An Explicit Finite-Difference Scheme for Simulation of Moving Particles

A. Perrin, H. H. Hu

Department of Mechanical Engineering and Applied Mechanics
University of Pennyslvania
297 Towne Building, 220 S. 33rd Street, Philadelphia, PA 19104

Abstract

We present an explicit finite-difference scheme for direct simulation of the motion of solid particles in a fluid. The method is based on a second order MacCormack finite-difference solver for the flow, and Newton's equations for the particles. The fluid is modeled with fully compressible mass and momentum balances; the technique is intended to be used at moderate particle Reynolds number. Several examples are shown, including a single stationary circular particle in a uniform flow between two moving walls, a particle dropped in a stationary fluid at particle Reynolds number of 20, the drafting, kissing, and tumbling of two particles, and 100 particles falling in a closed box.

1 Introduction

Direct numerical simulation of particulate flows remains a challenging problem. Finite element methods, such as the Particle-Mover, which is based on the arbitrary Lagrangian-Eulerian (ALE) method of Hu [9],[8],[10], or Johnson and Tezduyar's stabilized space-time method [11], are efficacious for small numbers of particles at moderate Re but have prohibitive computational requirements when particles fill the domain. The ALE method, for example, uses a moving unstructured grid with nodes on the particle surface that follow the particle. The movement of the interior nodes are computed from an elliptic equation, and at each time step the grid is updated according to the motion of the particles. When unacceptable grid distortion is detected, a new grid is generated and the flow fields are projected from the old grid to the new

Email addresses: aperrin@seas.upenn.edu (A. Perrin,), hhu@seas.upenn.edu (H. H. Hu).

one. When these methods are used to simulate dense particle suspensions, the mesh may need to be refined excessively. This gives rise to memory and/or processing issues due to the remeshing and projection procedures. A fixed, uniform grid becomes an attractive choice.

In solving the incompressible Navier-Stokes equations using the primitive variables (velocity and pressure), one numerical difficulty lies in the continuity equation. The continuity equation can be regarded either as a constraint on the flow field to determine the pressure or the pressure plays the role of the Lagrange multiplier to satisfy the continuity equation. In a flow field, the information (or disturbance) travels with both the flow and the speed of sound in the fluid. Since the speed of sound is infinite in an incompressible fluid, pressure disturbances are propagated instantaneously throughout the domain. In many numerical schemes for solving the incompressible Navier-Stokes equations, the pressure is obtained by solving a Poisson equation. The Poisson equation may occur in either continuous form or discrete form. Solving the pressure Poisson equation is often the most costly step in these schemes. One technique to surmount the difficulty of the incompressible limit is to introduce an artificial compressibility (AC) as Chorin did [4]. This formulation is normally used for steady problems with a pseudo-transient formulation. In the formulation, the continuity equation is replaced by,

$$\frac{\partial p}{\partial t} + c^2 \nabla \cdot \mathbf{u} = 0 \tag{1}$$

where c is an arbitrary constant and could be the speed of sound in a corresponding compressible fluid with the equation of state $p = c^2\rho$. The formulation is called pseudo-transient because (1) does not have any physical meaning before the steady state is reached. However, when c is large, (1) can be considered as an approximation to the unsteady solution of the incompressible Navier-Stokes problem[4]. Nourgaliev et al. [16] have pointed out that the AC method is both easily parallelized and economically coded, giving it some of the advantages of the lattice Boltzmann equation. Briefly, the idea of the lattice Boltzmann method (LBM) is to devise a simplified mesoscopic kinetic equation called the lattice Boltzmann equation from which the usual macroscopic quantities can be extracted. The main advantages of the LBM are ease of implementation, because it is an explicit scheme, and consequently, ease of parallelization. The LBM has been well established for the simulation of particulate flows by, among others, Ladd [13], Behrend [2], Aidun et al. [1], and Qi [18]. Chen and Doolen [3] provide a good review of the LBM and devote a section to particle problems. More recently, Nourgaliev et al. have written a tutorial in the method that describes its application to general multiphase flows [17].

This paper develops an explicit finite difference scheme for direct numerical solution of particles in a nearly incompressible Newtonian fluid. The present

work is closest in spirit to Norgaliev's Numerical Acoustic Relaxation (NAR) method [16]. Our scheme is fully explicit and second order in both time and space. Rather than use equation (1), we instead use the fully compressible continuity and momentum equations, but with an isothermal equation of state. The force and torque are calculated by integration over the interface, and the particles are then explicitly moved according to Newton's third law. An explicit scheme has an advantage for simulation of dense suspensions because the intrinsic time scale to resolve the particle motion is small, so the stability condition for such a scheme is not too restrictive.

This paper first discusses the details of our flow solver, a MacCormack scheme. It is validated using a driven cavity problem. We present several increasingly general example problems, starting with a flow over a stationary circular cylinder between two sliding walls, and followed by the formally (but not numerically) equivalent problem of the same cylinder translating at constant speed through an initially stationary fluid. The latter is then extended to allow the cylinder to fall freely.

2 Explicit MacCormack Scheme

Instead of using the artificial continuity equation of (1), one may start with the exact compressible Navier-Stokes equations with the equation of state. In Cartesian coordinates, the component form of the continuity equation and compressible Navier-Stokes equation in two dimensions can be written as

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} + \frac{\partial (\rho v)}{\partial y} = 0, \tag{2}$$

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

$$\frac{\partial}{\partial t}(\rho v) + \frac{\partial}{\partial x}(\rho u v) + \frac{\partial}{\partial y}\left(\rho v^2\right) = \rho g_y - \frac{\partial p}{\partial y} + \mu \nabla^2 v + \frac{\mu}{3}\frac{\partial}{\partial y}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) \tag{4}$$

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 [12]).

The explicit MacCormack scheme, after [14], is essentially a predictor-corrector scheme, similar to a second-order Runge-Kutta method commonly used to

solve ordinary differential equations. For a system of equations of the form,

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{E}\left(\mathbf{U}\right)}{\partial x} + \frac{\partial \mathbf{F}\left(\mathbf{U}\right)}{\partial y} = 0, \tag{5}$$

the explicit MacCormack scheme consists of two steps,

$$\mathbf{U}^{*}_{i,j} = \mathbf{U}^{n}_{i,j} - \frac{\Delta t}{\Delta x}\left(\mathbf{E}^{n}_{i+1,j} - \mathbf{E}^{n}_{i,j}\right) - \frac{\Delta t}{\Delta y}\left(\mathbf{F}^{n}_{i,j+1} - \mathbf{F}^{n}_{i,j}\right), \qquad \text{(Predictor)}$$

$$\mathbf{U}^{n+1}_{i,j} = \frac{1}{2}\left[\mathbf{U}^{n}_{i,j} + \mathbf{U}^{*}_{i,j} - \frac{\Delta t}{\Delta x}\left(\mathbf{E}^{*}_{i,j} - \mathbf{E}^{*}_{i-1,j}\right) - \frac{\Delta t}{\Delta y}\left(\mathbf{F}^{*}_{i,j} - \mathbf{F}^{*}_{i,j-1}\right)\right]. \qquad \text{(Corrector)}$$

The vector $\mathbf{U} = (\rho, \rho\mathbf{u}, \rho\mathbf{v})$ contains the update variables. The vectors $\mathbf{E}$, and $\mathbf{F}$ are functions of the update variables and some of their spacial derivatives. Notice that the spatial derivatives in (5) are discretized with opposite one-sided finite differences in the predictor and corrector stages. The star variables are supposed to be evaluated at time level t_{n+1}. This scheme is second-order accurate in both time and space.

