Journal of Computational Physics 227 (2008) 8776–8791

Contents lists available at ScienceDirect

Journal of Computational Physics

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

An explicit finite difference scheme with spectral boundary conditions for particulate flows

A. Perrin, H.H. Hu *

Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, 229 Towne Building, 220 S. 33rd Street, Philadelphia, PA 19104, United States

ARTICLE INFO

Article history:
Received 11 May 2007
Received in revised form 23 January 2008
Accepted 9 June 2008
Available online 20 June 2008

Keywords:
Direct numerical simulation
Explicit finite difference methods
Particulate flows

ABSTRACT

We present an explicit finite difference scheme for solving two-dimensional particulate flow problems with a special treatment of the boundary conditions on the particle surface based on spectral solutions to the Stokes equations. This scheme allows for accurate solution of particulate flows up to a particle Reynolds number of one hundred on coarse grids (10–20 grid spacings per particle diameter). The coarse grid provides additional computational benefit by allowing for larger time steps required by the stability constraint. The method is validated and demonstrated through a number of examples, which include flow over a stationary cylinder, a cylinder moving with constant velocity, sedimentation of a free particle, the drafting, kissing, and tumbling of two particles, and 248 particles falling in a closed box.

1. Introduction

Direct numerical simulation of particulate flows, which resolves the flow field around each particle, remains a computationally challenging problem. Finite element methods based on a moving unstructured mesh, such as the Particle-Mover arbitrary Lagrangian–Eulerian (ALE) method of Hu [9,8,11], or Johnson and Tezduyars stabilized space- time method [12], use meshes that conform to the shape of the particles, offer local mesh refinement around the particle surface, and are shown to be efficacious for small numbers of particles at moderate Reynolds numbers. However, they become computationally expensive for systems of dense particle suspensions due to frequent remeshing and projection.

Glowinsky's distributed Lagrange multiplier method illustrates some of the advantages of a fixed, uniform grid. In the distributed Lagrange multiplier method [6], the fluid flow equations are solved both outside and inside the particles, and a distributed Lagrange multiplier is introduced to enforce the rigid-body motion inside the particle boundaries. The time integration of the governing equations is performed using an operator splitting method. The use of a fixed and uniform grid on the combined fluid-solid domain allows for fast and efficient flow solvers.

The lattice Boltzmann method (for example, Ladd [14], Behrend [2], Aidun et al. [1], or Qi [20]) has also been shown to be effective for particulate flows. The lattice Boltzmann method relaxes the condition of incompressibility and allows the governing equations to be integrated explicitly on a fixed and uniform grid. The idea of the method is to devise a mesoscopic kinetic equation which is marched explicitly in time according to simple rules, from which macroscopic quantities can be recovered by averaging.

Inspired by the simplicity of fixed uniform grids and explicit time integration for slightly compressible flows, the authors formulated a MacCormack method for particulate flows with circular cylinders ("particles") in two dimensions [17]. The

* Corresponding author.
E-mail addresses: aperrin@seas.upenn.edu (A. Perrin), hhu@seas.upenn.edu (H.H. Hu).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.06.007

MacCormack method is a two step predictor-corrector explicit scheme. The spatial derivatives are discretized with opposite one-sided finite differences in the predictor and corrector stages. To enforce boundary conditions on the particle surfaces, a local Taylor series expansion was used to enforce no-slip approximately by assigning velocities to the grid points nearest the surface such that velocity on the surface would be zero. The stresses were numerically integrated on the particle surfaces to evaluate the hydrodynamic forces and moments acting on the particles, and the particles were moved accordingly. The scheme involved no matrix inversions and each time step was cheap to compute. However, it was found that a mesh of at least 20 grid spacings across the particle diameter was needed to obtain smooth pressures on the particle surfaces. The dense mesh also requires a smaller time step because of the stability condition for the explicit scheme. The effort to relieve this bottleneck led to a different implementation of boundary condition on the particle surfaces, inspired by Takagi et al. [22], that exactly satisfies the no-slip condition by means of a spectral expansion technique.

Prosperetti and Oguz [19] developed the PHYSALIS method, which in their initial paper simulated spheres in potential flow by matching an analytical solution of the Laplace equation for the velocity potential near the spheres' surfaces to a finite difference solution of the same equation in the bulk fluid. Takagi et al. [22] extended the method to deal with the incompressible Navier–Stokes equations around stationary cylinders by using the fact that close to the cylinder surfaces there is a region where inertial effects are small, so the equations for Stokes flow provide a reasonable approximation. A spectral solution to the Stokes equations around the cylinder could then be matched to a finite difference solution of the full Navier–Stokes equations while exactly satisfying the no-slip condition on the particle surface. Zhang and Prosperetti described the implementation for two-dimensional moving cylinders in [25], and extended PHYSALIS to the fully three-dimensional Navier–Stokes equations with moving spherical particles [26]. The Navier–Stokes flow solver in PHYSALIS is implicit in time, using Crank-Nicholson for the diffusive terms and Adams–Bashforth for the convective terms. At each time step, two sequences of iterations are performed, an outer iteration to transfer information between the spectral solution around the particles and the bulk flow, and an inner iteration to enforce incompressibility and determine the pressure.

In this paper, we present an explicit Lax–Wendroff method for solving particulate flow problems with a special treatment of the boundary conditions on the particle surface based on the spectral solutions to the Stokes equations. The Lax–Wendroff scheme is formally identical to the prior MacCormack scheme for convection-diffusion equations, but it is a one-step method rather than a predictor-corrector scheme. Consequently, the boundary conditions need only be applied once per time step. Besides this simplification, we found that the one-sided finite differences used in the MacCormack scheme introduced an asymmetry in the forces, leading to lift at low Reynolds numbers when the grid was relatively coarse. The Lax–Wendroff scheme uses only central differences, which eliminates the asymmetry.

This paper is divided into a section describing the finite difference scheme, a section explaining the implementation of the boundary conditions on the particle surfaces, and a series of examples for code validation and demonstration.

2. Explicit Lax–Wendroff scheme

In Cartesian coordinates, the component form of the continuity equation and compressible Navier–Stokes equations in two dimensions can be written as

$$\frac{\partial \rho}{\partial t}+\frac{\partial(\rho u)}{\partial x}+\frac{\partial(\rho v)}{\partial y}=0, \tag{1}$$

$$\frac{\partial}{\partial t}(\rho u)+\frac{\partial}{\partial x}(\rho u^2)+\frac{\partial}{\partial y}(\rho v u)=\rho g_x-\frac{\partial p}{\partial x}+\mu\nabla^2 u+\frac{\mu}{3}\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right), \tag{2}$$

$$\frac{\partial}{\partial t}(\rho v)+\frac{\partial}{\partial x}(\rho u v)+\frac{\partial}{\partial y}(\rho v^2)=\rho g_y-\frac{\partial p}{\partial y}+\mu\nabla^2 v+\frac{\mu}{3}\frac{\partial}{\partial y}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right) \tag{3}$$

with the equation of state $p=c^2\rho$, where c is the speed of sound in the medium. As long as the flows are limited to low Mach numbers (the ratio of the flow speed to the sound speed) and the conditions are almost isothermal, the solution to this set of equations should approximate the incompressible limit (see [13]).

Eqs. (1)–(3) can be cast into the vector form by introducing $\mathbf{U}=(\rho,\rho u,\rho v)$ which contains the update variables, and fluxes **E** and **F**, which are functions of **U** and its spatial derivatives as,

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{E}}{\partial x}+\frac{\partial \mathbf{F}}{\partial y}=0. \tag{4}$$

System (4) was discretized in the MacCormack scheme [17] as,

$$\begin{aligned}\mathbf{U}^*_{i,j}&=\mathbf{U}^n_{i,j}-\frac{\Delta t}{\Delta x}\left(\mathbf{E}^n_{i+1,j}-\mathbf{E}^n_{i,j}\right)-\frac{\Delta t}{\Delta y}\left(\mathbf{F}^n_{i,j+1}-\mathbf{F}^n_{i,j}\right),\quad \text{(Predictor)}\\ \mathbf{U}^{n+1}_{i,j}&=\frac{1}{2}\left[\mathbf{U}^n_{i,j}+\mathbf{U}^*_{i,j}-\frac{\Delta t}{\Delta x}\left(\mathbf{E}^*_{i,j}-\mathbf{E}^*_{i-1,j}\right)-\frac{\Delta t}{\Delta y}\left(\mathbf{F}^*_{i,j}-\mathbf{F}^*_{i,j-1}\right)\right].\quad \text{(Corrector)}\end{aligned} \tag{5}$$

The spatial derivatives in the MacCormack scheme are discretized with opposite one-sided finite differences in the predictor and corrector stages, for example forward differences in the predictor and backward in the corrector as shown in (5). It is found that the one-sided finite differences used in the MacCormack scheme introduce asymmetry in the flow field and the resulting hydrodynamics forces when flow is solved on relatively coarse grids.

