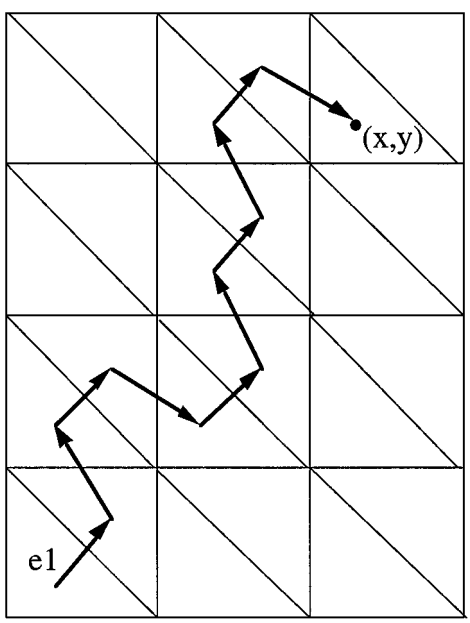

FIG. 5. Diagram of a search scheme to find the element where a given point (x, y) lies. The search starts in the element e_1. The arrow lines indicate the steps in the search process.

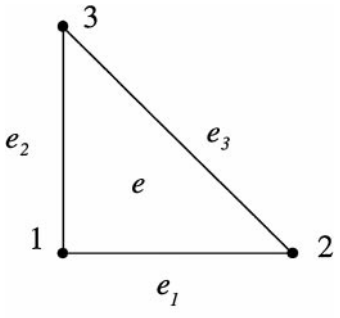

FIG. 6. The search proceeds to the next element e_2 if $r < 0$, or e_1 if $s < 0$, or e_3 if $1 - r - s < 0$.

search (in the updated mesh **x**). If the local coordinates calculated from (60) satisfy

$$0 \le r \le 1, \quad 0 \le s \le 1 \quad \text{and} \quad 0 \le 1 - r - s \le 1, \tag{61}$$

then the node (x, y) belongs to this element e. Otherwise, the values of the local coordinates and the information on element neighbors are used to proceed the search for the next element, using the general rules indicated in Fig. 6. These rules are modified next to boundaries or particle surfaces, where the search takes whatever path that is available. If a search encounters a dead end, it backtracks to the previous possible bifurcation point and selects a different path. This search scheme is able to get around particles in the domain and can be easily extended to three dimensions.

The second step in this interpolation scheme is to calculate the exact local coordinates for the node within the found element. During the search step, the elements are assumed to be linear (with straight sides). However, they may be curved. For curved high-order elements, the calculation for the local coordinates involves solving a set of nonlinear equations,

$$\mathbf{x} = \sum_{1 \le a \le N_\mathrm{d}} \mathbf{x}_a N_a(r, s), \tag{62}$$

where N_d is the total number of nodes in the element and $\mathbf{x}_a$ and $N_a(r, s)$ are the coordinates of the nodes and the corresponding interpolation functions in the element, respectively.

Once the local coordinates (r, s) for the new node are obtained, the interpolation of a variable at this node can be easily achieved by using the local interpolation functions and the nodal values of the variable on the located element, that is,

$$\overline{\varphi}'(\mathbf{y}, t_n) = \sum_{1 \le a \le N_\mathrm{d}} \overline{\varphi}(\mathbf{x}_a, t_n) N_a(r, s). \tag{63}$$

One can estimate the numerical error produced in this type of projection. A simple analysis (see Patankar [55]) shows that the projection error is of order h^2 for linear elements and is of order h^3 for quadratic elements, where h is the mesh size.

The second projection scheme uses a global least-squares method. With the notation described at the beginning of this section, the projection is done by minimizing the difference

between the function representations on the old mesh and on the new mesh, or

$$\text{Minimize} \int_{\Omega_0(t_{n+1})} [\bar{\varphi}'(\mathbf{y}, t_n) - \bar{\varphi}(\mathbf{x}, t_n)]^2 \, d\Omega(\mathbf{y}), \tag{64}$$

where the integration is performed over the new mesh. The weak formulation of (64) can be written as

Given $\bar{\varphi}(\mathbf{x}, t_n)$, find $\bar{\varphi}'(\mathbf{y}, t_n)$ such that

$$\begin{aligned} \int_{\Omega_0(t_{n+1})} \bar{\varphi}'(\mathbf{y}, t_n)\tilde{\varphi}(\mathbf{y}) \, d\Omega(\mathbf{y}) &= \int_{\Omega_0(t_{n+1})} \bar{\varphi}(\mathbf{x}, t_n)\tilde{\varphi}(\mathbf{y}) \, d\Omega(\mathbf{y}) \\ &= \int_{\Omega_0(t_{n+1})} \varphi(\bar{\mathbf{x}}, t_n)\tilde{\varphi}(\mathbf{y}) \, d\Omega(\mathbf{y}) \qquad \text{for } \forall\tilde{\varphi}(\mathbf{y}), \end{aligned} \tag{65}$$

where $\bar{\varphi}'(\mathbf{y}, t_n)$ and $\tilde{\varphi}(\mathbf{y})$ belong to the appropriate function spaces for the flow variables. The integration on the right-hand side of (65) can be performed numerically, where the values of $\bar{\varphi}(\mathbf{x}, t_n)$ at Gaussian quadrature points are needed. These values are calculated using the local interpolation scheme described above. Once the function is approximated by the finite-dimensional finite-element space, (65) reduces to a set of linear algebraic equations and can be solved by iterative methods such as the conjugate gradient method.

The global least-squares projection scheme generally performs better than the local interpolation scheme. This may be due to the fact that the right-hand side of (65) is the same as the terms on the right-hand side of (51) and (52). If the right-hand side of (65) is evaluated exactly, the global least-squares projection scheme would be exact. Thus the projection would not introduce any additional error, or is said to be consistent.

10. EXPLICIT–IMPLICIT SOLUTION PROCEDURE

So far we have described all the major steps needed for simulations of fluid–solid systems. In this section, we summarize these steps and present a solution procedure for fluid–solid systems. This procedure is termed explicit–implicit; the particle positions and the mesh nodes in the fluid domain are updated explicitly, while the particle velocities and the fluid flow field are determined implicitly.

SCHEME 2. Explicit-Implicit Scheme.

Initialization: $t_0 = 0$, $n = 0$ (index for time step).

Generate initial mesh $\mathbf{x}_0$ based on particle positions and orientations, $\mathbf{X}_i(0)$, $\Theta_i(0)$.

Initialize $u(x, t_0)$ and $\mathbf{V}_i(t_0)$, $\omega_i(t_0)$ for $i = 1, 2. \ldots, N$.

Do $n = 1, 2, \ldots, M$ (total number of time steps)

1. Select time step Δt_n:$t_n = t_{n-1} + \Delta t_n$.
2. Update particle positions:

$$\begin{aligned} \mathbf{X}_i(t_n) &= \mathbf{X}_i(t_{n-1}) + \mathbf{V}_i(t_{n-1})\Delta t_n + \dot{\mathbf{V}}_i(t_{n-1})(\Delta t_n)^2/2, \\ \Theta_i(t_n) &= \Theta_i(t_{n-1}) + \omega_i(t_{n-1})\Delta t_n + \dot{\omega}_i(t_{n-1})(\Delta t_n)^2/2. \end{aligned}$$

3. Update mesh nodes:

$$\mathbf{y}(t_n) = \bar{\mathbf{x}}(t_{n-1}) + \hat{\mathbf{u}}(\bar{\mathbf{x}}, t_{n-1})\Delta t_n + \hat{\mathbf{a}}(\bar{\mathbf{x}}, t_{n-1})(\Delta t_n)^2/2.$$

4. Check mesh quality; if the updated mesh $\mathbf{y}(t_n)$ is too distorted, then

   generate a new mesh $\mathbf{x}(t_n)$.

   project the flow field from $\mathbf{y}(t_n)$ onto $\mathbf{x}(t_n)$.