Applying the MacCormack scheme to the compressible Navier-Stokes equations (2)-(4) and replacing the pressure with $p = c^2\rho$, we have the predictor step,

$$\rho^{*}_{i,j} = \rho^{n}_{i,j} - c_1\left[(\rho u)^{n}_{i+1,j} - (\rho u)^{n}_{i,j}\right] - c_2\left[(\rho v)^{n}_{i,j+1} - (\rho v)^{n}_{i,j}\right] \tag{6}$$

$$\begin{aligned}(\rho u)^{*}_{i,j} = {} & (\rho u)^{n}_{i,j} - c_1\left[\left(\rho u^2 + c^2\rho\right)^{n}_{i+1,j} - \left(\rho u^2 + c^2\rho\right)^{n}_{i,j}\right] \\ & - c_2\left[(\rho uv)^{n}_{i,j+1} - (\rho uv)^{n}_{i,j}\right] \\ & + \frac{4}{3}c_3\left(u^{n}_{i+1,j} - 2u^{n}_{i,j} + u^{n}_{i-1,j}\right) + c_4\left(u^{n}_{i,j+1} - 2u^{n}_{i,j} + u^{n}_{i,j-1}\right) \\ & + c_5\left(v^{n}_{i+1,j+1} + v^{n}_{i-1,j-1} - v^{n}_{i+1,j-1} - v^{n}_{i-1,j+1}\right)\end{aligned} \tag{7}$$

$$\begin{aligned}(\rho v)^{*}_{i,j} = {} & (\rho v)^{n}_{i,j} - c_1\left[(\rho uv)^{n}_{i+1,j} - (\rho uv)^{n}_{i,j}\right] \\ & - c_2\left[\left(\rho v^2 + c^2\rho\right)^{n}_{i,j+1} - \left(\rho v^2 + c^2\rho\right)^{n}_{i,j}\right] \\ & + c_3\left(v^{n}_{i+1,j} - 2v^{n}_{i,j} + v^{n}_{i-1,j}\right) + \frac{4}{3}c_4\left(v^{n}_{i,j+1} - 2v^{n}_{i,j} + v^{n}_{i,j-1}\right) \\ & + c_5\left(u^{n}_{i+1,j+1} + u^{n}_{i-1,j-1} - u^{n}_{i+1,j-1} - u^{n}_{i-1,j+1}\right)\end{aligned} \tag{8}$$

Similarly, the corrector step is given by,

$$2\rho_{i,j}^{n+1} = \rho_{i,j}^{n} + \rho_{i,j}^{*} - c_1\left[(\rho u)_{i,j}^{*} - (\rho u)_{i-1,j}^{*}\right] - c_2\left[(\rho v)_{i,j}^{*} - (\rho v)_{i,j-1}^{*}\right] \quad (9)$$

$$\begin{aligned} 2\,(\rho u)_{i,j}^{n+1} = {} & (\rho u)_{i,j}^{n} + (\rho u)_{i,j}^{*} - c_1\left[\left(\rho u^2 + c^2\rho\right)_{i,j}^{*} - \left(\rho u^2 + c^2\rho\right)_{i-1,j}^{*}\right] \\ & - c_2\left[(\rho uv)_{i,j}^{*} - (\rho uv)_{i,j-1}^{*}\right] \\ & + \frac{4}{3}c_3\left(u_{i+1,j}^{*} - 2u_{i,j}^{*} + u_{i-1,j}^{*}\right) + c_4\left(u_{i,j+1}^{*} - 2u_{i,j}^{*} + u_{i,j-1}^{*}\right) \\ & + c_5\left(v_{i+1,j+1}^{*} + v_{i-1,j-1}^{*} - v_{i+1,j-1}^{*} - v_{i-1,j+1}^{*}\right) \end{aligned} \quad (10)$$

$$\begin{aligned} 2\,(\rho v)_{i,j}^{n+1} = {} & (\rho v)_{i,j}^{n} + (\rho v)_{i,j}^{*} - c_1\left[(\rho uv)_{i,j}^{*} - (\rho uv)_{i-1,j}^{*}\right] \\ & - c_2\left[\left(\rho v^2 + c^2\rho\right)_{i,j}^{*} - \left(\rho v^2 + c^2\rho\right)_{i,j-1}^{*}\right] \\ & + c_3\left(v_{i+1,j}^{*} - 2v_{i,j}^{*} + v_{i-1,j}^{*}\right) + \frac{4}{3}c_4\left(v_{i,j+1}^{*} - 2v_{i,j}^{*} + v_{i,j-1}^{*}\right) \\ & + c_5\left(u_{i+1,j+1}^{*} + u_{i-1,j-1}^{*} - u_{i+1,j-1}^{*} - u_{i-1,j+1}^{*}\right) \end{aligned} \quad (11)$$

The coefficients are defined as

$$c_1 = \frac{\Delta t}{\Delta x},\ c_2 = \frac{\Delta t}{\Delta y},\ c_3 = \frac{\mu \Delta t}{(\Delta x)^2},\ c_4 = \frac{\mu \Delta t}{(\Delta y)^2},\ \text{and}\ c_5 = \frac{\mu \Delta t}{12 \Delta x \Delta y}. \quad (12)$$

In both the predictor and corrector steps the viscous terms (the second-order derivative terms) are discretized with centered-differences to maintain second-order accuracy. For brevity, body force terms in the momentum equations are neglected here.

During the predictor and corrector stages of the explicit MacCormack scheme (6)-(11), one-sided differences are arranged in the FF and BB fashion, respectively. In the notation FF, the first F denotes the forward difference in the x-direction and the second F denotes the forward difference in the y-direction. Similarly, BB stands for backward differences in both x and y directions. We denote this arrangement as FF/BB. Similary, one may get BB/FF, FB/BF, and BF/FB arrangements. Tannehill, Anderson, and Pletcher ([19]) suggest cycling these arrangements, and we note that some balanced cyclings generate better results than others for a particular problem. For particle problems, the sequence FF/BB to FB/BF to BB/FF to BF/FB was found to help avoid asymmetries in the truncation error of the flow variables, which can result in spurious forces on the particle.

Tannehill, Anderson and Pletcher ([19]) give the following semi-empirical stability criterion for the explicit MacCormack scheme,

$$\Delta t \leq \frac{\sigma}{(1 + 2/\mathrm{Re}_\Delta)}\left[\frac{|u|}{\Delta x} + \frac{|v|}{\Delta y} + c\sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}}\right]^{-1}, \quad (13)$$

where σ is a safety factor (≈ 0.9), $\mathrm{Re}_\Delta = \min\left(\rho\,|u|\,\Delta x/\mu, \rho\,|v|\,\Delta y/\mu\right)$ is the

minimum mesh Reynolds number. This condition is quite conservative for flows with small mesh Reynolds numbers, and so the MacCormack scheme will be inefficient in this regime (flow Re smaller than 10). We find that at moderate flow Reynolds numbers (Re = 10 to 500) and small Mach numbers the stability condition in (13) reduces to the CFL condition,

$$\Delta t \leq \frac{0.5\Delta x}{c}. \tag{14}$$

For particulate flows, the time step for the particle motion is roughly determined by the CFL condition based on the particle velocity. The stability condition given above is not very restrictive for such flows, since the ratio of this time step (14) to the time step for the particle motion is proportional to the Mach number.

Chen [3] gives for a D_2Q_9 "stream-and-collide" lattice Boltzmann method the time step of

$$\Delta t \leq \frac{\Delta x}{\sqrt{3}c} \approx \frac{0.6\Delta x}{c}. \tag{15}$$

This suggests that the stable time step of (15) is quite similar to (14).

3 Driven Cavity Flow Problem

The driven cavity flow problem, in which a fluid-filled square box (cavity) is swirled by a uniformly translating lid as shown in Figure 1, is a classic problem in CFD. This problem is unambiguous with easily applied boundary conditions and has a wealth of documented analytical and computational results, for example Ghia et al. [6]. We will solve this flow using the explicit MacCormack scheme discussed in the previous section.

We may nondimensionlize the problem with the following scaling: lengths with D, velocity with U, time with D/U, density with a reference density ρ_0, and pressure with $\rho_0 U^2$. Using this scaling, the equation of state $p = c^2\rho$ becomes $p = \rho/M^2$, where $M = U/c$ is the Mach number. The Reynolds number is defined as $\mathrm{Re} = \rho_0 UD/\mu$.

The boundary conditions for this problem are relatively simple. The velocity components on all four sides of the cavity are well defined. There are two singularities of velocity gradient at the two top corners where velocity u drops from U to 0 directly underneath the sliding lid. However, these singularities will be smoothed out on a given grid, since the change of the velocity occurs linearly between two grid points. The boundary conditions for the density

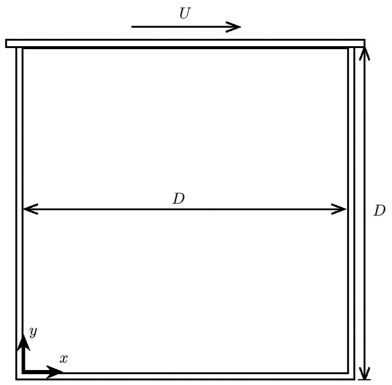


Fig. 1. Driven cavity flow problem. The cavity is filled with a fluid. The top is sliding with a constant velocity U.

(hence the pressure) are more involved. Since the density is not specified on a solid surface, an update scheme for values of density on all boundary points is needed. One option is to derive it using the continuity equation. That option works well on surfaces that lie exactly on the background grid, and it can be used for this problem. The momentum equation gives better results for curved surfaces, and will be discussed later.