The explicit Lax–Wendroff scheme [15] is a one-step method which is formally identical to the explicit MacCormack scheme for convection-diffusion equations away from boundaries. It can be developed by Taylor expanding **U** to second order in time and eliminating $\frac{\partial \mathbf{U}}{\partial t}$ and $\frac{\partial^2 \mathbf{U}}{\partial t^2}$ with (4) and $\frac{\partial}{\partial t}$ (4). This results in an update scheme containing only spatial derivatives, which are approximated as second order central differences:

$$\begin{aligned}\mathbf{U}_{i,j}^{n+1} &= \mathbf{U}_{i,j}^{n} + \Delta t \frac{\partial \mathbf{U}}{\partial t}\bigg|_{i,j}^{n} + \frac{1}{2}\Delta t^2 \frac{\partial^2 \mathbf{U}}{\partial t^2}\bigg|_{i,j}^{n} \\ &= \mathbf{U}_{i,j}^{n} + \Delta t[\mathbf{G}]|_{i,j}^{n} + \frac{1}{2}\Delta t^2 \left[\frac{\partial \mathbf{S}}{\partial x} + \frac{\partial \mathbf{T}}{\partial y}\right]\bigg|_{i,j}^{n}\end{aligned} \tag{6}$$

$$\mathbf{G} = -\left(\frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y}\right), \mathbf{S} = \frac{\partial \mathbf{E}}{\partial \mathbf{U}}\mathbf{G}, \mathbf{T} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}}\mathbf{G} \tag{7}$$

At the start of each time step, **G**, **S** and **T** are computed at every grid point, because they need to be known to compute Eq. (6).

Applying the Lax–Wendroff scheme to the compressible Navier–Stokes Eqs. (1)–(3) and replacing the pressure with $p = c^2\rho$, we have,

$$\rho_{i,j}^{n+1} = \rho_{i,j}^{n} - \Delta t F_{i,j}^{n} + \frac{\Delta t^2}{2}\left[\begin{array}{l}\frac{\partial^2}{\partial x^2}[\rho(c^2+u^2)] + \frac{\partial^2}{\partial y^2}[\rho(c^2+v^2)] \\ +2\frac{\partial^2}{\partial x \partial y}(\rho u v) - \frac{4}{3}\mu\nabla^2(div\mathbf{u})\end{array}\right]_{i,j}^{n} \tag{8}$$

$$(\rho u)_{i,j}^{n+1} = (\rho u)_{i,j}^{n} - \Delta t G_{i,j}^{n} + \frac{\Delta t^2}{2}\left[\begin{array}{l} c^2\left(\frac{\partial^2 \rho u}{\partial x^2} + \frac{\partial^2 \rho v}{\partial x \partial y}\right) + \frac{\partial P}{\partial x} + \frac{\partial R}{\partial y} \\ +\frac{\mu}{3}\left(4\frac{\partial^2 S}{\partial x^2} + 3\frac{\partial^2 S}{\partial y^2} + \frac{\partial^2 T}{\partial x \partial y}\right)\end{array}\right]_{i,j}^{n} \tag{9}$$

$$(\rho v)_{i,j}^{n+1} = (\rho v)_{i,j}^{n} - \Delta t H_{i,j}^{n} + \frac{\Delta t^2}{2}\left[\begin{array}{l} c^2\left(\frac{\partial^2 \rho u}{\partial x \partial y} + \frac{\partial^2 \rho v}{\partial y^2}\right) + \frac{\partial R}{\partial x} + \frac{\partial Q}{\partial y} \\ +\frac{\mu}{3}\left(3\frac{\partial^2 T}{\partial x^2} + 4\frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 S}{\partial x \partial y}\right)\end{array}\right]_{i,j}^{n} \tag{10}$$

where the following variables must be computed at each grid point before Eqs. (8)–(10) are applied:

$$F = \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} \tag{11}$$

$$G = c^2\frac{\partial \rho}{\partial x} + \frac{\partial}{\partial x}(\rho u u) + \frac{\partial}{\partial y}(\rho v u) - \frac{\mu}{3}\left(4\frac{\partial^2 u}{\partial x^2} + 3\frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 v}{\partial x \partial y}\right) \tag{12}$$

$$H = c^2\frac{\partial \rho}{\partial y} + \frac{\partial}{\partial x}(\rho u v) + \frac{\partial}{\partial y}(\rho v v) - \frac{\mu}{3}\left(3\frac{\partial^2 v}{\partial x^2} + 4\frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 u}{\partial y \partial x}\right) \tag{13}$$

$$P = 2uG - u^2F \tag{14}$$

$$Q = 2vH - v^2F \tag{15}$$

$$R = vG + uH - uvF \tag{16}$$

$$S = \frac{-G + uF}{\rho} \tag{17}$$

$$T = \frac{-H + vF}{\rho} \tag{18}$$

All spatial derivatives were approximated on a uniform grid as second order central differences, except on the boundaries of the computational domain, where one-sided second order differences were used. For example,

Central differences

$$\frac{\partial(\cdot)}{\partial x}\bigg|_i \to \frac{(\cdot)_{i+1} - (\cdot)_{i-1}}{2\Delta x} + O(\Delta x^2) \tag{19}$$

$$\frac{\partial^2(\cdot)}{\partial x^2}\bigg|_i \to \frac{(\cdot)_{i+1} - 2(\cdot)_i + (\cdot)_{i-1}}{\Delta x^2} + O(\Delta x^2) \tag{20}$$

One-sided differences

$$\frac{\partial(\cdot)}{\partial x}\bigg|_i \to \frac{-(\cdot)_{i+2} + 4(\cdot)_{i+1} - 3(\cdot)_i}{2\Delta x} + O(\Delta x^2) \tag{21}$$

$$\frac{\partial^2(\cdot)}{\partial x^2}\bigg|_i \to \frac{-(\cdot)_{i+3} + 4(\cdot)_{i+2} - 5(\cdot)_{i+1} + 2(\cdot)_i}{\Delta x^2} + O(\Delta x^2). \tag{22}$$

The outflow boundary conditions were,

$$
\begin{aligned}
(\rho u)_{i,j}^{n+1} &= \frac{1}{3}\left(4(\rho u)_{i-1,j}^{n+1} - (\rho u)_{i-2,j}^{n+1}\right)\\
(\rho v)_{i,j}^{n+1} &= \frac{1}{3}\left(4(\rho v)_{i-1,j}^{n+1} - (\rho v)_{i-2,j}^{n+1}\right)
\end{aligned}
\tag{23}
$$

corresponding to $\frac{\partial \rho \mathbf{u}}{\partial x} = 0$, where x is normal to the boundary. For brevity, body force terms in the momentum equations are neglected here.

Tannehill, Anderson and Pletcher [24] give the following semi-empirical stability criterion for the explicit MacCormack scheme (which is very similar to the present Lax–Wendroff),

$$
\Delta t \leqslant \frac{\sigma}{(1+2/Re_\Delta)}\left[\frac{|u|}{\Delta x} + \frac{|v|}{\Delta y} + c\sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}}\right]^{-1},
\tag{24}
$$

where σ is a safety factor (≈ 0.9), $Re_\Delta = \min\left(\frac{\rho|u|\Delta x}{\mu}, \frac{\rho|v|\Delta x}{\mu}\right)$ is the minimum mesh Reynolds number. This criterion suggests that the MacCormack and Lax–Wendroff schemes will be inefficient for flows with small Reynolds numbers, since excessively small time steps are required. However, it was found that the condition (24) is quite conservative for flows with small mesh Reynolds numbers [17]. Condition (24) was further simplified in the regime of moderate flow Reynolds numbers ($Re = 1$ to 100), small Mach numbers, and $\Delta x = \Delta y$, to the following CFL condition,

$$
\Delta t \leqslant \frac{0.5\Delta x}{c}.
\tag{25}
$$

These results are similar to what is found for lattice Boltzmann methods (see discussion of the issue in [16], for instance).

In simulation of particulate flows, the natural time step for the particle motion is roughly determined by how far the particles are allowed to move during that time step, which can be described by a CFL condition (25) based on the particle velocity with a similar safety factor. The stability condition in (25) is not very restrictive for such flows, since the ratio of this time step to the time step for the particle motion is proportional to the Mach number.

3. Boundary conditions on the particle surface

For a stationary particle, the fluid velocity is small in a region next to the particle and exactly zero at the particle surface. Therefore, a region exists close to the particle in which the local Reynolds number is much smaller than one. By the same logic, the local Mach number will also tend to zero near the particle surface. One can approximate the flow in this region as an incompressible Stokes flow, which can be represented by a truncated spectral expansion whose eigenfunctions exactly satisfy the no-slip condition on the particle surface ([22]).