5. Iteratively solve for the flow field $\mathbf{u}(\mathbf{x}(t_n), t_n)$, $p(\mathbf{x}(t_n), t_n)$, $\tau(\mathbf{x}(t_n), t_n)$, the mesh velocity $\hat{\mathbf{u}}(\mathbf{x}(t_n), t_n)$, and the particle velocities $\mathbf{V}_i(t_n)$ and $\omega_i(t_n)$.
6. Update $\dot{\mathbf{V}}_i(t_n)$, $\dot{\omega}_i(t_n)$, $\delta/\delta t\, \mathbf{u}(\mathbf{x}(t_n), t_n)$, $\delta/\delta t\, \tau(\mathbf{x}(t_n), t_n)$ from equations such as (47), (48), and (44).
7. Solve for the mesh acceleration $\hat{\mathbf{a}}(\mathbf{x}(t_n), t_n)$.

End Do

The choice of the time step Δt_n in the scheme depends on many factors. It can be used to restrict the maximum distance each particle is allowed to travel in that time step, to restrict the maximum change in the particle velocity, or to avoid the collisions between the particles and between the particle and the confining boundary walls. The time step should also be restricted to capture unsteady dynamic behavior of the fluid motion, such as vortex shedding in the flow.

This explicit–implicit scheme is second-order accurate in time and numerically stable. It was first described in Hu *et al.* [30], and since then it has been used in a number of studies of fluid–particle systems.

11. PARTICLE COLLISION

It is not possible to simulate the motion of even a moderately dense suspension of particles without a strategy to handle cases in which particles touch. In various numerical methods, frequent near collisions force large numbers of mesh points into the narrow gap between close particles and the mesh distorts rapidly, requiring an expensive high frequency of remeshing and projection. Different "collision models" were developed to prevent near collisions while still conserving mass and momentum. In this section, we discuss some of these models.

It can easily be proved that smooth rigid particles in a Newtonian fluid cannot touch—the gap between two particles cannot go to zero within a finite time. To have real collision of smooth rigid particles, it is necessary for the fluid film between the particles to rupture and film rupture requires physics and mathematics beyond the Navier–Stokes equations. Besides, in practical situations, the particles are normally neither perfectly smooth nor rigid.

The first approach in modeling "particle collision" is to provide a finer zone between the particles as the particles are approaching each other and to use smaller time steps. This approach attempts to capture the collision process as exactly as numerically possible without introducing any modeling. Local mesh refinement schemes, such as the ones discussed in Section 8, are necessary. Numerical experiments based on local mesh refinements show good stability and robustness properties [28, 30]. The smallest gap size between "collide" particles was allowed to be as small as 10^{-5} times the particle diameter. Nevertheless, this approach has the drawback that there is no control of the computational cost.

The next approach is to use the solid-body collision model with a coefficient of restitution. This approach only models the collision process of the solid particles while neglecting the collision process within the fluid. The fluid motion during the particle collision is quite complicated, experiencing a singularity at the time of the particle contact. The solid-body

collision model is only possible when using a fully explicit scheme as described in Section 3. In this model, at each time step once the total forces on the particles are obtained, the particle velocities and positions in (7) and (9) are explicitly updated by

$$\mathbf{V}_i^{n+1} = \mathbf{V}_i^n + \frac{\Delta t}{2m_i}\left(\mathbf{F}_i^{n+1} + \mathbf{F}_i^n\right) \tag{66}$$

and

$$\mathbf{X}_i^{n+1} = \mathbf{X}_i^n + \frac{\Delta t}{2}\left(\mathbf{V}_i^{n+1} + \mathbf{V}_i^n\right) \tag{67}$$

for $i = 1, 2, \ldots, N$, where Δt is the time step, and $\mathbf{F}_i^n$ and $\mathbf{F}_i^{n+1}$ are the total forces acting on the particle at time steps t_n and t_{n+1}, respectively. Using the new positions of the particles, if the particle A is found to overlap particle B, collision occurs between these two particles. For colliding particles A and B, their velocities after the collision need to be modified by

$$\hat{\mathbf{V}}_{\mathrm{A}}^{n+1} = \mathbf{V}_{\mathrm{A}}^{n+1} + \left[V_{n\mathrm{A}}^n - V_{n\mathrm{A}}^{n+1} - \frac{(1+e)m_{\mathrm{B}}}{m_{\mathrm{A}}+m_{\mathrm{B}}}\left(V_{n\mathrm{A}}^n + V_{n\mathrm{B}}^n\right)\right]\mathbf{n}_{\mathrm{A}} \tag{68}$$

$$\hat{\mathbf{V}}_{\mathrm{B}}^{n+1} = \mathbf{V}_{\mathrm{B}}^{n+1} + \left[V_{n\mathrm{B}}^n - V_{n\mathrm{B}}^{n+1} - \frac{(1+e)m_{\mathrm{A}}}{m_{\mathrm{A}}+m_{\mathrm{B}}}\left(V_{n\mathrm{A}}^n + V_{n\mathrm{B}}^n\right)\right]\mathbf{n}_{\mathrm{B}}, \tag{69}$$

where e is the coefficient of restitution, $V_{n\mathrm{A}} = \mathbf{V}_{\mathrm{A}} \cdot \mathbf{n}_{\mathrm{A}}$, $V_{n\mathrm{B}} = \mathbf{V}_{\mathrm{B}} \cdot \mathbf{n}_{\mathrm{B}}$, and $\mathbf{n}_{\mathrm{A}} = -\mathbf{n}_{\mathrm{B}}$ is the unit normal vector pointing from the center of particle A to the center of particle B. In deriving (68) and (69) it is assumed that linear momentum of the two particles is conserved and that the tangential forces are zero during the collision process. The velocity correction due to the collision is applied in an iterative fashion. The new particle positions (67) are updated with these corrected velocities, and other particle collisions are checked again.

Different collision models (for example, the ALE particle mover and the DLM particle mover) have been developed for the coupled solvers for the fluid and solid systems. These collision models aim to capture the "collision process" for both the solid particles and the fluid motion by introducing short-range forces as additional body forces acting on the particles. They all define a security zone around the particle such that when the gap between particles is smaller than the security zone a repelling force is activated. A repelling force can be thought to represent surface roughness, for example. The repelling force pushes the particles out of the security zone into the region in which fluid forces computed numerically govern. The different strategies differ in the nature of the repelling forces and how they are computed.