Consider the boundary on the left (at $x = 0$). Since $v = 0$ along the surface, the continuity equation (2) reduces to

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} = 0. \tag{16}$$

One can use a predictor-corrector scheme to update density on this surface with a one-sided second-order accurate discretization for the spatial derivative,

$$\left(\frac{\partial f}{\partial x}\right)_i = \frac{1}{2\Delta x}\left(-f_{i+2} + 4f_{i+1} - 3f_i\right) + O\left(\Delta x^2\right). \tag{17}$$

Therefore, on the surface of $x = 0$ (for $i = 0$ including two corner points on the left), we have the following update scheme for density,

$$\begin{aligned} &\text{predictor } \rho^*_{i,j} = \rho^n_{i,j} - \tfrac{\Delta t}{2\Delta x}\left[-\left(\rho u\right)^n_{i+2,j} + 4\left(\rho u\right)^n_{i+1,j} - 3\left(\rho u\right)^n_{i,j}\right], \\ &\text{corrector } \rho^{n+1}_{i,j} = \tfrac{1}{2}\left[\rho^n_{i,j} + \rho^*_{i,j} - \tfrac{\Delta t}{2\Delta x}\left[-\left(\rho u\right)^*_{i+2,j} + 4\left(\rho u\right)^*_{i+1,j} - 3\left(\rho u\right)^*_{i,j}\right]\right]. \end{aligned} \tag{18}$$

Similarly, on the right side of the cavity (for $x = D$, $i = nx - 1$ where nx is the number of grid points in the x-direction, including two corner points on

the right),

$$\begin{aligned}&\text{predictor } \rho^*_{i,j} = \rho^n_{i,j} + \tfrac{\Delta t}{2\Delta x}\left[-(\rho u)^n_{i-2,j} + 4(\rho u)^n_{i-1,j} - 3(\rho u)^n_{i,j}\right],\\ &\text{corrector } \rho^{n+1}_{i,j} = \tfrac{1}{2}\left[\rho^n_{i,j} + \rho^*_{i,j} + \tfrac{\Delta t}{2\Delta x}\left[-(\rho u)^*_{i-2,j} + 4(\rho u)^*_{i-1,j} - 3(\rho u)^*_{i,j}\right]\right].\end{aligned} \tag{19}$$

On the bottom of the cavity $y = 0$ ($j = 0$),

$$\begin{aligned}&\text{predictor } \rho^*_{i,j} = \rho^n_{i,j} - \tfrac{\Delta t}{2\Delta y}\left[-(\rho v)^n_{i,j+2} + 4(\rho v)^n_{i,j+1} - 3(\rho v)^n_{i,j}\right],\\ &\text{corrector } \rho^{n+1}_{i,j} = \tfrac{1}{2}\left[\rho^n_{i,j} + \rho^*_{i,j} - \tfrac{\Delta t}{2\Delta y}\left[-(\rho v)^*_{i,j+2} + 4(\rho v)^*_{i,j+1} - 3(\rho v)^*_{i,j}\right]\right].\end{aligned} \tag{20}$$

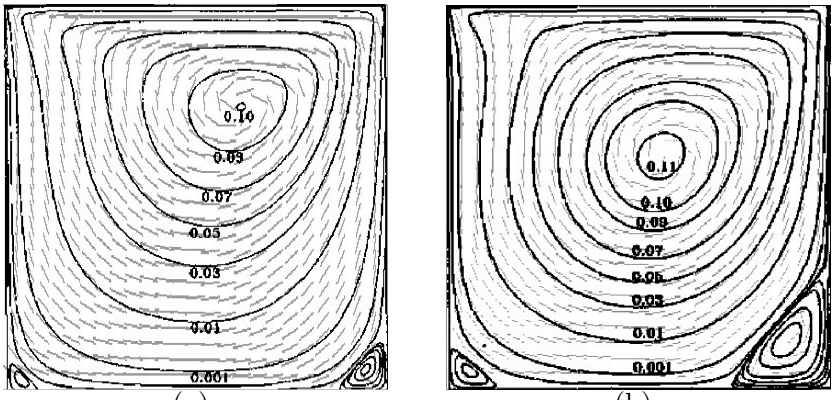
Finally, on the top of the cavity $y = D$ ($j = ny - 1$ where ny is the number of grid points in the y-direction), the density needs to be updated from slightly different expressions since $\partial \rho u/\partial x = U \partial \rho/\partial x$ is not zero there,

$$\begin{aligned}&\text{predictor}\\ &\rho^*_{i,j} = \rho^n_{i,j} - \tfrac{\Delta t U}{2\Delta x}\left[\rho^n_{i+1,j} - \rho^n_{i-1,j}\right] + \tfrac{\Delta t}{2\Delta y}\left[-(\rho v)^n_{i,j-2} + 4(\rho v)^n_{i,j-1} - 3(\rho v)^n_{i,j}\right],\\ &\text{corrector}\\ &\rho^{n+1}_{i,j} = \tfrac{1}{2}\left[\rho^n_{i,j} + \rho^*_{i,j} - \tfrac{\Delta t U}{2\Delta x}\left[\rho^*_{i+1,j} - \rho^*_{i-1,j}\right] + \tfrac{\Delta t}{2\Delta y}\left[-(\rho v)^*_{i,j-2} + 4(\rho v)^*_{i,j-1} - 3(\rho v)^*_{i,j}\right]\right].\end{aligned} \tag{21}$$

Next we present some of the results and compare them with those in the paper by Hou et al. [7] obtained by a lattice Boltzmann method. Hou et al. has been carefully validated against Ghia et al. in [7]. To keep the flow almost incompresible, the Mach number is chosen as $M = 0.1$. Flows with two Reynolds numbers, Re $= \rho_0 UD/\mu$=100 and 400 are simulated. At these Reynolds numbers, the flow will eventually be steady. Thus calculations need to be run long enough to get to the steady state. A uniform grid of 256 by 256 was used for this example.

Figure 2 shows comparisons of the velocity field calculated by the explicit MacCormack scheme with the streamlines from Hou at Re=100 and 400. The agreement seems reasonable. It was also observed that the location of the center of the primary eddy agrees even better. When Re=100, the center of the primary eddy is found at $(0.62 \pm 0.02, 0.74 \pm 0.02)$ from the MacCormack scheme in comparison with (0.6196, 0.7373) from Hou. When Re=400, the center of the primary eddy is found at $(0.57 \pm 0.02,\ 0.61 \pm 0.02)$ from the MacCormack scheme in comparison with (0.5608, 0.6078) from Hou.

Figure 3 contains a comparison of pressure contours at Re=400 calculated from the explicit MacCormack scheme (light gray lines) with those of Hou (dark



(a) (b)

Fig. 2. Comparisons of results from the explicit MacCormack scheme (light gray, velocity vector field) and those from Hou et al. [7] (dark solid streamlines) calculated using a Lattice Boltzmann Method. (a) Re=100, (b) at Re=400.

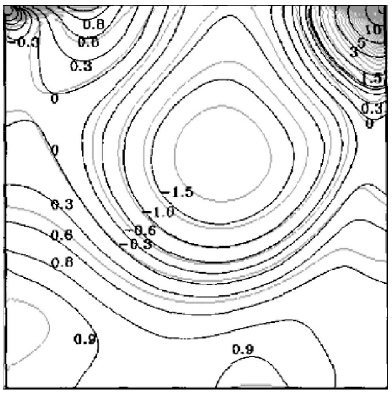


Fig. 3. Comparison of pressure contours at Re=400. The light gray lines are from the explicit MacCormack scheme. The dark solid lines are from Hou et al. [7].

solid lines). The contour lines from the explicit MacCormack scheme were selected at even intervals between the minimum and the maximum values of pressure. However, the contour lines from Hou were presented differently, thus the values of those contour lines do not correlate exactly. The overall pattern of the pressure field matches. For a more quantitative comparison, Figure 4 plots the velocity profile along a vertical line cut through the center of the cavity ($x = D/2$). The velocity profiles for two Reynolds numbers, Re=100 and 400, are compared. The results from the explicit MacCormack scheme are shown in solid and dashed lines. The data points in symbols were directly converted from Hou's paper. The agreement is excellent.

4 Flow Over a Circular Cylinder Between Sliding Walls

To test the scheme for the case of flow over an immersed body, we simulated a circular cylinder in a channel with two moving side walls and a uniform

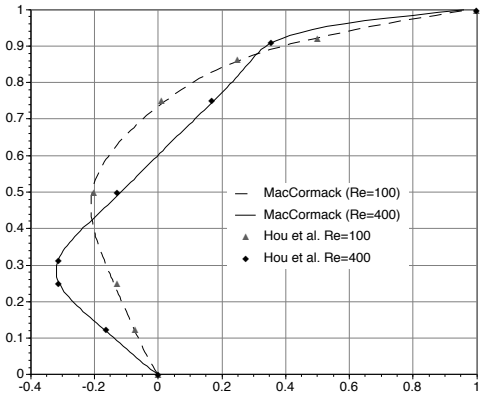


Fig. 4. Comparison of velocity profiles along a line cut through the center of the cavity (x=0.5D) at Re=100 and 400.