For the Stokes flow around a circular cylinder, the biharmonic equation for the stream function is solved (see [22] or [7]) to obtain a general solution of the form,

$$
\psi = (s^2 - 2\ln s - 1)A_0 + \sum_{n=1}^{\infty}\begin{bmatrix}\psi_n(s)(A_n\cos n\theta + \widetilde{A}_n\sin n\theta)\\ +\tilde{\psi}_n(s)(B_n\cos n\theta + \widetilde{B}_n\sin n\theta)\end{bmatrix}
\tag{26}
$$

where ψ is the streamfunction, $s = r/a$ is the radial coordinate normalized by the particle radius, θ is the angular coordinate, and the basis functions are given by

$$
\psi_n = ns^{n+2} - (n+1)s^n + s^{-n}
\tag{27}
$$

and

$$
\tilde{\psi}_n = \begin{cases} s\ln s - \frac{s}{2} + \frac{1}{2s} & n = 1\\ ns^{-n+2} - (n-1)s^{-n} - s^n & n > 1\end{cases}.
\tag{28}
$$

The pressure and vorticity are given by,

$$
\begin{aligned}
p &= p_0 + \frac{\mu}{a^2}[8s(-A_1\sin\theta + \widetilde{A}_1\cos\theta) + 2s^{-1}(B_1\sin\theta - \widetilde{B}_1\cos\theta)]\\
&\quad + \frac{\mu}{a^2}\sum_{n=2}\infty\begin{bmatrix}4n(n+1)s^n(-A_n\sin(n\theta) + \widetilde{A}_n\cos(n\theta))\\ +4n(n-1)s^{-n}(\widetilde{B}_n\cos(n\theta) - B_n\sin(n\theta))\end{bmatrix}
\end{aligned}
\tag{29}
$$

$$
\omega = -4a^{-2}A_0 - a^{-2}\sum_{n=1}^{\infty}\begin{bmatrix}((\psi_n)'' + s^{-1}(\psi_n)' - s^{-2}n^2\psi_n)\begin{pmatrix}A_n\cos n\theta\\ +\widetilde{A}_n\sin n\theta\end{pmatrix}\\ ((\tilde{\psi}_n)'' + s^{-1}(\tilde{\psi}_n)' - s^{-2}n^2\tilde{\psi}_n)\begin{pmatrix}B_n\cos n\theta\\ +\widetilde{B}_n\sin n\theta\end{pmatrix}\end{bmatrix}.
\tag{30}
$$

When truncated to N modes, the A_n's, B_n's, $\widetilde{A}_n$'s, $\widetilde{B}_n$'s and p_0 yield $4N+2$ unknown coefficients which must be determined by doing a least squares fit to the external flow, where the full Navier–Stokes equations are solved. Since B_1, $\widetilde{B}_1$, and A_0 are directly proportional to the hydrodynamic forces and the moment acting on the cylinder, this computation avoids a surface integration that would otherwise be necessary, as in the MacCormack implementation.

Next, a set of grid points roughly one grid spacing from the particle surface (the "cage" in the language of [22]) is chosen for the least squares fit. The algorithm described in [17] for drawing the particle on the grid was used here to choose the cage points. The points need not be chosen in any particular order. Fig. 1 shows the particle, the cage points (circles), and the points to be calculated using the analytical solution (triangles).

It now remains to describe the least-squares fit to the outer flow. The choice of which quantities to fit can make a dramatic difference to the results. It was found that fitting the velocity components directly yields poor results because the low velocity near the particle produces a low condition number of the pseudoinverse (a poor "signal to noise ratio") resulting in a bad fit. Based on the suggestion in [22], the vorticity and pressure were fitted. The motivation for using vorticity is that it is generated on the particle surface, so the closer the cage gets to the particle surface, the stronger the "signal"—the opposite of what happens if one uses the velocity components. The cage should be near the surface because otherwise the Stokes flow approximation will not be valid.

If the series in Eqs. (29) and (30) are truncated to N modes, and there are P cage points, with two equations per point (one vorticity and one pressure), then one obtains an overdetermined system of $2P$ equations for $4N+2$ unknowns which can be solved by least squares to obtain a best fit. In [22] a method based on Singular Value Decomposition was used, however this was found to be too computationally expensive for the current work, so we do Cholesky decomposition of the pseudoinverse instead.

The fit itself is done at the end of each time step. To calculate the vorticity on the cage points, one can evaluate $\left(\frac{\partial v}{\partial x}-\frac{\partial u}{\partial y}\right)^{n+1}$ using one-sided differences directed into the bulk fluid, which has already been updated to t_{n+1}. However, this option introduces asymmetries into the problem, as we found in the MacCormack scheme. As an alternative, we calculate ω^n with central differences using,

$$\omega^n_{i,j}=\frac{v^n_{i+1,j}-v^n_{i-1,j}}{2\Delta x}-\frac{u^n_{i,j+1}-u^n_{i,j-1}}{2\Delta y} \tag{31}$$

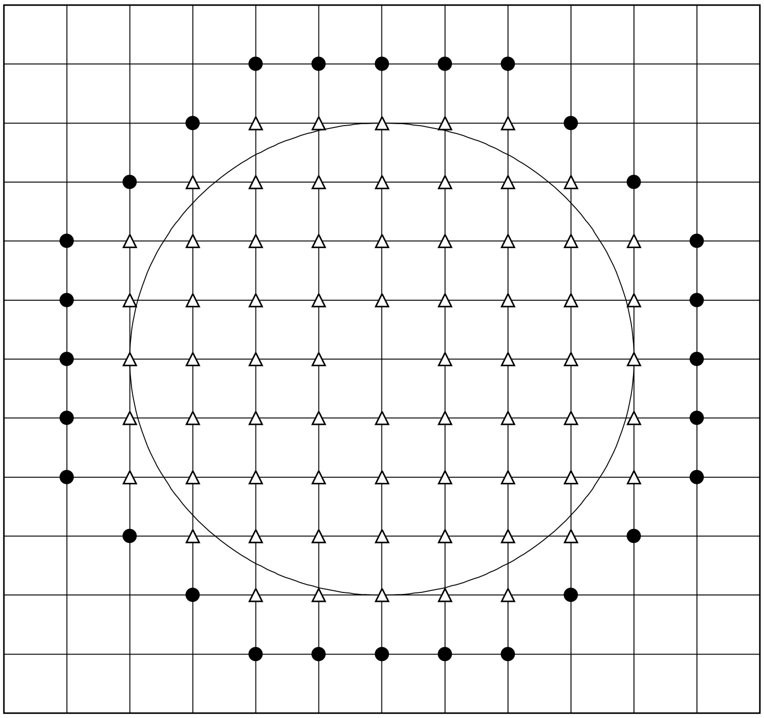

Fig. 1. An example of a particle with eight grid spacings across its diameter, with the cage points marked with circles. Once the coefficients are determined by the described fitting, the value of velocity at grid points within the cage are evaluated from the analytical solution. These grid points are marked with triangles. There is a singularity in the analytical solution at the particle's center, so that point is omitted.

then extrapolate the vorticity at each cage point p to the next time step using the vorticity equation,

$$\begin{aligned}
\frac{\partial\omega}{\partial t} &= -\frac{\partial(\omega u)}{\partial x} - \frac{\partial(\omega v)}{\partial y} + \frac{\mu}{\rho}\left[\begin{array}{l} \nabla^2\omega + \frac{1}{\rho}\frac{\partial\rho}{\partial y}\left(\nabla^2 u + \frac{1}{3}\frac{\partial}{\partial x}(\nabla\cdot\mathbf{u})\right) \\ -\frac{1}{\rho}\frac{\partial\rho}{\partial x}\left(\nabla^2 v + \frac{1}{3}\frac{\partial}{\partial y}(\nabla\cdot\mathbf{u})\right) \end{array}\right], \\
\omega_p^{n+1} &= \omega_p^n + \frac{1}{2}\left[3\Delta t\left(\frac{\partial\omega}{\partial t}\right)_p^n - \Delta t\left(\frac{\partial\omega}{\partial t}\right)_p^{n-1}\right] \quad p = 1,\ldots,P
\end{aligned} \tag{32}$$

where all derivatives are evaluated with second order central differences. The entire velocity field is known at time n, so differencing inside the cage is not a problem. In the moving particle case, the velocities in the update should be taken in the particle's reference frame.

It is important to emphasize that the vorticity equation isn't being marched in time—at time n, vorticity is calculated directly from the velocity, as (31), and then these values are advanced to time $n+1$ with Eq. (32), instead of repeatedly applying (32) which would accumulate integration error.

To apply the boundary conditions on the surface of a moving particle, it is necessary to make a change of variables between the laboratory coordinates and a frame attached to the particle. The transformation is described in Takagi et al. [22] and repeated here for completeness. If $\mathbf{x}$ is the position relative to the particle center, $\mathbf{U}$, p, $\mathbf{w}$, and $\mathbf{\Omega}$ are the fluid velocity, pressure, particle velocity, and particle angular velocity in the inertial frame, and $\tilde{\mathbf{u}}$ and $\tilde{p}$ are the fluid velocity and pressure in the particle frame, then the change of variables will be,

$$\begin{aligned}
\mathbf{U} &= \tilde{\mathbf{u}} + \frac{|\mathbf{x}|^4 - a^4}{8v|\mathbf{x}|^2}(\dot{\mathbf{\Omega}} \times \mathbf{x}) + \mathbf{w} + \mathbf{\Omega} \times \mathbf{x} \\
p &= \frac{1}{2}\rho_0(\mathbf{\Omega} \times \mathbf{x})^2 - \rho_0\dot{\mathbf{w}}\cdot\mathbf{x} + \tilde{p}.
\end{aligned} \tag{33}$$

Dots are time derivatives following the particle, ρ_0 is fluid density, $v = \mu/\rho_0$ is kinematic viscosity, and a is the particle radius. The drag, lift, and moment on the particle per unit length in the laboratory reference frame are, respectively,

$$\begin{aligned}
F_x &= \frac{4\pi\mu}{a}\widetilde{B}_1 + \rho_0\pi a^2\dot{w}_x \\
F_y &= -\frac{4\pi\mu}{a}B_1 + \rho_0\pi a^2\dot{w}_y \\
M &= -8\pi\mu A_0 + \rho_0\pi a^4\dot{\Omega}.
\end{aligned} \tag{34}$$

where θ is measured from the x-axis in the eigenfunction expansions (26)–(30).