The scheme used by Glowinski *et al.* [24] introduces a short-range repulsion force between particles near contact. This force takes an explicit form

$$\mathbf{G}_{i,j}^{\mathrm{p}} = \begin{cases} \mathbf{0}, & d_{ij} > R_i + R_j + \delta \\ \frac{1}{\varepsilon_{\mathrm{p}}}(\mathbf{X}_i - \mathbf{X}_j)(R_i + R_j + \delta - d_{ij})^2, & d_{ij} \le R_i + R_j + \delta, \end{cases} \tag{70}$$

where d_{ij} is the distance between the centers of particles i and j, R_i is the radius of the ith particle, δ is the force range, and ε_{p} is a small positive "stiffness" parameter. A similar repulsive force is introduced to handle the collision between the particle and wall,

$$\mathbf{G}_{i,j}^{\mathrm{w}} = \begin{cases} \mathbf{0}, & d'_{ij} > 2R_i + \delta \\ \frac{1}{\varepsilon_{\mathrm{w}}}(\mathbf{X}_i - \mathbf{X}'_{i,j})(2R_i + \delta - d'_{ij})^2, & d'_{ij} \le 2R_i + \delta. \end{cases} \tag{71}$$

In (71) an imaginary particle of the same size is introduced. It is located symmetrically on the other side of the wall, at $\mathbf{X}'_{i,j}$. The distance d'_{ij} is between the particle i and its image with respect to the wall section j. Another positive "stiffness" parameter ε_{w} is introduced to control the magnitude of this force.

Therefore, the extra body force on the ith particle, due to the collision with all other particles and the walls, can be expressed as

$$\mathbf{G}_i^{(\mathrm{c})} = \sum_{j=1,\, j\neq i}^{N} \mathbf{G}_{i,j}^{\mathrm{p}} + \sum_{j=1}^{N_{\mathrm{wall}}} \mathbf{G}_{i,j}^{\mathrm{w}}. \tag{72}$$

This collision force is used in the combined momentum equations for the fluid and solid to determine the fluid velocity field and solid velocities. In this collision scheme, the choice of the "stiffness" parameters, ε_{p} and ε_{w}, is essential. If the parameters were too large (the force were too small), the collisions would not be prevented. In contrast, if the parameters were too small, the repulsive force would be too strong, and the particle would bounce too much during the collision. In general, the optimum values of these two parameters may vary from the test cases. In this collision scheme, there is no control on the minimum distance between the colliding particles. The particles may still overlap.

Another strategy due to Maury [50] uses the lubrication force to separate touching particles and also requires the touching particles to transfer tangential as well as normal momentum.

In the collision scheme used in the ALE particle mover by Hu [28], the expression of the repelling force is not specified. The magnitudes of these forces are such that the particles are forces just to the edge of the security zone. One can imagine that the particles are not smooth, and there is a contact force between the particles when they are approaching within a distance of the size of "particle roughness." The magnitude of this contact force is iteratively determined by requiring that the particles not overlap and that they be in contact just at the edge of the contact zone. In this collision model, after the new particle position is explicitly updated using

$$\mathbf{X}_i(t_{n+1}) = \mathbf{X}_i(t_n) + \mathbf{V}_i(t_n)\Delta t + \dot{\mathbf{V}}_i(t_n)\Delta t^2/2 \tag{73}$$

in step 2 of the explicit–implicit scheme described in Section 10. The collisions between the particles and between the particle and wall are first detected. If particle i is found to contact particle j, then a contact force $\mathbf{G}_{i,j}$ is introduced, which satisfies the relation $\mathbf{G}_{i,j} = -\mathbf{G}_{j,i}$. The new positions of the particles are modified according to

$$\hat{\mathbf{X}}_i(t_{n+1}) = \mathbf{X}_i(t_{n+1}) + \frac{\Delta t^2}{2m_i}\left(\sum_{j=1,\, j\neq i}^{N+N_{\mathrm{wall}}} \mathbf{G}_{i,j}\right), \tag{74}$$

where N_{wall} is the number of boundary segments. The contact forces on particle i include all contributions from the neighboring particles and the boundary segments. For particles not in contact, $\mathbf{G}_{i,j} = 0$. The values of all the contact forces are iteratively determined such that the new distance between the particles originally in contact equals the size of the security zone,

$$d_{ij} = |\hat{\mathbf{X}}_i(t_{n+1}) - \hat{\mathbf{X}}_j(t_{n+1})| = R_i + R_j + \delta, \tag{75}$$

where R_i are radius of the ith particle, and δ is the thickness of the security zone. The procedure is applied iteratively, since the new particle positions may create new contact

points. The final values of the contact forces contribute to a "collision" force on each particle,

$$\mathbf{G}_i^{(c)} = \sum_{j=1,\, j\neq i}^{N+N_{\text{wall}}} \mathbf{G}_{i,j}, \tag{76}$$

which is introduced as the additional body forces in the calculations of the flow field and the particle motion.

Maury [51] further developed the idea presented above into a more solid mathematical framework. The objective of the collision model is to find a modified particle configuration set (positions and orientations)

$$Y = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N, \Theta_1, \Theta_2, \ldots, \Theta_N) \in \Re^{6N} \tag{77}$$

to minimize a functional

$$\Psi(Y) = \sum_{d_{ij}<\delta} (d_{ij}(Y) - \delta)^2, \tag{78}$$

where d_{ij} is the distance between the particles i and j. The optimum configuration set Y_e can be obtained by performing a steepest descent algorithm on $\Psi(Y)$. This strategy was developed for two-dimensional smooth bodies of arbitrary shape.

It should be noted that the collision strategies that use a security zone to prevent close contact of the particles tend to keep the particles farther apart than they ought to be, resulting in lower particle volume fractions in the fluid–solid mixture. The size of the security zone should be kept as small as possible. It is a balance of the accuracy of the numerical scheme and the computational cost.

12. SAMPLE APPLICATIONS

In this section we present results of some of the direct numerical simulations using the ALE particle mover. Most of the results presented are the works of Patankar [55] and Zhu [70].

12.1. *Sedimentation of a Single Sphere in a Tube*

To check the validity of the ALE particle mover, a number of tests were performed. Here, we shall present a few of them. All of the numerical results were obtained under the conditions that they are insensitive (to less than 1%) to further mesh refinement, to increase of the computation domain size, and to reduction of the time step.

The first test checks the drag on a sphere settling along the center of an infinitely long cylindrical tube filled with a Newtonian fluid. In the simulation the fluid inertia is turned off so that we can compare the numerical results with the exact Stokes flow solutions. We define the ratio of the sphere diameter d to the tube diameter D as the blockage ratio

$$\alpha = d/D, \tag{79}$$

and we define the wall factor

$$K = F_D/F_\infty \tag{80}$$

as the ratio of the terminal drag F_D exerted on the sphere in the tube to the drag F_∞ in the

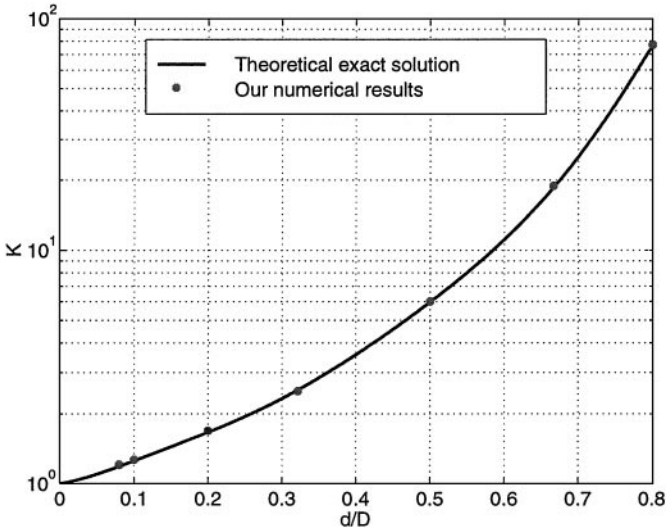

FIG. 7. Wall factor as a function of the blockage ratio for the settling of a sphere in a circular tube under Stokes flow conditions. Our numerical results and the theoretical ones from Haberman and Sayre [25] are compared.