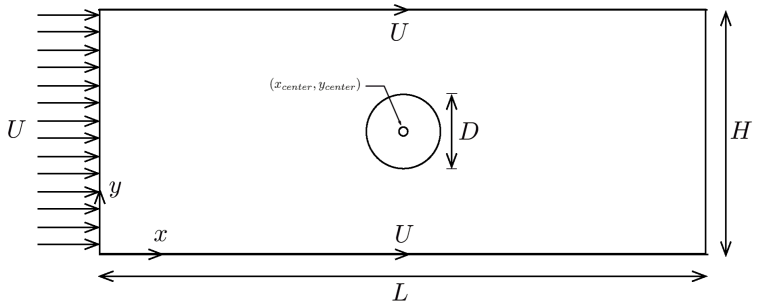


Fig. 5. Flow over a circular cylinder between two sliding walls. Inlet and wall velocities are both U, channel length is L, channel width is H, cylinder diameter is D, and the center of the cylinder is at (x_{center}, y_{center}).

inlet velocity profile (Figure 5). This problem is mathematically equivalent to the problem of a cylinder moving in a fluid at constant speed down a channel with stationary walls, although the numerical treatment of these two cases differ. Three new issues arise that have not been discussed above. First, how should points be chosen on a uniform background grid so as to give the best approximation of the curved surface of the cylinder? Second, how should the no slip condition be enforced when the boundary gridpoints do not lie precisely on the surface of the cylinder? Finally, how should the density boundary condition be implemented for the boundary gridpoints?

4.1 Selecting Boundary Points

The boundary points on the grid should be selected to be as close to the actual cylinder surface as possible. The current algorithm is specific to the geometry of a circular cylinder, but it can be straightforwardly generalized to handle arbitrary geometry (in 2D). The steps are:

(1) If the cylinder's center is (x_{center}, y_{center}) and the radius is $a = D/2$, then

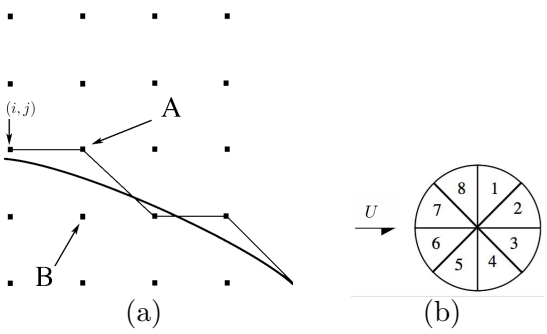


Fig. 6. (a) The candidate points A and B in the first octant of the cylinder that correspond to (i, j). (b) The cylinder octants, numbered.

the first boundary point will be near $(x_{center}, y_{center} + a)$. Choose it to be $(\mathrm{i}, \mathrm{j}) = \left(round(\frac{x_{center}}{\Delta x}), round(\frac{y_{center}+a}{\Delta y})\right)$ where the *round* function rounds its argument to the nearest integer.

(2) Next, consider the two points $A = (i+1, j)$ and $B = (i+1, j-1)$ in Figure 6. These will be the candidates for the next boundary point. Calculate the distance between each candidate and the cylinder surface. Select the point that lies nearest the surface as the new (i, j).

(3) Repeat step 2 until $x - x_{center} = y - y_{center}$; this is where the next octant of the cylinder surface starts. In the second octant, the candidate points will $A = (i + 1, j - 1)$ and $B = (i, j - 1)$. Stop when $y - y_{center} = 0$.

(4) Repeat all of the above for the other 6 octants, with appropriate choices of A and B for each octant.

Note that if (i, j) is a candidate boundary point, the point on the surface nearest (i, j), called (x_s, y_s), can be found from

$$\mathbf{x}_s = \mathbf{x}_{center} + \frac{a\left(\mathbf{x}_{ij} - \mathbf{x}_{center}\right)}{|\mathbf{x}_{ij} - \mathbf{x}_{center}|} \tag{22}$$

where $\mathbf{x}_s = (x_s, y_s)$, $\mathbf{x}_{center} = (x_{center}, y_{center})$, and $\mathbf{x}_{ij} = (i\Delta x, j\Delta y)$. For a circular cylinder, the point on the surface that minimizes the distance to (i, j) will lie on the line that passes through both (i, j) and the circle's center. This algorithm will work even when the center of the cylinder is not on a gridpoint, a property that will be necessary in later problems in which the cylinder is allowed to drift.

4.2 The No-Slip Condition

One would think that applying the no-slip condition on a stationary cylinder would be as easy as setting the velocity components equal to zero on all the boundary gridpoints. This method will yield the jagged pressure distribution shown in Figure 7. The trouble is that the flow would see the boundary gridpoints as if they are the true cylinder surface— a rather jagged surface. To

avoid this, one can Taylor expand from the boundary gridpoint to the true surface and enforce no-slip at a point on the true surface, similar to the interpolation method of Udaykumar et al. [20]. As a byproduct of the boundary point selection algorithm described in the previous section, the point on the cylinder surface nearest to each boundary point, (x_s, y_s), was calculated. It is natural to enforce no-slip on these points, because it minimizes the truncation error in the Taylor series.

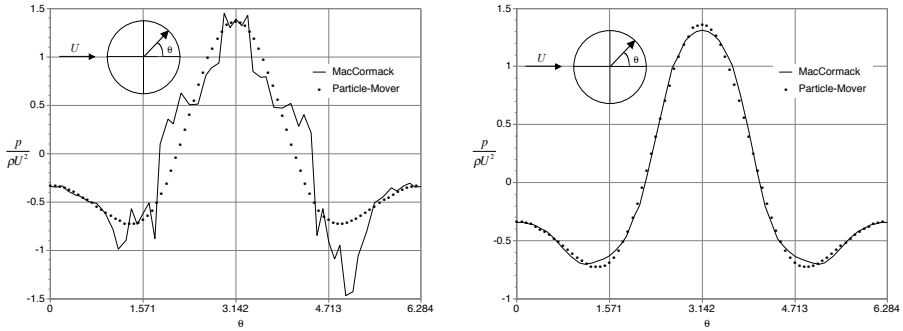


Fig. 7. Non-dimensional pressure distribution (Re = 20, M = 0.05, $\Delta x = \Delta y = D/20$) with (a) $u = v = 0$ on the boundary points, or (b) $u = v \approx 0$ on the surface point nearest each boundary point. The dots are the results computed by ALE Particle-Mover [10].

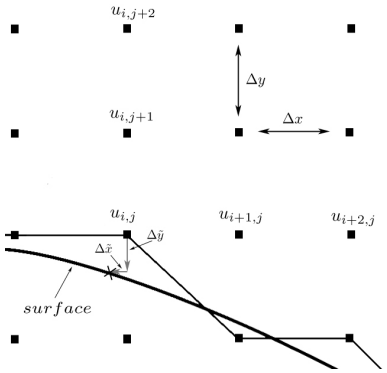


Fig. 8. A magnified portion of the grid near the particle surface, with quantities labeled.

At a given time step, all the bulk fluids points outside the cylinder have already been updated before we update the boundary. Suppose one wants to calculate the velocity u for a boundary point (i, j) in the first quadrant of the cylinder (see Figure 8). The velocity will not be zero at (i, j) because no-slip is to be applied on the cylinder surface, not the boundary point. Call the horizontal distance from (i, j) to (x_s, y_s), as shown in Figure 8, $\Delta\tilde{x}$, and the vertical distance $\Delta\tilde{y}$, and Taylor expand from (i, j) to (x_s, y_s):

$$u_{surface} \equiv u_{sur} = u_{i,j} + \Delta\tilde{x}\,(u_x)_{i,j} + \Delta\tilde{y}\,(u_y)_{i,j} + O(\Delta\tilde{x}^2, \Delta\tilde{x}\Delta\tilde{y}, \Delta\tilde{y}^2). \quad (23)$$

The derivatives u_x, and u_y can be eliminated by substituting the one-sided

finite differences,

$$(u_x)_{i,j} = \frac{1}{2\Delta x}\left(-u_{i+2,j} + 4u_{i+1,j} - 3u_{i,j}\right) + O(\Delta x^2),$$
$$(u_y)_{i,j} = \frac{1}{2\Delta y}\left(-u_{i,j+2} + 4u_{i,j+1} - 3u_{i,j}\right) + O(\Delta y^2). \tag{24}$$

The quantities in the bulk of the fluid, $u_{i+2,j}, u_{i+1,j}, u_{i,j+1}$, and $u_{i,j+2}$ are all known, since the bulk is updated before the boundary points. Substituting (24) into (23) and solving for $u_{i,j}$, one gets

$$(u)_{i,j} = \frac{-\Delta\tilde{x}\Delta y\left(-u_{i+2,j} + 4u_{i+1,j}\right) - \Delta x\Delta\tilde{y}\left(-u_{i,j+2} + 4u_{i,j+1}\right) + 2\Delta x\Delta y\left(u_{sur}\right)}{2\Delta x\Delta y - 3\left(\Delta\tilde{x}\Delta y + \Delta\tilde{y}\Delta x\right)}. \tag{25}$$

The procedure is identical for $v_{i,j}$; replace u with v in eqn. 25. Similarly, for the other three quadrants, just change the sign of $\Delta x, \Delta y$, and the indices to get the appropriate result.