One of the major advantages of this code is its low storage requirements, which may be estimated as follows. In the finite difference portion of the code, there are 16 variables stored at each gridpoint, named: *F*, *G*, *H*, *P*, *Q*, *R*, *S*, *T*, *rho*, *u*, *v*, *rhoU*, *rhoV*, and *rho*2, *rhoU*2, *rhoV*2. The last three are for the updated values of density and momentum. u and v are reused for both times n and $n+1$.

For particles of diameter D, roughly $\pi D/\Delta x$ cage points have to be stored, and my implementation stores six values for each cage point (a grid point index, the cosine and sine of the angle the cage point makes with the horizontal relative to the particle center, the distance of the cage point relative to the particle center, the vorticity, and the pressure). Also, $4N+2$ spectral coefficients are stored per particle, where N is usually 3.

Putting all of this together, for a domain of length L, height H, grid spacing Δx, and P particles, the memory usage with double precision floating point (64 bit numbers) will be,

$$\text{RAM} \approx (64\text{bits})\left(16\underbrace{\frac{L\cdot H - \frac{1}{4}\pi D^2 P}{\Delta x^2}}_{\#\text{ of finite difference nodes}} + P\left(6\underbrace{\frac{\pi D}{\Delta x}}_{\#\text{ cage points}} + \underbrace{4N+2}_{\#\text{ of spectral coefficients}}\right)\right). \tag{35}$$

As a practical matter, several additional values are stored at each gridpoint in our implementation, purely as a convenience in post-processing, so the actual memory usage is about three times the minimum requirement posed by (35).

4. Flow over a stationary circular cylinder between sliding walls

To validate the scheme, we first simulated the flow over a stationary circular cylinder in a channel (Fig. 2) located midway between two moving side walls. The channel had a uniform inlet velocity profile, and the outflow boundary condition (10). This problem is mathematically equivalent to the problem of a particle moving in a fluid at constant speed down a channel with stationary walls, although the numerical treatment of these two cases differ slightly.

The flow was steady at $Re = 20$, and a vortex street formed at $Re = 100$. Fig. 3a and b show the convergence of the drag at Reynolds 20 and the convergence of the drag and maximum lift coefficients at $Re = 100$. In the latter case, an average drag

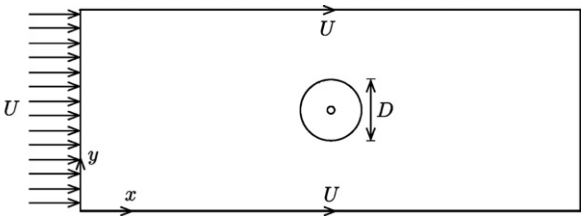

Fig. 2. A stationary circular particle between sliding walls.

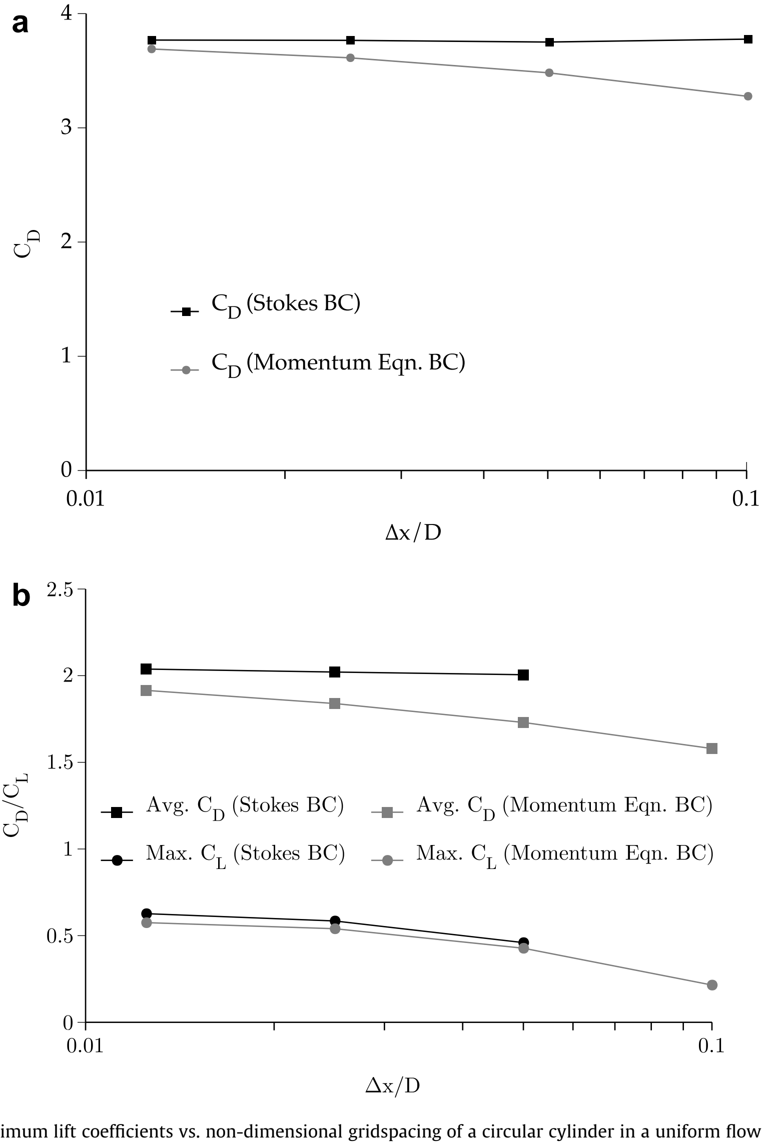

Fig. 3. Average drag and maximum lift coefficients vs. non-dimensional gridspacing of a circular cylinder in a uniform flow between sliding plates for (a) $Re = 20$, and (b) $Re = 100$, with $M = 0.05$, and plate separation of $4D$ in both figures. The present Stokes boundary condition converged faster than a boundary condition based on the momentum equation from [17].

coefficient, the arithmetic mean over two vortex-shedding periods, was used. Results for two different particle boundary conditions are shown in the figure; the first is the Stokes flow boundary condition described in the previous section, and the second is a condition derived from the momentum equation described in [17]. The bulk flow was solved by the same Lax–Wendroff scheme in both cases, and all other test parameters were identical. For a particle diameter of D, a Mach number $M = U/c$ of 0.05 was chosen, with a channel width of $4D$, channel length of $35D$, and the particle center in the middle of the channel and 15.5D from the inlet. For the Stokes flow boundary condition, the number of modes N in (29) and (30) was taken to be 3, 4, 5 and 8 for $D/\Delta x$ of 10, 20, 40, and 80 respectively. The time step was changed according to (25) as Δx was reduced. The memory requirements (from (35)) for $D/\Delta x$ of 10, 20, 40, and 80 were 1.7 MB, 6.8 MB, 27 MB, and 109 MB respectively. The scheme is typically run at $D/\Delta x$ of 15.

In all cases, the present Stokes flow boundary condition converged faster than the one based on the momentum equation from [17]. For the $Re = 20$ case, 10 grid spacings per particle diameter was sufficiently converged for the Stokes condition, and at $Re = 100$, 20 grid spacings per diameter gave an average drag coefficient within 1.6% of the converged value at the finest resolution, a maximum lift coefficient that was still 27% off of the converged value. The slow convergence of the lift coefficient could be caused by the linearization of flow near the particle, because the lift is a nonlinear effect. Refining the mesh brings the cage points closer to the particle surface, reducing the linearized region and eventually recovering the correct maximum lift coefficient, as shown by Fig. 3b. The converged lift coefficient was found to be 0.626 for the present scheme, and 0.593 for the Particle-Mover FEM.

In Fig. 4, the drag coefficient is plotted against the number of modes N in Eqs. (29) and (30). The resolution was fixed at 20 grid spacings per diameter. Since the number of equations must always be more than or equal to the number of unknown coefficients, this imposes a restriction on the number of modes that can be fitted, $N \leqslant (P-1)/2$, where P is the number of points on the cage. For the present case, there were 57 cage points, so N had to be smaller than 28. In practice, using too many terms can introduce aliasing errors, as observed by [22], before the $N = 28$ bound is reached. The curve in Fig. 4 indicates that the series converges rapidly for a very small N, becomes flat, and eventually starts to diverge again due to the aliasing errors when N is greater than 8. For an N of 10, the error is significant enough that the simulation becomes unstable. If N is chosen to be between a quarter and half the number of grid spacings per diameter, the series is found to converge, in line with [22].