infinite domain. In our numerical simulations, we calculate the terminal velocity V of a sphere released in a fluid of viscosity η. The drag on the sphere at the terminal velocity is determined by its effective weight,

$$F_{\mathrm{D}} = \frac{\pi}{6} d^3 (\rho_{\mathrm{s}} - \rho_{\mathrm{f}}) g, \tag{81}$$

and the drag F_∞ is given by the Stokes drag law,

$$F_\infty = 3\pi \eta V d. \tag{82}$$

The wall factor K is directly calculated using (81) and (82). Figure 7 compare the wall factors between the results of our numerical simulations and the theoretical ones from Haberman and Sayre [25] at various blockage ratios. The numerical results agree very well with Haberman and Sayre's [25] exact solution. The maximum deviation is less than 0.4%.

The next test checks the drag on a sphere settling at finite Reynolds numbers. Figure 8 shows the drag coefficient C_{D} as a function of the Reynolds number Re at a blockage ratio of $\alpha = 0.3125$. The drag coefficient and the Reynolds number of the flow are defined as

$$C_{\mathrm{D}} = \frac{F_{\mathrm{D}}}{\rho_{\mathrm{f}} V^2 \pi d^2/8} = \frac{4}{3}\frac{gd}{V^2}\left(\frac{\rho_{\mathrm{s}}}{\rho_{\mathrm{f}}} - 1\right) \tag{83}$$

and

$$\mathrm{Re} = \frac{\rho_{\mathrm{f}} V d}{\eta}, \tag{84}$$

where V is the terminal velocity of the sphere, and the drag on the sphere is calculated from (81). The comparison is given between our numerical results and the experimental measurements by McNown *et al.* [53]. Again our numerical results at finite Reynolds numbers agree with the experimental results very well. The maximum deviation between our numerical results and the experimental results of McNown *et al.* [53] is less than 0.5%.

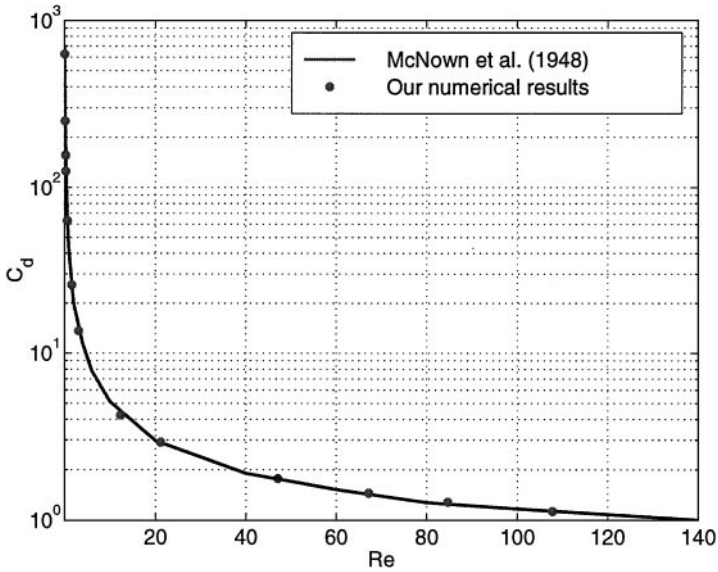

FIG. 8. Drag coefficient as a function of Reynolds number. Our numerical results and the experimental measurements of McNown *et al.* [53] are compared. The blockage ratio of the flow is $\alpha = 0.3125$.

The third test checks the transient behavior of a sphere settling along the center of an infinitely long circular tube filled with a viscoelastic fluid. The fluid is an Oldroyd-B fluid with viscometric properties similar to those of the M1 fluid (see Sridhar [66a]). The material properties for this fluid are $\eta = 30$ poises, $\lambda = 0.1$ s, $\rho_f = 0.868$ g cm^{-3}, and the viscosity ratio $\eta_2/\eta = 1/8$. In the test, the density of the sphere is $\rho_s = 3.581$ g cm^{-3} and the diameter of the sphere is $d = 2$ cm. Figure 9 shows the evolution of the settling velocity of the sphere

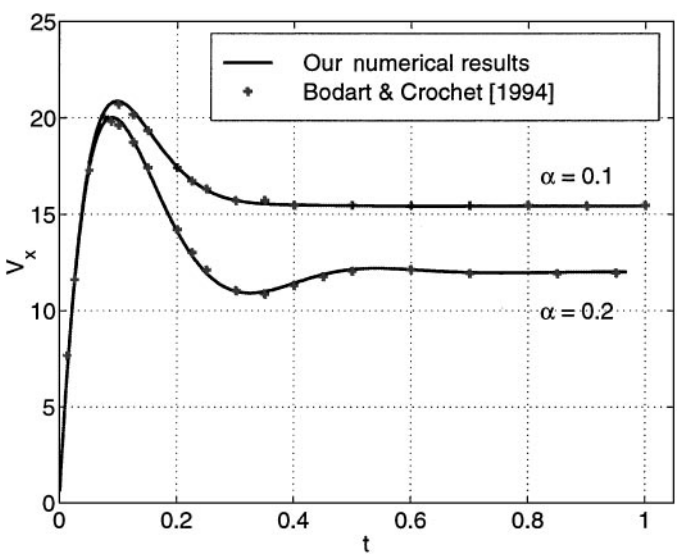

FIG. 9. Comparison of the settling velocities obtained from our full 3-D numerical simulations and from the 2-D axisymmetric simulations of Bodart and Crochet [7]. The sphere is released in a tube filled with a viscoelastic fluid. The blockage ratios for the simulations are $\alpha = 0.1$ and 0.2.

after it is released from rest. The comparison is between the results of our fully three-dimensional simulations and those of the two-dimensional, axisymmetric, finite-element calculations by Bodart and Crochet [7]. Our numerical results agree almost perfectly with their results. The maximum deviation between the results of our numerical simulations and those by Bodart and Crochet [7] is less than 0.5%.

12.2. *Migration of a Neutrally Buoyant Sphere in a Poiseuille Flow*

Direct numerical simulation is an ideal tool for studying the behavior of motion of a very few particles in a given fluid flow. It can be used to examine the mechanisms of particle migration, particle–particle interaction, particle–wall interaction, etc. Here we show an example of the migration of a particle in a Poiseuille flow.