4.3 Density Boundary Condition on Particle Surface

The continuity equation provides one possible boundary condition for density (or pressure, since they are proportional here). When this condition is used on the cylinder, it results in a jagged pressure distribution on the surface. The reason for this is not yet understood. A boundary condition based on the momentum equation gave better results. This condition will now be derived.

The component of the pressure gradient normal to the cylinder surface is just $\hat{n} \cdot \nabla p$, with equations (3) and (4) substituted in,

$$\frac{\partial\rho}{\partial n} = \frac{\mu}{c^2}\left(\frac{\hat{n}_x}{3}\left(4\frac{\partial^2 u}{\partial x^2} + 3\frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 v}{\partial x\partial y}\right) + \frac{\hat{n}_y}{3}\left(3\frac{\partial^2 v}{\partial x^2} + 4\frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 u}{\partial y\partial x}\right)\right)$$
$$- \frac{\rho}{c^2}\left(\hat{n}_x\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}\right) + \hat{n}_y\left(\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y}\right)\right) \tag{26}$$

where the equation of state $p = c^2\rho$ has been used to replace pressure with density, and $\hat{\mathrm{n}}_x$ and $\hat{\mathrm{n}}_y$ are the x- and y-components of the surface normal. Each term can be evaluated from one-sided derivatives.

In the first quadrant, we used the one-sided 2nd order differences,

$$
\begin{aligned}
(u_{xx})_{i,j} &= \tfrac{1}{\Delta x^2}\left(2u_{i,j} - 5u_{i+1,j} + 4u_{i+2,j} - u_{i+3,j}\right) + O(\Delta x^2), \\
(u_{yy})_{i,j} &= \tfrac{1}{\Delta y^2}\left(2u_{i,j} - 5u_{i,j+1} + 4u_{i,j+2} - u_{i,j+3}\right) + O(\Delta y^2), \\
(u_{xy})_{i,j} &= \tfrac{1}{4\Delta x \Delta y}\begin{pmatrix} -\left(-u_{i+2,j+2} + 4u_{i+1,j+2} - 3u_{i,j+2}\right) \\ +4\left(-u_{i+2,j+1} + 4u_{i+1,j+1} - 3u_{i,j+1}\right) \\ -3\left(-u_{i+2,j} + 4u_{i+1,j} - 3u_{i,j}\right) \end{pmatrix} + O(\Delta x^2, \Delta x \Delta y, \Delta y^2),
\end{aligned}
\tag{27}
$$

and the same differences for the v and ρ derivatives.

Putting the pieces together, the density boundary condition from the momentum equation is

$$
\frac{\partial \rho}{\partial n} = \frac{\mu}{c^2} A - \rho B, \tag{28}
$$

where

$$
\begin{aligned}
A &= \frac{1}{3}\left(\hat{n}_x\left(4\frac{\partial^2 u}{\partial x^2} + 3\frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 v}{\partial x \partial y}\right) + \hat{n}_y\left(3\frac{\partial^2 v}{\partial x^2} + 4\frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 u}{\partial y \partial x}\right)\right), \\
B &= \frac{1}{c^2}\left(\hat{n}_x\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}\right) + \hat{n}_y\left(\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y}\right)\right).
\end{aligned}
$$

Since the bulk fluid (non-boundary) gridpoints are evaluated first, $A_{i,j}^{n+1}$ is known. Expanding the ρ derivatives in the first quadrant gives

$$
\frac{\hat{\mathrm{n}}_x}{2\Delta x}\left(-\rho_{i+2,j} + 4\rho_{i+1,j} - 3\rho_{i,j}\right) + \frac{\hat{\mathrm{n}}_y}{2\Delta y}\left(-\rho_{i,j+2} + 4\rho_{i,j+1} - 3\rho_{i,j}\right) + \rho_{i,j} B_{i,j} = \frac{\mu}{c^2} A_{i,j}
$$

which can be rearranged to solve for ρ^{n+1} in terms of the known quantities from the bulk fluid, all evaluated at time $(n+1)$.

$$
\begin{aligned}
\rho_{i,j}^{n+1} = &-\frac{\hat{\mathrm{n}}_x \Delta y}{\left(-3\hat{\mathrm{n}}_x \Delta y - 3\hat{\mathrm{n}}_y \Delta x + 2\Delta x \Delta y B_{i,j}\right)}\left(-\rho_{i+2,j}^{n+1} + 4\rho_{i+1,j}^{n+1}\right) \\
&- \frac{\hat{\mathrm{n}}_y \Delta x}{\left(-3\hat{\mathrm{n}}_x \Delta y - 3\hat{\mathrm{n}}_y \Delta x + 2\Delta x \Delta y B_{i,j}\right)}\left(-\rho_{i,j+2}^{n+1} + 4\rho_{i,j+1}^{n+1}\right) \\
&+ \frac{2\Delta x \Delta y\left(\mu/c^2\right)}{\left(-3\hat{\mathrm{n}}_x \Delta y - 3\hat{\mathrm{n}}_y \Delta x + 2\Delta x \Delta y B_{i,j}\right)} A_{i,j}^{n+1}
\end{aligned}
\tag{29}
$$

See Figure 7b for an example of the results obtained with this boundary condition on a mesh with 20 grid spacings across the cylinder diameter. There is reasonable agreement with the results of Particle-Mover finite element analysis. Particle-Mover is described in [9], [8], and [10]. The Particle-Mover pro-

gram has been validated against the literature and against some analytical results in [8] and [10].

4.4 Summary of Stationary Cylinder Algorithm

Here is a brief summary of the algorithm for flow around a stationary cylinder.

(1) Initialize all the fields. The density, ρ, should be set to one. In our tests, velocity was generally set to $u = 1, v = 0$.
(2) March in time from t_n to t_{n+1}:
   (a) In the bulk of the fluid, update ρ^*, $(\rho u)^*$, and $(\rho v)^*$ in that order using the predictor equations (6)-(8).
   (b) Calculate $u = (\rho u)^*/(\rho^*)$ and $v = (\rho v)^*/(\rho^*)$ in the bulk of the fluid.
   (c) Calculate u, v on the boundary points using the Taylor series technique to satisfy no slip on the actual cylinder surface. $u = U, v = 0$ on the sidewalls. $\frac{\partial(\rho u)}{\partial x} = \frac{\partial(\rho v)}{\partial x} = 0$ at the exit, implemented as one-sided second order finite differences, e.g. $(\rho u)^*|_{exit=i,j} = (4(\rho u)^*_{i-1,j} - (\rho u)^*_{i-2,j})/3$. Momentum BC is used on all boundary nodes (both cylinder and all four walls) to update ρ.
   (d) In the bulk of the fluid, update ρ^{n+1}, $(\rho u)^{n+1}$, and $(\rho v)^{n+1}$ in that order using the corrector equations (9)-(11). These will be stored in the same variables used for time n.
   (e) Calculate $u = (\rho u)^{n+1}/(\rho^{n+1})$ and $v = (\rho v)^{n+1}/(\rho^{n+1})$ in the bulk of the fluid.
   (f) Calculate u, v on the boundary points. $u = 1, v = 0$ on the sidewalls. $\frac{\partial(\rho u)}{\partial x} = \frac{\partial(\rho v)}{\partial x} = 0$ at the exit, implemented as one-sided second order finite differences, but now using the (n+1) variables. Momentum BC is used on all boundary nodes (both cylinder and all four walls) to update ρ.

4.5 Results

In this section, the MacCormack scheme for flow over a cylinder is validated. In the tests that follow, $L = 35D$, $H = 4D$, and $\Delta x = \Delta y$ in all cases. The cylinder center is 15.5 diameters from the inlet. All lengths have been non-dimensionalized using the cylinder diameter D, and the pressure and shear stress with ρU^2. We examine lift and drag coefficients C_L and C_D as functions of time. ($C_L = 2F_y/(\rho U^2 D)$ and $C_D = 2F_x/(\rho U^2 D)$ where F_x and F_y are the computed drag and lift forces.) Lift and Drag coefficients were determined by numerical integration of the pressure and shear stress on the particle surface using the trapezoidal rule. When calculating the pressure and shear stress on the particle surface, the density and velocity gradient terms were Taylor ex-

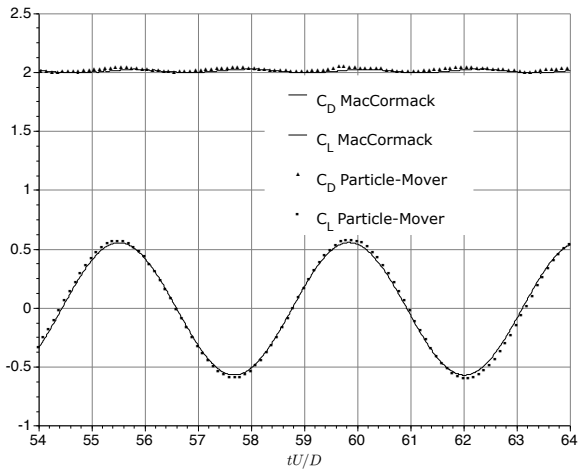


Fig. 9. C_D and C_L vs. time at Re=100. There are 40 grid spacings across the cylinder diameter, and the Mach number is 0.05. The Strohl number for the MacCormack scheme is 0.233, and for Particle-Mover it is 0.227. Note that the walls are only 1.5D away from the cylinder.

panded from the boundary point to the surface in the same manner described above for the velocity.