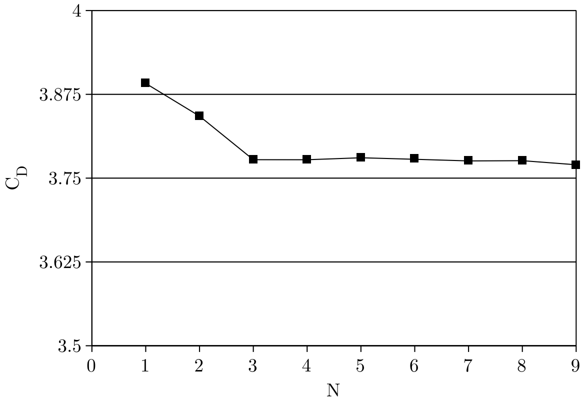

Fig. 4. Drag coefficient C_D vs. number of modes N of a circular cylinder in a uniform flow between sliding plates. $Re = 20$, $M = 0.05$, plate separation = 4D, and there are 20 grid spacings across the circle's diameter. An N of 3 is usually sufficient.

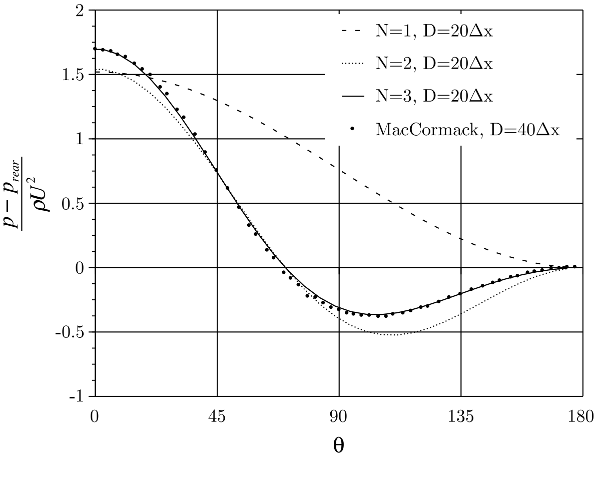

Fig. 5. Pressure on the particle surface at $Re = 20$ for three different modes, and a comparable solution on a fine mesh from the MacCormack scheme. Relative to the MacCormack scheme, there are large computational savings. θ is measured in degrees from the front of the cylinder for this graph.

One can get a feel for how quickly the spectral expansion converges by considering the pressure on the particle surface at $Re = 20$, shown in Fig. 5. The first three modes are plotted, along with results from our previous MacCormack scheme on a very fine mesh of $40\Delta x$ per particle diameter. The same results can be obtained at half that resolution using 3 modes (14 coefficients) of the Stokes flow solution. The Mach number was 0.05 for the Lax–Wendroff and MacCormack tests.

To gauge the accuracy of the method, we calculated drag coefficient as a function of Reynolds number for Re between 1 and 100 with all the boundaries far away, and compared to Sucker and Brauer [21], Takami and Keller [23], Dennis and Chang [3], and Hu and Joseph [10]. The results are in Table 1 and Fig. 6. For this test, two resolutions were considered, one coarse, with 10 grid spacings per diameter, and one fine, with 20 grid spacings per diameter, with $N = 3$ in each instance. Mach number was 0.05. The sliding side walls of the domain were 27.5 diameters from the center of the cylinder for all Reynolds numbers, and the particle was 20.5 diameters from the inlet and 34.5 diameters from the outlet. The cage was one grid spacing from the particle surface. The memory required for the 10 grid spacing per diameter case was 37 MB, and 147 MB for $D/\Delta x$ of 20 using Eq. (35).

The drag coefficients were in good agreement with the reference results for all Reynolds numbers between 1 and 60 for the coarse mesh. Beyond $Re = 60$ it was necessary to use a finer mesh to get accurate results, and on the coarse mesh the simulation actually became unstable beyond $Re = 80$. For Reynolds numbers below 40, the coarse mesh gave converged values, so we didn't compute the corresponding values on the fine mesh, except at $Re = 7$ and $Re = 20$ as a check. At $Re = 1$, the reference data ranged between $C_D = 10.28$ from Takami and Keller to $C_D = 11.13$ from Hu and Joseph, with $C_D = 11.30$ for the Lax–Wendroff scheme. Wall effects were the obvious culprit, and indeed we found that by moving the particle from 20.5

Table 1
$C_D = 2F_x/\rho U^2 D$ versus Re for a stationary cylinder far from any boundaries

Re	$\Delta x = D/10$, $N = 3$	$\Delta x = D/20$ $N = 3$	Sucker and Brauer	Takami and Keller	Dennis and Chang	Hu and Joseph
1	11.30	–	10.57	10.28	–	11.13
2	7.052	–	–	6.637	–	6.953
4	4.622	–	4.502	4.437	–	4.583
5	4.101	–	–	–	4.116	–
6	3.701	–	3.651	3.565	–	–
7	3.417	3.437	3.365	3.291	3.421	3.385
8	3.185	–	3.127	–	–	–
10	2.842	–	2.831	2.754	2.846	2.831
15	2.333	–	2.456	2.266	–	2.340
20	2.046	2.054	2.178	2.003	2.045	2.061
30	1.724	–	1.825	1.717	–	1.743
40	1.547	1.498	1.633	1.536	1.522	1.560
50	1.439	1.363	1.520	1.418	–	–
60	1.376	1.264	1.435	1.325	–	–
70	1.354	1.192	1.371	–	1.212	–
80	1.372	1.118	1.321	–	–	–
90	unstable	1.088	–	–	–	–
100	unstable	1.046	1.243	–	1.056	–

The results are plotted in Fig. 6. The references values are from Sucker and Brauer [21], Takami and Keller [23], Dennis and Chang [3], and Hu and Joseph [10].

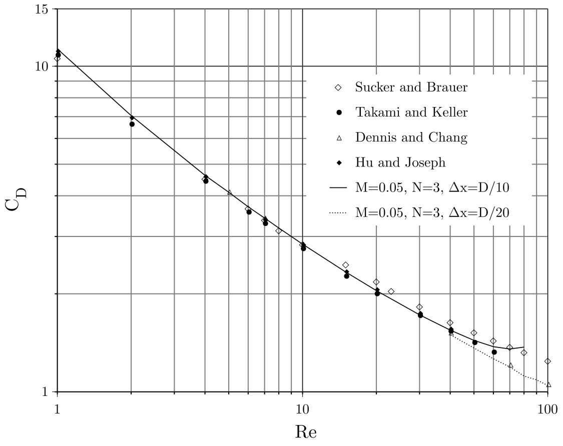

Fig. 6. $C_D = 2F_x/\rho U^2 D$ versus Re for a stationary cylinder far from any boundaries. The numerical values are given in Table 1.

diameters from the inlet to 27.5 diameters, the drag coefficient decreased by one percent. The same issue may be responsible for the scatter in the reference data.

To push the scheme as far as possible, a case at $Re = 300, M = 0.05$ was computed, with the same setup as the earlier tests (Fig. 2). However, a very fine mesh of $100\Delta x$ across the particle's diameter was required at this Reynolds number to obtain a stable solution.

5. Dragged cylinder

To verify that the scheme preserves Galilean invariance, one can take the stationary cylinder with sliding walls from the last section (Fig. 2) and instead hold the walls stationary and move the particle. Fig. 7 shows the lift and drag coefficients as a function of time for the moving particle as solid lines, with the corresponding values for the stationary particle plotted as dots. The forces agree up to small fluctuations. There are two types of fluctuations. The first type is a high frequency chatter from moving the particle over the fixed grid, common to most fixed grid methods. The other type is specific to compressible schemes and comes from sound waves generated by the particle reflecting off the walls and returning to hit the particles.

The high frequency chatter from the particle movement can be minimized by a couple of implementation techniques. The analytical expression for the velocity, derived from the streamfunction in (26), can be used to write velocities to grid points inside the particle as well as inside the cage. This initializes them to reasonable values so that they don't cause problems when the particles move, exposing these points. In addition to that, we found that allowing the cage to ‘‘crawl'' over the grid by choosing a new set of cage points at each time step as the particle moves (see Fig. 8a) is preferable to picking a single rigid cage and moving it in hops. Fig. 8b compares the noise in the drag coefficient for a crawling cage (Xes) to a cage that hops one grid spacing as the particle center passes the half grid spacing points (squares) at $Re = 20$. (The crawling cage data is from Fig. 7.) The noise amplitude for the crawling cage is four times smaller than for a hopping cage. The algorithm for the crawling cage was identical to the one used to draw the particle in [17], and further details may be found in that paper. In both the MacCormack scheme and in this scheme, the fluctuations decreased as Mach number increased. Mach number of 0.1 was the best compromise between maintaining nearly incompressible flow and reducing the chatter.

Sound waves generated by particle motion also cause force fluctuations when they reflect off the walls and return to the particles. The waves decay naturally because of the fluid viscosity and numerical dissipation, so no special treatment is required. However, if elimination of the waves is desired, non-reflecting wall boundary conditions can be constructed, for example Poinsot and Lele [18].