Karnis *et al.* [45] performed numerous experiments on the migration of various particles, including spheres, rods, and disks, in a Poiseuille flow within a capillary tube. The ALE particle mover is capable of simulating the migration of spheres under the same conditions as those used in the experiments by Karnis *et al.* [45]. In our simulations, a neutrally buoyant sphere is released at a given initial radial position in a fully developed Poiseuille flow. Figure 10 presents two trajectories for the spheres released at radial positions of $r/R = 0.21$ and 0.68, where $R = D/2$ is the radius of the tube. In the simulations, the fluid properties are $\rho_f = 1.05\,\mathrm{g \cdot cm^{-3}}$ and $\eta = 1.2$ poises, the flow rate is $Q = 7.11 \times 10^{-2}\,\mathrm{cm^3/s}$, the tube diameter is $D = 0.4$ cm, and the sphere diameter is $d = 0.122$ cm. The two trajectories are compared with the measured ones from the experiments of Karnis *et al.* [45]. We see that our numerical results agree very well with the experimental ones with a deviation of less than 5%. From Fig. 10, we notice that the sphere released near the wall migrates inwards, while the sphere released near the tube center migrates outwards. Therefore, there exists an equilibrium position for the particle in a Poiseuille flow, which is the well-known

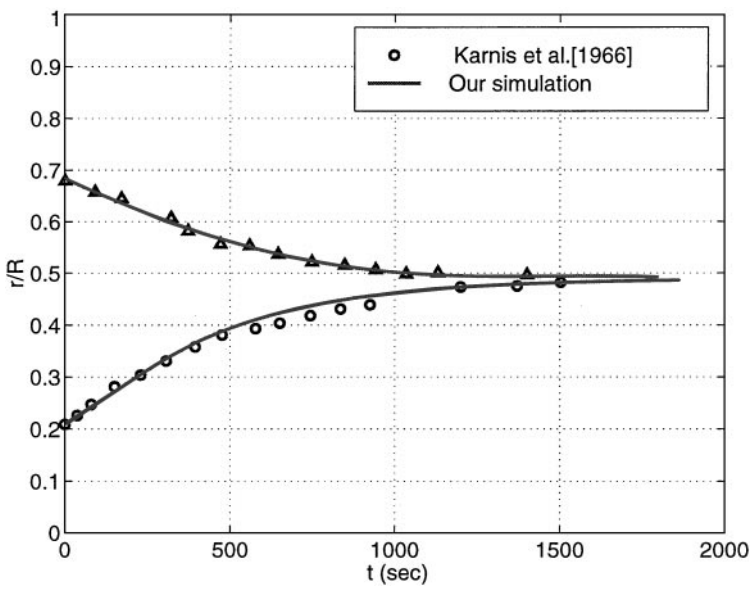

FIG. 10. Comparison of the migration trajectories of a neutrally buoyant sphere calculated in our simulations with the ones measured in the experiments of Karnis *et al.* [45].

Segré–Silberberg effect. Using the ALE particle mover, Zhu [70] is able to perform an extensively study on the effects of various parameters influencing the particle migration in this Poiseuille flow.

12.3. *Interaction of a Pair of Particles in a Newtonian Fluid: Drafting–Kissing–Tumbling*

One very important mechanism that controls the particle microstructure in flows of a Newtonian fluid is called "drafting, kissing, and tumbling" (Hu *et al.* [29]). There is a wake with low pressure at the back of a fluidized or sedimenting particle. If a trailing particle is caught in the wake of the leading one, it experiences a reduced drag and thus falls faster than the leading particle. This is called drafting, after the well-known bicycle racing strategy that is based on the same principle. The increased speed of fall impels the trailing particle into a kissing contact with the leading particle. Kissing particles form a long body that is unstable in a Newtonian fluid, when its line of centers is along the stream. The same couples which force a long body to float broadside-on cause kissing particles to tumble. Tumbling particles in a Newtonian fluid induce anisotropy of suspended particles since on the average the line of centers between particles must be across the stream.

Figure 11 displays a numerically simulated drafting–kissing–tumbling sequence. In the simulation, two spheres are dropped in tandem into an infinitely long tube filled with a Newtonian fluid; the fluid properties are selected as $\rho_f = 1\ \mathrm{g}\cdot\mathrm{cm}^{-3}$ and $\eta = 1$ poise, the particle density is $\rho_s = 2\ \mathrm{g}\cdot\mathrm{cm}^{-3}$, the particle diameter is $d = 2$ cm, and the tube diameter is $D = 20$ cm. This case corresponds to a particle Reynolds number of 22.

12.4. *Interaction of Particles in a Viscoelastic Fluid: Chaining*

The interaction of two particles in a viscoelastic fluid is quite different from that in a Newtonian fluid. The particles still undergo drafting and kissing. However, because broadside-on sedimentation of a long body is stable, kissing particles form a long body that is stable and will not tumble in a viscoelastic fluid under certain conditions (see Joseph [44]). Here we simulate the motion of two spheres of the same size released side-by-side into a tube filled with an Oldroyd-B fluid. The ratio of the sphere diameter to the tube

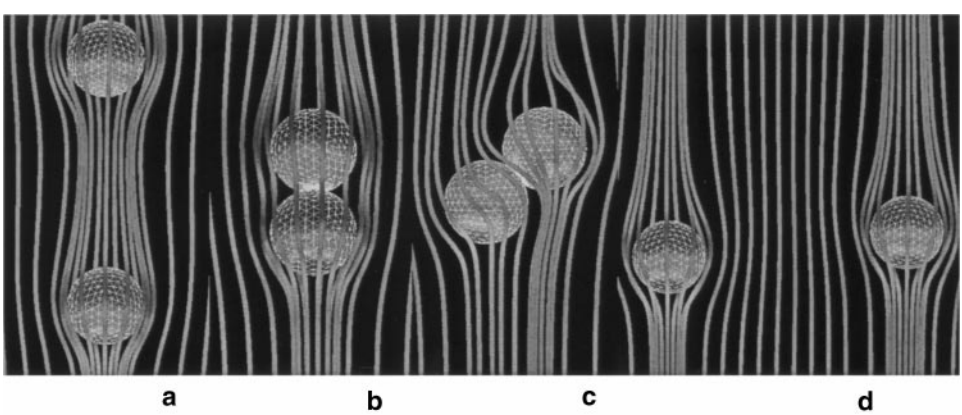

FIG. 11. Interaction between a pair of particles settling in a Newtonian fluid: drafting–kissing–tumbling. (a) streamlines at $t = 1.08$ s (drafting); (b) streamlines at $t = 1.45$ s (kissing); (c) streamlines at $t = 1.67$ s (tumbling); (d) streamlines at $t = 2.58$ s.

diameter is $\alpha = 0.2$. The spheres are symmetrically placed in the tube with respect to the axis of the tube with a separation distance of 1.5 times the sphere diameter. The material properties for this fluid are $\eta = 30$ poises, $\lambda = 0.1$ s, and $\rho_f = 0.868$ g cm^{-3}, and the viscosity ratio $\eta_2/\eta = 1/8$. In the test, the density ratio of the solid to the fluid is $\rho_s/\rho_f = 2$, and the diameter of the sphere is $d = 2$ cm.