The lift and drag coefficients as a function of time are shown for Re=100 (M=0.05) in Figure 9. Several important features are evident. The oscillation of the lift coefficient C_L is induced by vortex shedding.

Figures 10a and 10b show convergence of the drag coefficient of the circular cylinder with the momentum boundary condition at different Mach and Reynolds numbers. The incompressible results for the same geometry, computed with the Particle-Mover finite element package, are also shown, at a resolution of 20 elements across the cylinder diameter. The Mach number does make a difference to the final result. Note that the dimensionless time step was set proportional to the grid spacing in these tests ($\Delta t = \sigma \Delta x M$), to satisfy the stability condition. The grid spacing Δx is non-dimensionalized with the cylinder diameter D, Δt with D/U, and the Mach number is U/c. For the case of $Re = 100$, the value plotted is the arithmetic mean drag coefficient over several vortex-shedding cycles.

The pressure and shear stress distributions on a fine mesh (40 grid spacings across the cylinder diameter) are shown in Figure 11. The non-dimensional pressure distribution plot also includes the results of a Particle-Mover finite element calculation with 20 elements across the cylinder diameter. At Re=100, the flow is unsteady, so the pressure and shear distribution are instantaneous values at non-dimensional time $t = 77.36$. Time was non-dimensionalized by L/U.

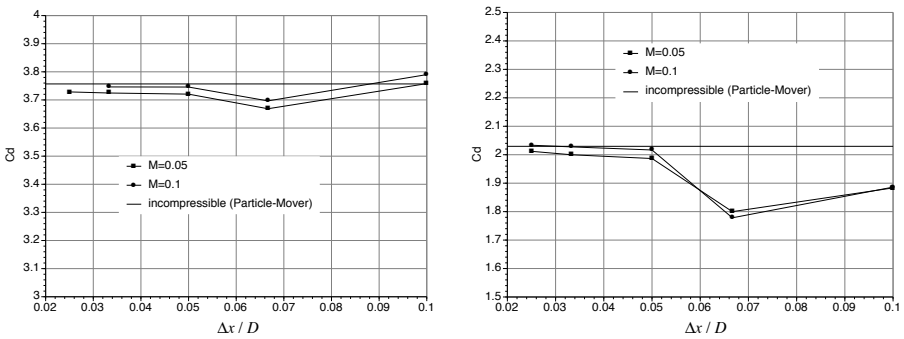


Fig. 10. (a) Convergence of drag coefficient for Re=20 at M=0.05 and 0.1. (b) Re=100 at M=0.05 and 0.1. In both graphs, the results of the finite element code Particle-Mover are shown, computed with 20 elements across the cylinder diameter.

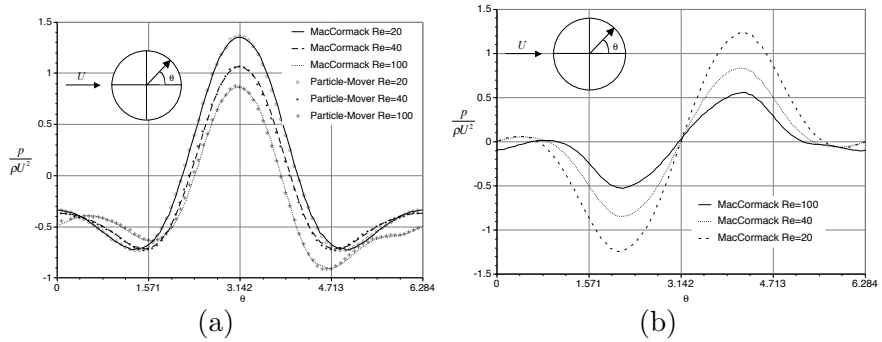


Fig. 11. (a) Non-dimensional pressure distribution. (b) Non-dimensional shear stress distribution. $Re = 20, 40$, and 100; $M = 0.05$; $\Delta x = \Delta y = 1/40$.

5 Circular Cylinder Translating at Constant Speed

Next we consider the case of a cylinder dragged at constant speed through a stationary fluid in a channel. This introduces some new implementation issues. The boundary gridpoints must be picked anew each time the cylinder moves, resulting in points that were formerly inside the cylinder leaking from the rear. (See Figure 12.) In the previous case of the stationary cylinder, these interior points did not need values assigned to them— they were invisible to the fluid. Since this is no longer the case, the velocity and density must now be chosen for these points.

Our method of dealing with the "leaking" of interior points into the bulk fluid is to assign them reasonable values before the cylinder is moved. Only the first layer of interior points needs to be considered, since the particle always moves less than one grid spacing per time step due to the CFL condition. The first step is to identify these interior points. The boundary gridpoints are already known, so the first layer of interior points is simply all the points immediately adjacent inside the cylinder surface. For example, consider a boundary point in the first quadrant. Then points to the left, down, and diagonally left and

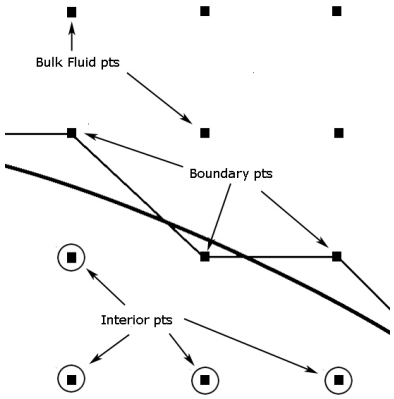


Fig. 12. Gridpoints in the vicinity of the top rear surface (first quadrant) of the cylinder. Interior points (circled) are inside the cylinder surface, and boundary points (gray squares) are the points nearest the surface.

down will all be considered part of the interior layer. By marching around the boundary, a list may be generated of points on the interior layer corresponding to each boundary point. The next step makes use of the fact that the boundary points have already been updated. Expanding in a Taylor series from the boundary point to each associated interior point, the density and the velocity components are assigned to the interior points. (The derivatives of the velocity and density at the boundary point are already known from the boundary point update.)

5.1 Results

The drag and lift coefficients for the translating cylinder, shown in Figures 13 and 14, demonstrate that the results obtained for a moving cylinder in a stationary fluid agree with those for a stationary cylinder in a moving fluid. This is expected, but given that the boundary points representing the cylinder are changing with each iteration, direct confirmation is important. In both cases, the Mach number was 0.05. For the stationary case, the cylinder center was 15.5 diameters from the inlet, and the channel length was 35 diameters. The channel length for the moving cylinder was 70 diameters. In both cases, the cylinder was centered horizontally in the channel, which was 4 diameters wide. For the case of the stationary cylinder, the outflow boundary condition was used. The channel for the moving cylinder was closed (velocity set to zero) on all boundaries. In both cases there were 40 grid spacings across the cylinder diameter, and the time step was chosen according to equation (14) with a safety factor of 0.5.

The results for the moving cylinder show some high-frequency "noise" caused by the change in the boundary points used to represent the cylinder as the cylinder surface moves. Additionally, sound waves caused by the initial tran-

sient continue to bounce between the ends of the channel. The sound wave results are more visible in the stationary cylinder case because the shorter channel causes them to have a higher frequency (like the short pipes in a pipe organ). They are also more visible at high Reynolds number because there is less damping.

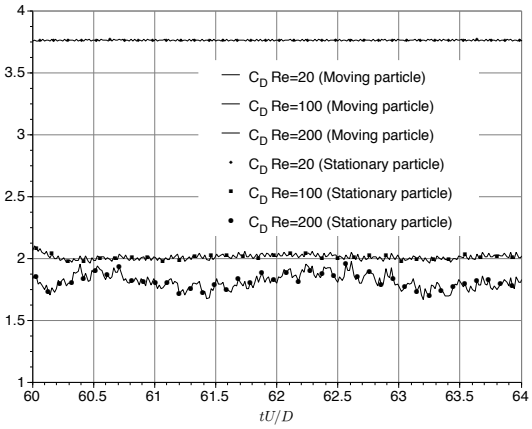


Fig. 13. The drag coefficient C_D versus time for Reynolds 20, 100, and 200. The thin lines represents the moving cylinder, and the dots represents the stationary cylinder.