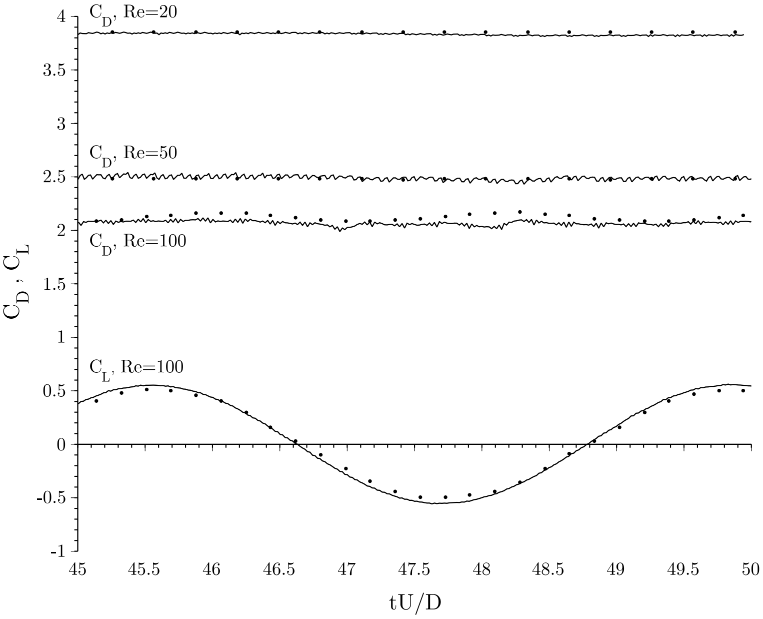

Fig. 7. Lift and drag coefficients for a cylinder translating at constant speed through a 4*D* wide channel at $Re = 20$, $Re = 50$, and $Re = 100$. Solid lines indicate the moving cylinder, and circles correspond to the stationary cylinder.

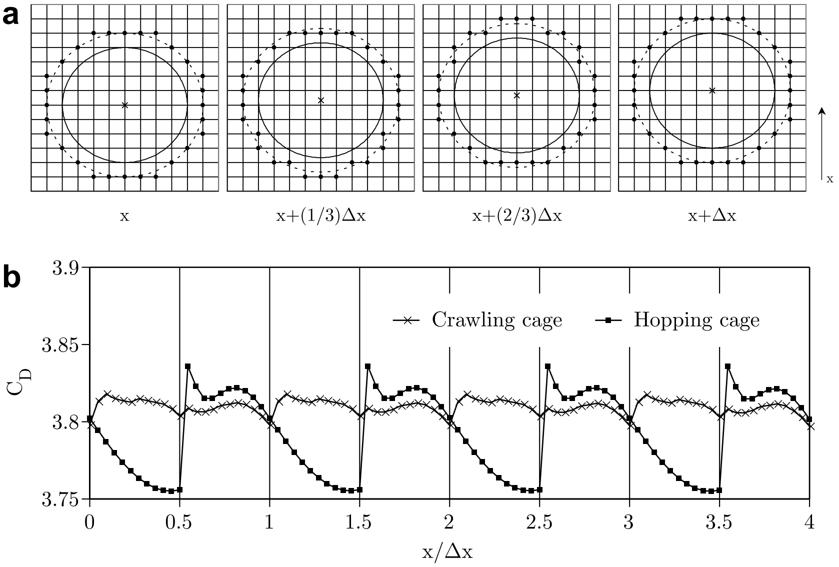

Fig. 8. (a) A particle with a "crawling" cage moves upward by one grid spacing. The inner circle is the particle surface, and the outer circle is displaced from the particle surface by one grid spacing. The cage is constructed around the outer circle. (b) The fluctuations in the drag coefficient at $Re = 20$, $M = 0.1$ are plotted for a cage that crawls as the particle moves, and for a cage that hops rigidly as the particle center passes the half grid spacing points.

6. Freely falling particles

Next we simulate the motion of free cylindrical particles by integrating the equations of motion for the particles and moving them accordingly.

The equations of motion are the result of a force and moment balance (per unit length) on each particle. They are given as

$$\begin{aligned} m\frac{\mathrm{d}\mathbf{w}}{\mathrm{d}t} &= \int_S \hat{\mathbf{n}}\cdot\sigma dS + (m - m_{\text{fluid}})\mathbf{g} \\ &= \left(\frac{4\pi\mu}{a}\widetilde{B}_1 + \rho_0\pi a^2\dot{w}_x\right)\hat{\imath} + \left(-\frac{4\pi\mu}{a}B_1 + \rho_0\pi a^2\dot{w}_y\right)\hat{\ } + (m - m_{\text{fluid}})\mathbf{g} \\ I\frac{\mathrm{d}\Omega}{\mathrm{d}t} &= \mathbf{k}\cdot\int_S \mathbf{r}\times(\hat{\mathbf{n}}\cdot\sigma)dS = -8\pi\mu A_0 + \rho_0\pi a^4\dot{\Omega} \end{aligned} \tag{36}$$

where m and I are the particle mass and moment of inertia, $m_{\text{fluid}} = \rho_0(\pi D^2/4)$ is the mass of fluid displaced by the particle, σ is the total stress tensor, $\mathbf{g}$ is the gravitational acceleration, $\mathbf{w}$ is the velocity of the particle center, Ω is the particle's angular velocity, and $\hat{\imath}$, $\hat{\jmath}$ are x- and y- unit vectors. The particle position is updated using with

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \Delta t\mathbf{w}^n + \frac{1}{2}\Delta t^2\frac{\mathbf{F}^n_{\text{net}}}{m} \tag{37}$$

where $\mathbf{F}_{\text{net}}$ is the net force and $\mathbf{x}$ is the center of the particle.

A term corresponding to the weight of fluid displaced by the particle—the buoyancy—is included in the particle equations of motion (36). Because the flow is nearly incompressible, the body force terms in the fluid momentum equations can be neglected provided gL/c^2 is much smaller than unity, where L is the characteristic length of the domain and c is the speed of sound. This number will be exactly zero for an incompressible fluid, and it measures the effect of gravity on small density fluctuations, such as sound waves. In real fluids this is well satisfied under most circumstances.

Here is a summary of the algorithm.

(1) Choose Δx, Δt, N, and initialize the flow fields and the particle states.
(2) March in time:
 (a) Choose cage points for all the particles.
 (b) Compute F, G, H, P, Q, R, S, and T at each point in the bulk fluid and on walls. These should be computed everywhere, including inside the particles' cages.
 (c) Compute $(\rho u)^{n+1}$, $(\rho v)^{n+1}$, and ρ^{n+1} from Eqs. (8)–(10) in the bulk fluid and on the walls. Omit points inside the particles' cages.
 (d) Calculate the pressure and vorticity on the cage points in the particle's reference frame, making use of Eqs. (32) and (33).
 (e) Solve for the spectral coefficients by least squares.

(f) Update the velocities inside the cage and inside the particles using

$$v_r = \frac{a^{-1}}{s}\frac{\partial\psi}{\partial\theta}, \quad v_\theta = -a^{-1}\frac{\partial\psi}{\partial s} \quad \text{with } \psi \text{ given by (26).}$$

(g) Move the particles according to (36). Check for collisions and add collision forces as necessary.

In applying the change of variables, Eqs. (33) and (34), there are two possibilities. One can evaluate $\mathbf{w}$, $\dot{\mathbf{w}}$, $\mathbf{\Omega}$, and $\dot{\mathbf{\Omega}}$ at time step n, in which case the procedure is to transform to particle coordinates, do the least squares fit to determine the spectral coefficients, and then transform back to obtain the velocities and pressure inside the cage as described above. An improved procedure is to approximate $\mathbf{w}$ and $\mathbf{\Omega}$ at time $n+1$ using e.g. $\mathbf{w}^{n+1} = \mathbf{w}^n + \Delta t\dot{\mathbf{w}}^{n+1}$, and simultaneously least squares fit $\dot{\mathbf{w}}^{n+1}$, $\dot{\mathbf{\Omega}}^{n+1}$, and the equations of motion of the particle at t_{n+1}. This leaves the finite difference scheme explicit, but makes the particle equations of motion implicit. The advantage is that when the density ratio of particle to fluid is close to one, the particle equations of motion become unconditionally stable (up to the stability limit of the outer finite difference scheme). Otherwise, it is possible for the particles' equations of motion to be more unstable than the Lax–Wendroff method and limit the overall stability. This was not an issue for the two dimensional results shown in this paper, but in ongoing work on extending the scheme to three dimensions, it was found to improve the stability for the neutrally buoyant particle cases. The numerical output of the scheme is not significantly affected by this modification.

Fig. 9 shows particle acceleration, velocity, and displacement in the vertical direction for a particle dropped in the center of a long closed channel, 70D long and 4D wide, as a function of dimensionless time. These quantities are compared to prior results from the MacCormack scheme. The converged MacCormack result was obtained on a fine mesh of 40Δx per diameter, the converged Lax–Wendroff resolution was 15Δx per diameter, and for reference the 15Δx MacCormack result is also shown. For this test, the density ratio of solid to fluid was 10.55, which corresponds to a terminal Reynolds number of 20 and a terminal Mach number of 0.1 in both schemes. The results are non-dimensionalized with U, the terminal speed for the converged MacCormack particle, and the particle diameter D. The memory estimate is 7.7 MB by Eq. (35). The test demonstrates that in this case, nearly identical results can be obtained on a much coarser grid (and correspondingly larger time step) with the Stokes flow boundary conditions. In general, the smoothness of the acceleration deteriorates somewhat as the density ratio approaches one, but the the fluctuations remain within 10% of the magnitude of the drag force at $D = 15\Delta x$ for the Lax–Wendroff scheme.