Figure 12 displays the snapshots of the positions and orientations of the particles at various time instants. Figure 13 plots the particle trajectories, where y is the direction along the initial particle separation. It is observed that after the particles are released, they attract each other. This attraction is caused by the strong shear flow on the outside surfaces of the

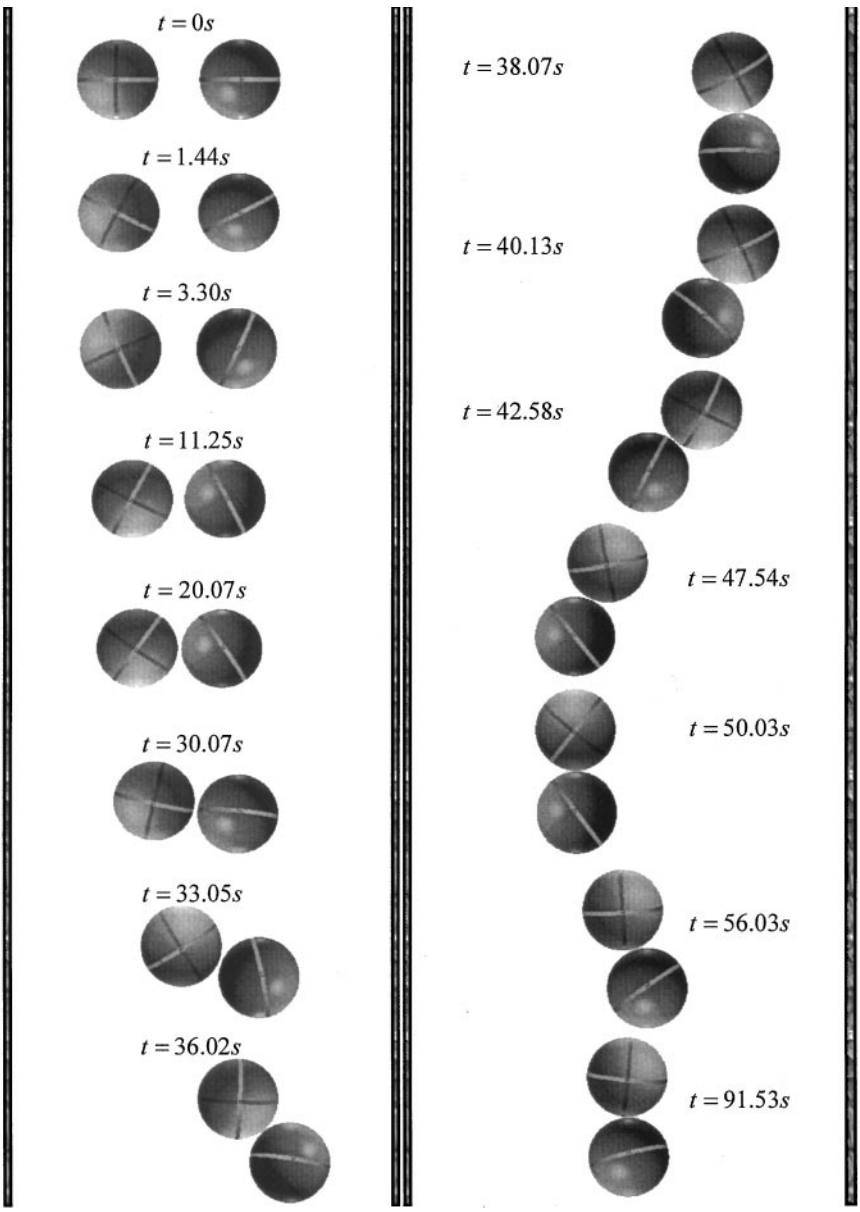

FIG. 12. Snapshots of the sedimentation of two spheres in a viscoelastic fluid.

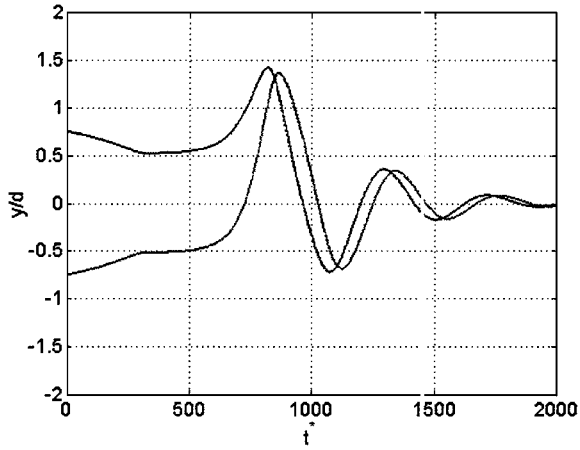

FIG. 13. Trajectories of two spheres released side-by-side in a cylindrical tube filled with a viscoelastic fluid.

particles, which produces a high pressure that pushes them toward each other. Once they are almost in contact, they momentarily form a long body. However, broadside-on orientation of a long body is unstable in a viscoelastic fluid, and the long body tends to turn longside along the direction of the fall. Thus two spheres turn and eventually form a chain as they settle.

If more particles are involved in the system, they tend to form longer chains. Because a longer chain falls faster than a shorter one, there exists a critical chain length, or a critical number of particles in a chain (Patankar and Hu [57]). The chaining of particles in a viscoelastic fluid only occurs when the elastic behavior of the fluid dominates. When both the fluid inertia and the fluid elasticity are important, the particles tend to form clusters (Patankar [55]).

12.5. *Lubrication in Pressure-Driven Particulate Flows*

Direct numerical simulation is very useful for studying the global behavior of a fluid–sparticle suspension. One can both examine the short-time rheology of the suspension for

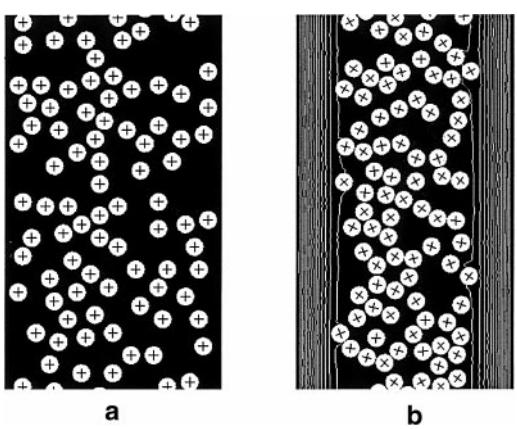

FIG. 14. Snapshots of the particle positions in a pressure-driven channel flow. White lines represent iso-lines of velocity in the x-direction. (a) Initial positions of the particles; (b) particle positions at $t = 16.4$ s.

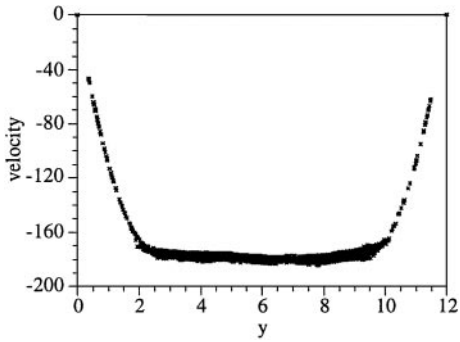

FIG. 15. Velocity component along the channel as a function of the coordinate across the channel.