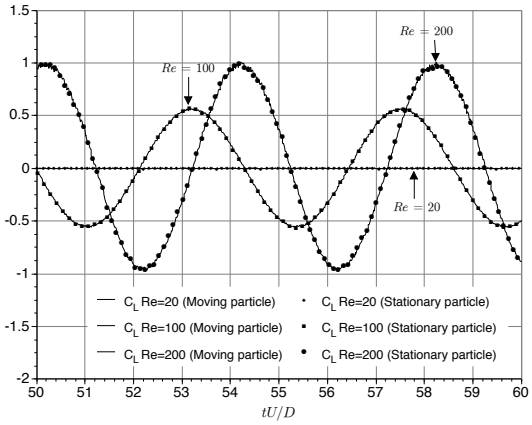


Fig. 14. The lift coefficient C_L versus time for Reynolds 20, 100, and 200. The thin lines represents the moving cylinder, and the dots represents the stationary cylinder.

6 Freely Falling Cylinders

The translating cylinder simulation may now be modified to deal with the case of freely falling cylinders by adding a collision scheme. 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. It will also be necessary to integrate the equations of motion for the cylinders, but this is done as a part of the collision scheme. For some details

on the collision scheme, see [10]. Broadly, the idea is to define a "safety zone" of width δ around each particle (walls are just particles of infinite mass and radius) such that when the surface of one particle enters the safety region of another, a contact force is generated that is exactly enough to separate the two particles by δ. (See Figure 15.) This represents an inelastic collision, which is reasonable because in principle the lubrication forces should cause the particles to come to rest relative to each other. Because the collision scheme itself is slightly ad hoc, the width of the safety zone should be kept as small as possible, but sufficiently large to ensure that enough gridpoints remain in the fluid to evaluate the boundary conditions when a collision occurs.

The equations of motion, which are solved as part of the collision scheme, are the result of a force balance on each particle. They are given as

$$\begin{aligned} m\frac{d\mathbf{U}_{center}}{dt} &= \int_S \sigma\hat{\mathbf{n}}dS + (m - m_{fluid})\,\mathbf{g} \\ I\frac{d\omega}{dt} &= \mathbf{k}\cdot\int_S \mathbf{r}\times\left(\tau\hat{\mathbf{t}}\right)dS \end{aligned} \tag{30}$$

where m and I are the particle mass and moment of inertia, m_{fluid} is the mass of fluid displaced by the particle, σ is the total stress tensor, τ is shear stress, $\mathbf{g}$ is the gravitational acceleration, $\mathbf{U_{center}}$ is the velocity of the particle center, ω is the angular velcoity, and the integrals are taken on the surface of the particle. Finally, the system is closed using the the kinematic relationships,

$$\begin{aligned} \dot{x}_{center} &= U_{center}, \\ \dot{y}_{center} &= V_{center}. \end{aligned} \tag{31}$$

It is not necessary to calculate θ because the particles have circular symmetry.

A term corresponding to the weight of fluid displaced by the particle—the buoyancy—is included in the particle equations of motion (30). In the momentum equations for the fluid, the body force was accounted for by replacing the pressure p with a dynamic pressure, $p+\rho_0 gx$, where g is taken as positive, and the particle falls in the $-x$ direction. This is implemented by taking the fluid density to be a constant reference density ρ_0. The equations of motion, (2)-(4), will not change, but in these equations the density ρ should be reinterpreted as a density perturbation ρ' such that $\rho = \rho_0(1 + gx/c^2) + \rho'$. In equation (30), $m_{fluid} = \rho_0(\pi D^2/4)$.

For the case of a single particle dropped at a terminal particle Reynolds number of about 20, the particle remained in the center of the channel. (There was some off-center wandering due asymmetries in the MacCormack discretization, but in all cases, the wandering was by less than a single grid spacing.) Results for the acceleration, velocity, and position versus time are shown in Figures 16, 17, and 18.

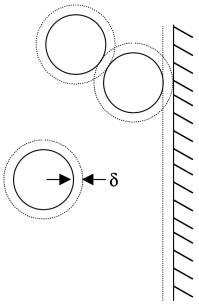


Fig. 15. Several particles colliding with a wall. The dotted lines indicate the safety region around each particle. The actual thickness of the safety region should be much smaller than represented in the figure.

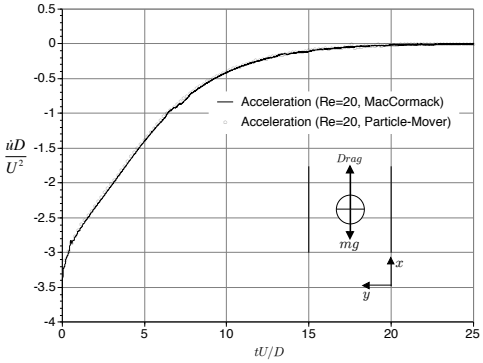


Fig. 16. Dimensionless acceleration versus tU/D for a single falling particle. U is the terminal velocity.

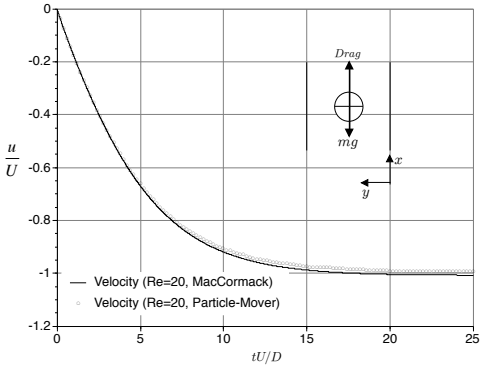


Fig. 17. Velocity (non-dimensionalized by U) versus non-dimensional time tU/D for a single falling particle. U is the terminal velocity.

For this test, there were 40 grid spacings across the particle diameter. The channel length was 70 diameters and its width was 4 diameters. The time step was given by the stability criterion (14). The particle center was initially placed at the center of the channel and five diameters from the top. The density ratio of the solid to the liquid was 10.55. The Reynolds number based on particle diameter and terminal velocity was 19.72, and the Mach number at terminal velocity was 0.0493.

If two particles in a fluid are initially placed one above the other and re-

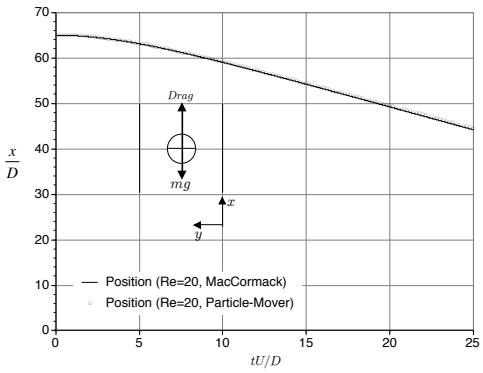


Fig. 18. Position (non-dimensionalized by D) versus non-dimensional time tU/D for a single falling particle. U is the terminal velocity.

leased, a well-known sequence of events occurs. The top particle initially falls faster than the bottom particle because of entrainment by the bottom particle's wake. This is called "drafting." The two particles touch ("kiss"), and then they rotate until they are side-by-side ("tumble"). This is because two adjacent particles in a flow will act like a single long body, and asymmetrical bodies falling lengthwise are unstable to small perturbations in their orientation (for example, canoes will orient themselves perpendicular to the current if left alone). The particle that was formerly on the bottom will then be pushed sideways by the tumbling motion and drift upward relative to the 'top' particle, which is now on the bottom. If sufficient time passes, the motion can repeat. The drafting-kissing-tumbling sequence was first observed experimentally by Fortes et al. [5], simulated in two dimensions by Hu et al. [9] and in three dimensions by Johnson et al. [11].

This behavior was observed in the MacCormack simulation when the two particles were initially separated by two diameters (measured center to center). In the simulation, the sidewalls, top, and bottom move upward at a prescribed speed U, and the fluid and particles are given an initial upward velocity of $u = U$ and $v = 0$, to simulate the effect of a camera following the particle.

The drafting-kissing-tumbling sequence is illustrated in Figure 19. The locus of particle positions in the laboratory reference frame is shown in Figure 20 alongside the results from Particle-Mover. The density ratio of the solid phase to the fluid was 1.04. There were 30 grid spacings across the diameter of each particle. The initial particle separation was two diameters. The channel was 8 diameters wide, and there were 40 diameters between the top and bottom (the "moving camera" means the channel length is effectively infinite). The safety zone thickness, δ, was set to three grid spacings, or a tenth of the particle diameter. (δ is limited by the number of grid spacings required to compute the derivatives in the boundary conditions.) The maximum Reynolds number was around 80. The results of the finite element solution and the MacCormack agree only qualitatively because the details of the collision scheme differ. After the tumble, wiggles in the particle path appear due to vortex shedding.