As a second demonstration of the scheme's accuracy, one can examine particle trajectories for the case of a particle dropped off-center in a long channel with terminal Reynolds and Mach numbers of 8.33 and 0.1 respectively. In Fig. 10, trajectories from Feng et al. [4] and Zhang and Prosperetti [25] are plotted beside results from the current work. The trajectory was found to be closer to Feng's result. The density ratio of solid to fluid was $\rho_s/\rho_0 = 1.03$. The channel was 40D long, and 4D wide. The particle center was initially 2D from the top of the channel, and at $y/H = 0.19$. The grid was chosen to yield 15 grid spacings (or 16 grid points) across the particle diameter, and the total size was 641×65 grid points. The memory estimate was 4.4 MB. The walls were stationary. Four modes were used ($N = 4$), to facilitate direct comparison with [25] although prior tests indicated that three modes would have been sufficient. The time step was $4\nu\Delta t/D^2 = 0.001$ for this test.

For the case of multiple moving particles, a collision scheme is necessary because under most circumstances the lubrication forces will only become large enough to prevent collisions in a time that is much smaller than Δt. We use the same collision scheme described in [17], which guarantees that the particles will come no closer than a certain gap δ.

Fig. 11 illustrates the "drafting, kissing, and tumbling" process observed when two particles positioned on the centerline of a long channel are released. In this configuration, the top particle is pulled in by the wake of the bottom particle ("drafting") until the two touch and fall together for some time ("kissing") then because the configuration is unstable, the particles turn sideways and separate ("tumbling"). To increase the effective length of the channel, a uniform inflow is introduced with walls that slide upward at the same rate as the inflow. (This is the same as the Galilean transformation demonstrated earlier for one particle.)

The channel is 8 cm by 2 cm, the particle diameter is 0.25 cm, μ is 0.01 Pa-s, the fluid density is 1 g/cm^3, the particle density is 1.01 g/cm^3, g is 981 cm/s^2, the speed of sound is 13.8 cm/s, the security zone around the particle for the collision scheme is $\delta = 1\Delta x$, the cage is also at roughly one grid spacing from the particle surface, there are 20 grid spacings per diameter, $N = 5$, and the updraft and wall velocity are 1.15 cm/s. The simulation needed 780 KB of RAM. The center of the bottom particle was initially 4.5 cm from the channel inlet, and half way between the two walls. The center of the top particle was 5 cm from the inlet and offset from the centerline by $\Delta x/40$, to encourage the particles to tumble. If the top particle is not offset, it will take a very long time for sufficient numerical error to build up and trigger the tumbling instability.

Particle paths for this problem are shown in Fig. 12 superimposed on top of the same problem calculated using the finite element scheme Particle-Mover from [8]. The displacement due to the updraft has been subtracted off to allow comparison to the Particle-Mover result, which was computed in an infinitely long channel. The location where the particle hits the right wall was matched to the Particle-Mover solution by adjusting the initial offset of the top particle. Aside from this adjustment, the particle tracks agree very well with the finite element code in most other locations.

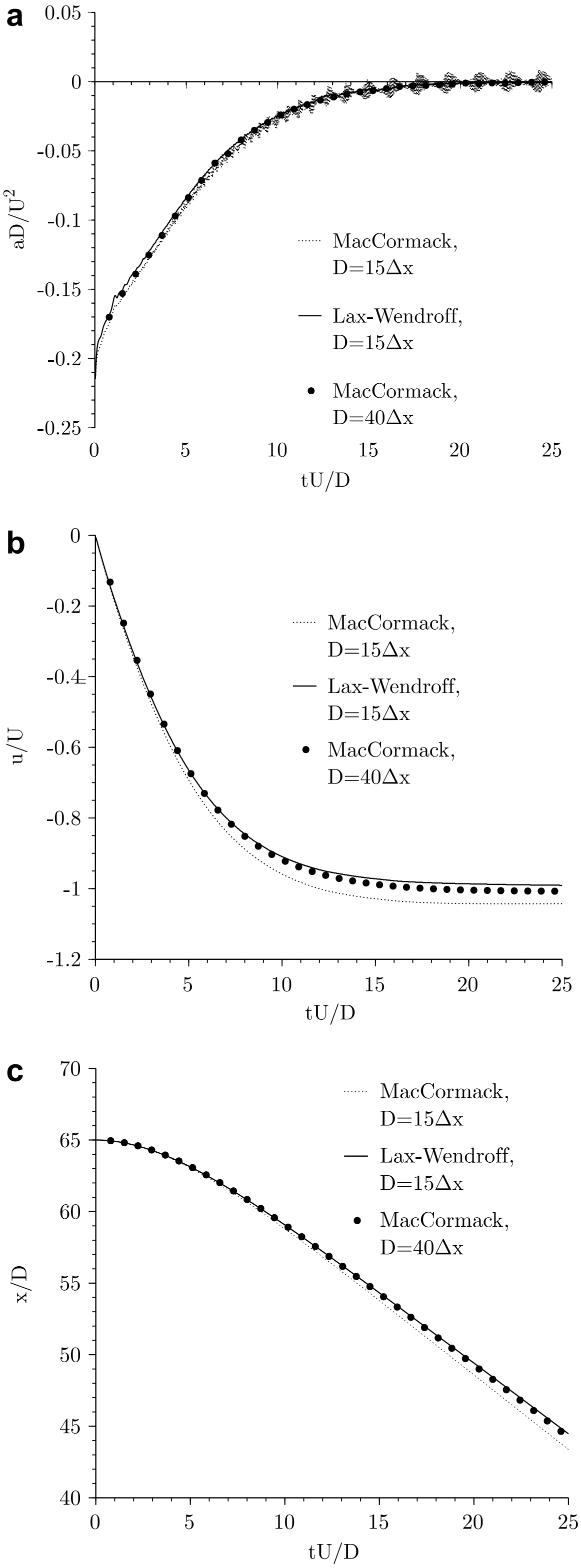

Fig. 9. A single particle falling in a channel. The terminal Reynolds number is 20, and the channel is 4D wide. The Lax–Wendroff $D = 15\Delta x$ result is converged.

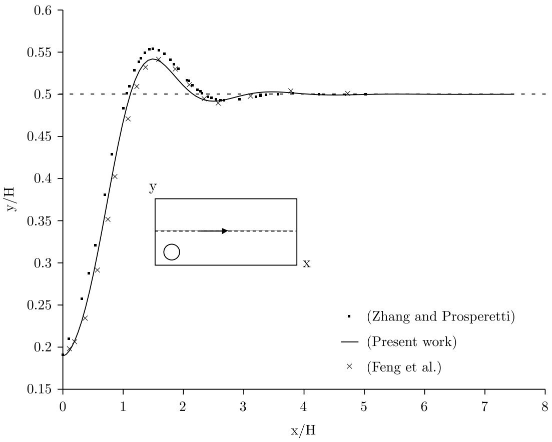

Fig. 10. These are the trajectories for a particle dropped off-center in a long channel, with terminal *Re* of 8.33 and terminal *M* of 0.1. The particle falls from left to right. The dots are from Zhang and Prosperetti in [25], the crosses are from Feng et al. [4], and the solid line is from the present scheme. *H* is the channel width.

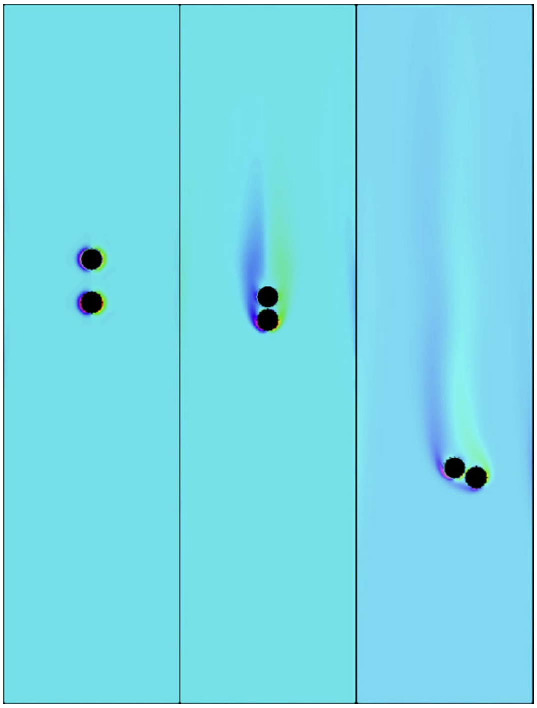

Fig. 11. Two particles draft (left image), kiss (center), and tumble (right). There is a 1.15 cm/s updraft from the bottom of the channel. Coloring indicates vorticity.

Fig. 13 shows 248 particles sedimenting in a closed rectangular box. The particles are initially in a staggered arrangement. When the particles drop, the center falls fastest, entraining fluid and causing updrafts near the walls. Two counter-rotating eddies form, dragging particles around for a while until viscosity dissipates the initial potential energy in the system and the particles settle to the bottom of the box. Similar eddies form in Glowinsky et al.'s simulation [5], although the direction of the eddies' rotation is opposite the current case, because the particles' initial condition is not staggered in Glowinsky.