a given microstructure (that is, the spatial distribution of the particles) and investigate the long-time evolution of the microstructures in the suspension. Here we show an example of investigating the long-time evolution of the particulate flow in a vertical channel driven by an externally applied pressure gradient. The applied pressure gradient either assists the gravity, causing the heavy particles to fall faster, or is sufficiently strong to pump the fluid and the particles against the gravity. The shear stress at the channel wall is sufficiently high to induce a velocity gradient in the fluid adjacent to the wall, causing the particles to migrate away from the wall. We have lubricated transport of the particulates (Patankar [55]). Figure 14 shows a typical case of this lubricated flow. It shows snapshots of 90 particles falling in a Newtonian fluid at the initial instant and at a later time. For this simulation, the channel width is 12 times the particle diameter, and the volume fraction of the particles is $\phi = 26.8\%$. The Reynolds number of the flow is

$$\mathrm{Re} = \frac{\rho_f V d}{\eta} = 12.26, \tag{85}$$

where we have used the slip velocity between the solid and fluid, $V = (\rho_s - \rho_f) g d^2 / 4\eta$. The nondimensional pressure gradient is defined as

$$C_p = \frac{1}{\phi g(\rho_s - \rho_f)} \frac{dp}{dx} = 2 \tag{86}$$

and the density ratio is $\rho_s/\rho_f = 1.1$. We observe that the particles migrate toward the center of the channel forming a prominent core. Figure 15 shows that the velocity profile of the fluid becomes blunt as the result of migration of the particles away from the wall and their concentration at the center of the channel. The core nearly falls like a rigid body so that the velocity of the fluid varies almost linearly from the walls to the particle-rich core owing to the absence of particles in that region. This motion can be considered to be similar to that established when a porous piston that occupies the zone of plug flow falls inside a channel.

13. SUMMARY

We have described a numerical method (the ALE particle mover) for simulations of fluid–solid flow systems. This method is based on a combined formulation of the fluid and particle

momentum equations. It uses the arbitrary Lagrangian–Eulerian (ALE) technique with a moving, unstructured, finite-element mesh to deal with the movement of the particles. In this method the moving finite-element mesh in the fluid flow and the particle positions are updated explicitly, while the fluid flow and the particle velocities are solved implicitly, at each time step. We have shown that this scheme is stable. A mesh movement and update strategy is described, and the remeshing criteria are discussed. Detailed schemes for the projection of the flow field from one mesh to another are presented. Different models of particle collision are also examined.

We next computed the sedimentation and migration of spheres in both Newtonian and viscoelastic fluids, and our results agree quantitatively with those in the literature. We also examined the interaction of the sedimenting particles in the Newtonian and viscoelastic fluids, and we observed the contrasting behavior of the particles: drafting–kissing–tumbling in a Newtonian fluid against drafting–kissing–chaining in a viscoelastic fluid. Last, we showed the long-time evolution of the microstructure of a fluid–particle suspension and demonstrated the rheological effect on the suspension.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation under KDI/NCC Grant NSF/CTS-9873236.

REFERENCES

1. C. K. Aidun and Y. Lu, Lattice Boltzmann simulation of solid suspensions with impermeable boundary, *J. Stat. Phys.* **81**, 49 (1995).
2. C. K. Aidun, Y. Lu, and E. Ding, Direct analysis of particulate suspensions with inertia using the discrete Boltzmann equation, *J. Fluid Mech.* **373**, 287 (1998).
3. M. J. Andrews and P. J. O'Rourke, The multiphase particle-in-cell (MP-PIC) method for dense particulate flows, *Int. J. Multiphase Flow* **22**, 379 (1996).
4. M. Behr and T. E. Tezduyar, Finite element solution strategies for large-scale flow simulations, *Comput. Meth. Appl. Mech. Eng.* **112**, 3 (1994).
5. O. Behrend, Solid–fluid boundaries in particle suspension simulations via the lattice Boltzmann method, *Phys. Rev. E* **52**, 1164 (1995).
6. R. B. Bird, R. C. Armstrong, and O. Hassager, *Dynamics of Polymeric Liquids* (Wiley-Interscience, New York, 1987), Vol. 1.
7. C. Bodart and M. J. Crochet, The time-dependent flow of a viscoelastic fluid around a sphere, *J. Non-Newtonian Fluid Mech.* **54**, 303 (1994).
8. J. F. Brady, Stockesian dynamics simulation of particulate flows, in *Particulate Two-Phase Flow*, edited by M.C. Roco (Butterworth–Heinemann, Stoneham, MA, 1993), p. 971.
9. J. F. Brady and G. Bossis, Stokesian dynamics, *Annu. Rev. Fluid Mech.* **20**, 111 (1988).
10. Deleted in proof.
11. Deleted in proof.
12. S. Chen and G. D. Doolen, Lattice Boltzmann method for fluid flows, *Annu. Rev. Fluid. Mech.* **30**, 329 (1998).
13. D. A. Drew and S. L. Passman, *Theory of Multicomponent Fluids* (Springer-Verlag, New York, 1999).
14. A. Esmaeeli and G. Tryggvason, Direct numerical simulations of bubbly flows, Part 2. Moderate Reynolds number arrays, *J. Fluid Mech.* **385**, 325 (1999).

15. L. S. Fan and C. Zhu, *Principles of Gas–Solid Flows* (Cambridge Univ. Press, Cambridge, UK, 1998).
16. J. Feng, H. H. Hu, and D. D. Joseph, Direct simulation of initial value problems for the motion of solid bodies in a Newtonian fluid. Part 1: Sedimentation, *J. Fluid Mech.* **261**, 95 (1994a).
17. J. Feng, H. H. Hu, and D. D. Joseph, Direct simulation of initial value problems for the motion of solid bodies in a Newtonian fluid. Part 2: Couette and Poiseuille flows, *J. Fluid Mech.* **277**, 271 (1994b).
18. J. Feng, P. Y. Huang, and D. D. Joseph, Dynamic simulation of the motion of capsules in pipelines, *J. Fluid Mech.* **282**, 233 (1995).
19. J. Feng, P. Y. Huang, and D. D. Joseph, Dynamic simulation of sedimentation of solid particles in an Oldroyd-B fluid, *J. Non-Newtonian Fluid Mech.* **63**, 63 (1996).
20. Deleted in proof.
21. Deleted in proof.
22. P. L. George, *Automatic Mesh Generation, Application to Finite Element Methods* (Wiley, New York, 1991).
23. D. Gidaspow, *Multiphase Flow and Fluidization Continuum and Kinetic Theory Description* (Academic Press, Boston, 1994).
24. R. Glowinski, T. W. Pan, T. I. Hesla, and D. D. Joseph, A distributed Lagrange multiplier/fictitious domain method for particulate flows, *Int. J. Multiphase Flow* **25**, 755 (1999).
25. W. L. Haberman and R. M. Sayre, Motion of rigid and fluid spheres in stationary and moving liquids inside cylindrical tubes, *David Taylor Model Basin Report* 1143 (1958).
26. P. Hansbo, The characteristic streamline diffusion method for the time-dependent incompressible Navier–Stokes equations, *Comput. Meth. Appl. Mech. Eng.* **99**, 171 (1992).
27. H. H. Hu, Motion of a circular cylinder in a viscous liquid between parallel plates, *Theor. Comput. Fluid Dyn.* **7**, 441 (1995).
28. H. H. Hu, Direct simulation of flows of solid–liquid mixtures, *Int. J. Multiphase Flow* **22**, 335 (1996).
29. H. H. Hu, A. Fortes, and D.D. Joseph, Experiments and direct simulations of fluid particle motions, *Int. Video J. Eng. Res.* **2**, 17 (1993).
30. H. H. Hu, D. D. Joseph, and M. J. Crochet, Direct simulation of fluid particle motions, *Theor. Comput. Fluid Dyn.* **3**, 285 (1992).
31. Deleted in proof.
32. P. Y. Huang, J. Feng, H. H. Hu, and D. D. Joseph, Direct simulation of the motion of solid particles in Couette and Poiseuille flows of viscoelastic fluids, *J. Fluid Mech.* **343**, 73 (1997).
33. P. Y. Huang, J. Feng, and D. D. Joseph, The turning couples on an elliptic particle settling in a vertical channel, *J. Fluid Mech.* **271**, 1 (1994).
34. P. Y. Huang, H. H. Hu, and D. D. Joseph, Direct simulation of the sedimentation of elliptic particles in Oldroyd-B fluids, *J. Fluid Mech.* **362**, 297 (1998).
35. A. Huerta and W. K. Liu, Viscous flow with large free surface motion, *Comput. Meth. Appl. Mech. Eng.* **69**, 277 (1998).
36. T. J. R. Hughes and G. M. Hulbert, Space-time finite element methods for elasto-dynamics: Formulations and error estimates, *Comput. Meth. Appl. Mech. Eng.* **66**, 339 (1988).
37. T. J. R. Hughes, W. K. Liu, and T. K. Zimmerman, Lagrangian–Eulerian finite element formulation for incompressible viscous flows, *Comput. Meth. Appl. Mech. Eng.* **29**, 329 (1981).
38. Deleted in proof.
39. M. Ishii, *Thermo-Fluid Dynamic Theory of Two-Phase Flows* (Eyrolles, Paris, 1975).
40. A. Johnson, *Mesh Generation and Update Strategies in Parallel Computation of Flow Problems with Moving Boundaries and Interfaces*, Ph.D. thesis (Univ. Minnesota, 1994).
41. A. Johnson and T. E. Tezduyar, Simulation of multiple spheres falling in a liquid-filled tube, *Comput. Meth. Appl. Mech. Eng.* **134**, 351 (1996).
42. A. Johnson and T. E. Tezduyar, 3D simulation of fluid–particle interactions with the number of particles reaching 100, *Comput. Meth. Appl. Mech. Eng.* **145**, 301 (1997).

43. A. Johnson and T. E. Tezduyar, Advanced mesh generation and update methods for 3D flow simulations, *Comput. Mech.* **23**, 130 (1999).
44. D. D. Joseph, Flow induced microstructure in Newtonian and viscoelastic fluids, in *Proc. 5th World Congress of Chem. Eng. Particle Technology Track, San Diego, July 14–18. AIChE J.* **6**, 3 (1996).
45. A. Karnis, H. L. Goldsmith, and S. G. Mason, The flow of suspensions through tubes, V. Inertial effects, *Can. J. Chem. Eng.* **44**, 181 (1966).
46. A. J. C. Ladd, Numerical simulations of particulate suspensions via a discretized Boltzmann equation. I. Theoretical foundation, *J. Fluid Mech.* **271**, 285 (1994a).
47. A. J. C. Ladd, Numerical simulations of particulate suspensions via a discretized Boltzmann equation. II. Numerical results, *J. Fluid Mech.* **271**, 311 (1994b).
48. A. J. C. Ladd, Sedimentation of homogeneous suspension of non-Brownian spheres, *Phys. Fluids* **9**, 491 (1997).
49. L. Little and Y. Saad, *Block LU Preconditioners for Symmetric and Nonsysmmetric Saddle Point Problem*, Univ. Minnesota Supercomputing Inst. Res. Rep. UMSI 99/104, 1999.
50. B. Maury, A many-body lubrication model, *C.R. Acad. Sci. Paris* **325**, Ser. I, 1053 (1997).
51. B. Maury, Direct simulations of 2D fluid-particle flows in biperiodic domains, *J. Comp. Phys.* **156**, 325 (1999).
52. J. B. McLaughlin, Numerical computation of particle–turbulent interaction, *Int. J. Multiphase Flow* **20**, 211 (1994).
53. J. S. McNown, H. M. Lee, M. B. McPherson, and S. M. Ebgez, *Proc. 7th Int. Congr. Appl. Mech.* **2**, 17 (1948).
54. T. Nomura and T. J. R. Hughes, An arbitrary Lagrangian–Eulerian finite element method for interaction of fluid and a rigid body, *Comput. Meth. Appl. Mech. Eng.* **95**, 115 (1992).
55. N. A. Patankar, *Numerical Simulation of Particulate Two-Phase Flow*, Ph.D. thesis (Univ. Pennsylvania, 1997).
56. N. Patankar and H. H. Hu, Two-dimensional periodic mesh generation, in *Proceedings of the 5th International Conference on Numerical Grid Generation in Computational Field Simulations* (1996), p. 1175.
57. N. Patankar and H. H. Hu, A numerical investigation of the detachment of the trailing particle from a chain sedimenting in Newtonian and viscoelastic fluids, *J. Fluid Engineering* **122**, (2000).
58. Deleted in proof.
59. D. Qi, Lattice-Boltzmann simulations of particles in non-zero-Reynolds-number flows, *J. Fluid Mech.* **385**, 41 (1999).
60. Y. Saad, *Iterative Methods for Sparse Linear Systems* (PWS Publ. Co., Boston, 1996).
61. A. S. Sangani and A. K. Didwania, Dynamic simulations of flows of bubbly liquids at large Reynolds numbers, *J. Fluid Mech.* **250**, 307 (1993).
62. A. S. Sangani and A. Prosperetti, Numerical simulation of the motion of particles at large Reynolds numbers, in *Particulate Two-Phase Flow*, edited by M. C. Roco (Butterworth-Heinemann, Stoneham, MA, 1993), p. 971.
63. V. Sarin and A. Sameh, An efficient iterative method for the generalized Stokes problem, *SIAM J. Sci. Comput.* **19**, 206 (1998).
64. Deleted in proof.
65. P. Singh, D. D. Joseph, T. I. Hesla, R. Glowinski, and T.-W. Pan, A distributed Lagrange multiplier/fictitious domain method for viscoelastic particulate flows, *J. Non-Newtonian Fluid Mech.* **91**, 165 (2000).
66. D. M. Snider, P. J. O'Rourke, and M. J. Andrews, Sediment flow in inclined vessels calculated using a multiphase particle-in-cell model for dense particle flows, *Int. J. Multiphase Flow* **24**, 1359 (1998).

66a. T. Sridhar, An overview of the project MI, *J. Non-Newtonian Fluid Mech.* **35**, 85 (1990).

67. T. E. Tezduyar, J. Liou, and M. Behr, A new strategy for finite element computations involving moving boundaries and interfaces—The DSD/ST procedure. I. The concept and the preliminary numerical tests, *Comput. Meth. Appl. Mech. Eng.* **94**, 339 (1992a).

68. T. E. Tezduyar, J. Liou, M. Behr, and S. Mittal, A new strategy for finite element computations involving moving boundaries and interfaces—The DSD/ST procedure. II. Computation of free-surface flows, two-liquid flows, and flows with drafting cylinders, *Comput. Meth. Appl. Mech. Eng.* **94**, 353 (1992b).
69. D. Z. Zhang and A. Prosperetti, Averaged equations for inviscid disperse two-phase flow, *J. Fluid Mech.* **267**, 185 (1994).
70. M.-Y. Zhu, *Direct Numerical Simulation of Particulate Flow of Newtonian and Viscoelastic Fluids*, Ph.D. thesis (Univ. Pennsylvania, 2000).