The collision scheme calculates the exact force necessary to prevent the particles from violating the safety zone. (It does not use an ansatz for the lubrication force.) At the bottom of the box, a particle's net force is completely cancelled, so the acceleration and velocity terms in Eqs. (36) and (37) are almost exactly zero, and the particles just sit there.

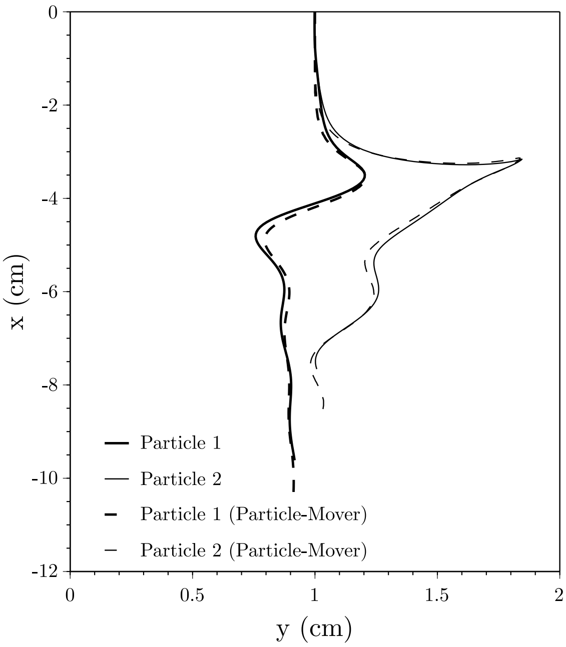

Fig. 12. Particle tracks for the drafting, kissing, and tumbling of two particles. The initial condition of the top particle has been adjusted to make the location where the particle bounces off the right wall the same as for the Particle-Mover finite element result.

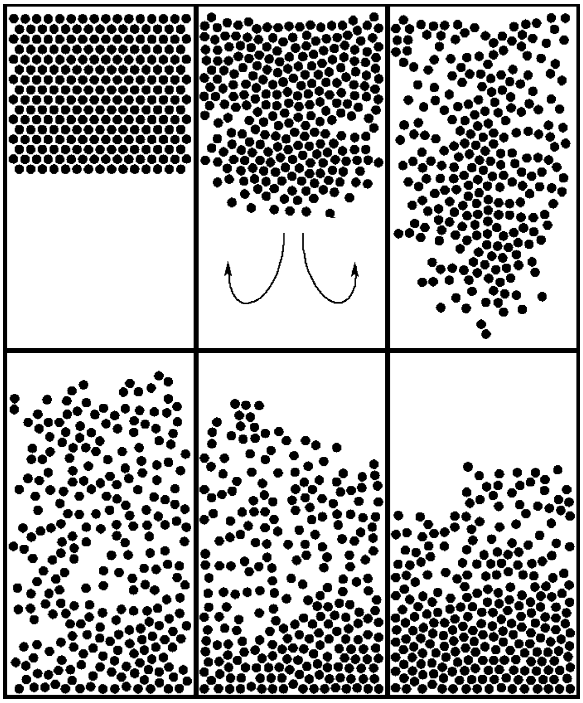

Fig. 13. 248 particles sedimenting in a closed box. Two counter-rotating eddies form, with flow direction indicated in the second frame.

The particle diameters were 1.875 mm, the dynamic viscosity was 0.01 Pa-s, the fluid density was 1 g/cm^3, the particle density was 1.01 g/cm^3, g was 981 cm/s^2, the speed of sound was 11 cm/s, and the box was 6.6625 cm high and 3.6375 cm wide. There were 15 grid spacings across each particle diameter, and the minimum allowed particle gap was 10% of the radius. The single particle terminal Reynolds and Mach numbers were 20.86 and 0.101 respectively, the density ratio was 1.01, gD/U^2 was 148.75, and the box was 35.53 by 19.4 diameters.

The memory estimate for this problem was 14 MB. The actual memory usage was 45 MB, since several extra quantities are stored at each node for use in post-processing, purely as a matter of convenience.

7. Concluding remarks

An explicit Lax–Wendroff scheme for simulating particulate flows has been developed that allows for accurate computation of the particle forces on coarse grids. Because the simulation can run on a coarse grid, one can take large enough time steps to make the explicit scheme attractive. Explicit schemes on regular grids use much less memory than their implicit cousins (as much as two orders of magnitude less) and produce compact code that can be rapidly modified to include additional forces. The memory advantage for the explicit scheme also becomes more obvious in 3D.

Acknowledgments

A. Perrin acknowledges the support from the Graduate Assistance in Areas of National Need (GAANN) fellowship from the US Department of Education. The work is also partially supported by the Nano/Bio Interface Center through the NSF NSEC DMR-045780.

References

[1] C.K. Aidun, Y. Lu, E.-J. Ding, Direct analysis of particulate suspensions with inertia using the discrete boltzmann equation, J. Fluid Mech. 373 (1998) 287–311.

[2] O. Behrend, Solid-fluid boundaries in particle suspension simulations via the lattice boltzmann method, Phys. Rev. E 52 (1) (1995) 1164–1175.

[3] S.C.R. Dennis, G.Z. Chang, Numerical solutions for steady flow past a circular cylinder at reynolds numbers up to 100,J. Fluid Mech. Digital Arch. 42 (03) (2006) 471–489.

[4] J. Feng, H.H. Hu, 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 (February) (1994) 95–134.

[5] R. Glowinski, T.W. Pan, T.I. Helsa, D.D. Joseph, J. Périaux, A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow, J. Comput. Phys. 169 (2) (2001) 363–426.

[6] R. Glowinski, T.-W. Pan, T.I. Hesla, D.D. Joseph, Distributed lagrange multiplier/fictitious domain method for particulate flows, Intl. J. Multiphase Flow 25 (5) (1999) 755–794.

[7] J. Happel, H. Brenner, Low Reynolds Number Hydrodynamics: With Special Applications to Particulate Media, Springer, 1981.

[8] H.H. Hu, Direct simulation of flows of solid-liquid mixtures, Intl. J. Multiphase Flow 22 (2) (1996) 335–352.

[9] H.H. Hu, D.D. Joseph, M.J. Crochet, Direct simulation of fluid particle motions, Theoret. Comput. Fluid 3 (1992) 285–306.

[10] H.H. Hu, D.D. Joseph, Numerical simulation of viscoelastic flow past a cylinder, J. Non-Newtonian Fluid Mech. 37 (1990) 347–377.

[11] H.H. Hu, N.A. Patankar, M.Y. Zhu, Direct numerical simulations of fluid-solid systems using the arbitrary Lagrangian–Eulerian technique, J. Comput. Phys. 169 (2001) 427–462.

[12] A.A. Johnson, T.E. Tezduyar, Simulation of multiple spheres falling in a liquid-filled tube, Comput. Methods Appl. Mech. Engrg. 134 (1996) 351–373.

[13] P.K. Kundu, I.M. Cohen, Fluid Mechanics, third ed., Elsevier Academic Press, 2004. (pp. 686–687).

[14] A.J.C. Ladd, Short-time motion of colloidal particles: numerical simulation via a fluctuating Lattice-Boltzmann equation, Phys. Rev. Lett. 70 (9) (1993) 1339–1342.

[15] P.D. Lax, B. Wendroff, Systems of conservation laws, Commun. Pure. Appl. Math. 13 (1960) 217–237.

[16] R.R. Nourgaliev, T.N. Dinh, T.G. Theofanous, D. Joseph, The Lattice Boltzmann equation method: theoretical interpretation, numerics and implications, Intl. J. Multiphase Flow 29 (2003) 117–169.

[17] A. Perrin, H.H. Hu, An explicit finite-difference scheme for simulation of moving particles, J. Comput. Phys. 212 (2006) 166–187.

[18] T.J. Poinsot, S.K. Lele, Boundary conditions for direct simulations of compressible viscous flows, J. Comput. Phys. 101 (1992) 104–129.

[19] A. Prosperetti, H.N. Oguz, Physalis: a new o(n) method for the numerical simulation of disperse systems: potential flow of spheres, J. Comput. Phys. 167 (2001) 196–216.

[20] D. Qi, Lattice–Boltzmann simulations of particles in non-zero-reynolds-number flows, J. Fluid Mech. 385 (1999) 41–62.

[21] D. Sucker, H. Brauer, Fluiddynamik bei quer angeströmten zylindern, Heat Mass Transfer 8 (3) (1975) 149–158.

[22] S. Takagi, H.N. Oguz, Z. Zhang, A. Prosperetti, Physalis: a new method for particle simulation part ii: two-dimensional Navier–Stokes flow around cylinders, J. Comput. Phys. 187 (2003) 371–390.

[23] H. Takami, H.B. Keller, Steady two-dimensional viscous flow of an incompressible fluid past a circular cylinder, Physics of Fluids 12 (12) (1969) II-51–II-56.

[24] J.C. Tannehill, D.A. Anderson, R.H. Pletcher, Computational Fluid Mechanics and Heat Transfer, second ed., Taylor & Francis, Washington, DC, 1997.

[25] Z. Zhang, A. Prosperetti, A method for particle simulation, Trans. ASME 70 (2003) 64–74.

[26] Z. Zhang, A. Prosperetti, A second-order method for three-dimensional particle simulation, J. Comput. Phys. 210 (2005) 292–324.