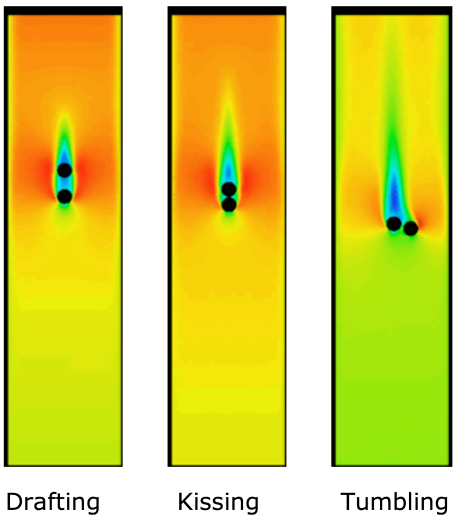
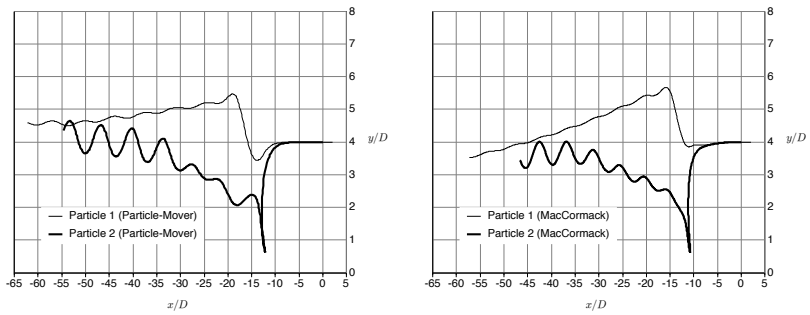


Fig. 19. Drafting, kissing, and tumbling of two particles. Color indicates the relative values of u-velocity at a given instant. $\rho_s/\rho_f = 1.04$, $Re \approx 80$.



(a) (b)

Fig. 20. The paths of the center of each particle in the drafting-kissing-tumbling sequence. The particles fall in the $-x$ direction. The color indicates the relative values of v-velocity at each instant. For the MacCormack result, $\delta/D = 7.5\%$, and for Particle-Mover it was $\delta/D = 3\%$. (a) Particle-Mover finite element result. (b) MacCormack result. In both cases, the thick line corresponds to the particle initially on the bottom.

A horizontal line of particles falling in a channel is known to be a particularly stable configuration. If another particle is dropped on top of the line, it will displace the particle it falls on, and the line will reform. The sequence of events is shown in Figure 21. The line appears to move upward because the camera is moving down faster than the particles are falling.

For a given resolution, the computational effort required to compute any number of particles hardly changes if the collision scheme is efficient. The collision algorithm used in the present work is the same as the one in Particle-Mover, but the implementation is different. Some details of the algorithm are in [10]. The Particle-Mover implementation was found to be efficient when used in that program, representing less than one percent of the processing time. The

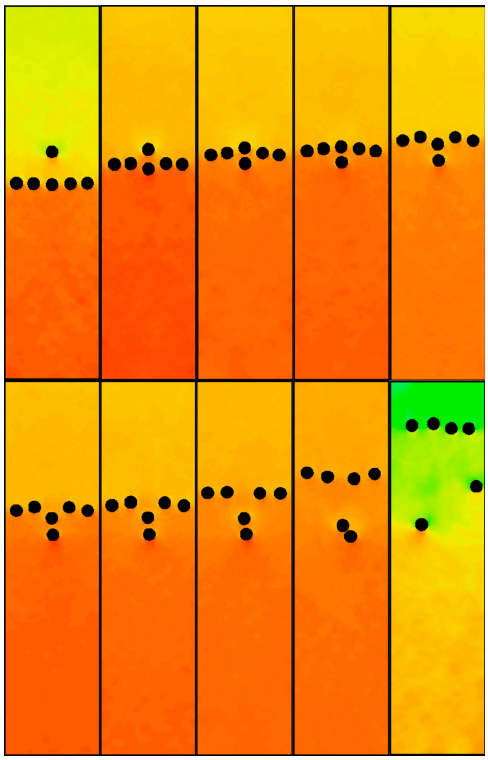


Fig. 21. A lone particle falling on a line of particles. The sequence proceeds in time from top to bottom and left to right. $\rho_s/\rho_f = 1.01$. The channel width is 8 particle diameters, and there are 20 grid spacings across each particle. The safety zone thickness $\delta = 3\Delta x$.

collision scheme will take a higher percentage of the processing time in the current explicit finite difference scheme since its time step is smaller and the collision scheme is called more frequently.

One hundred particles are shown sedimenting in a closed box in Figure 22. The sides of the box are 24.2 particle diameters long (a non-integer number of particle diameters is required because the particles must be separated by at least the safety zone thickness δ). The solid phase is twice as dense as the fluid. There are 20 grid spacings across each particle.

The memory requirements for the MacCormack scheme are easily estimated by counting the number of variables that must be stored at each grid point. On a per grid point basis then, one must store eight 64-bit numbers, or 64 bytes/grid point. The Particle-Mover FE scheme requires storage for the coefficient (stiffness) matrix and its preconditioner for the iterative solver, which comes to roughly 50 times the storage per node required by the MacCormack scheme. It is impossible to directly compare the two schemes, however, because Particle-Mover uses an adaptive (non-uniform) mesh but MacCormack is on a uniform mesh. For dense particulate systems, the FE mesh may need to be very fine everywhere, so the MacCormack memory requirements could be as much as two orders of magnitude lower than Particle-Movers in such sit-

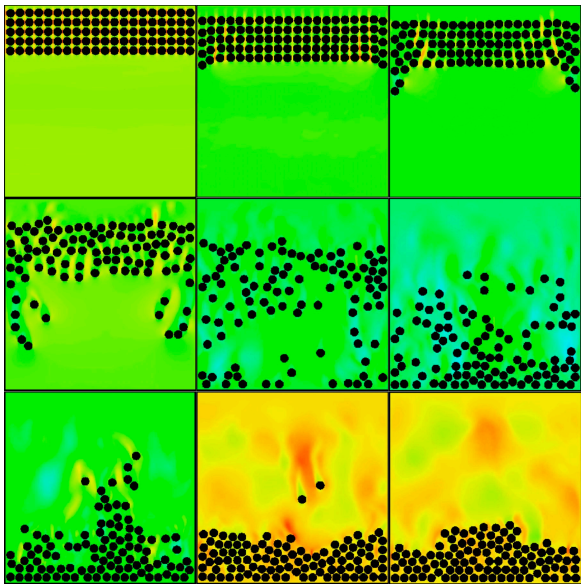


Fig. 22. One hundred particles sedimenting in a closed box. The time increases from left to right and top to bottom. $\rho_s/\rho_f = 2$. The box side length is 24.2 particle diameters, and there are 20 grid spacings across each particle. The safety zone thickness $\delta = 3\Delta x$.

uations. In situations involving only a few particles, Particle-Mover will likely have the advantage, especially if the domain dimensions are large compared to the particle diameter.

The processing issues for MacCormack are significant because of the small time steps required by the stability condition. Each time step can be computed very rapidly compared to a finite element calculation, but one must take a lot of them. The experience so far is that Particle-Mover is very much the faster of the two programs, although how much faster is sensitive to what is being simulated. However, MacCormack can be parallelized rather easily, while Particle-Mover would be more challenging.

7 Concluding Remarks

A second order explicit MacCormack finite difference scheme for moving particles has been described. The scheme solves the compressible Navier-Stokes equations on a uniform grid, and is quite efficient for simulation of concentrated suspensions in inertial flow regimes.

Some advantages include access to high Reynolds numbers, the ability to handle dense flows without running into memory constraints, ease of implementation, and the potential to be parallelized to improve performance.

There are also significant disadvantages. If the domain is large compared to

the particle size, and there are few particles, the fixed mesh will result in wasted time and processing. Finite elements are better suited to such cases. The scheme is not as efficient for flows with small Reynolds numbers (smaller than 1) due to the stability constraint.

Several examples were demonstrated, including a circular cylinder translating at constant speed, a single freely falling cylinder, two cylinders drafting, kissing, and tumbling, and one hundred particles sedimenting in a closed box.

8 Acknowledgements

A. Perrin acknowledges the support from the Graduate Assistance in Areas of National Need (GAANN) fellowship from the U.S. Department of Education.

References

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

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

[3] S. Chen and G. Doolen. Lattice boltzmann method for fluid flows. *Annu. Rev. Fluid Mech.*, 30:329–364, 1998.

[4] A. J. Chorin. A numerical method for solving incompressible viscous flow problems. *J. Comput. Phys.*, 2:12–26, 1967.

[5] F. Fortes, D.D. Joseph, and T.S. Lundgren. Non-linear mechanics of fluidization of beds of spherical particles. *J. Fluid Mech.*, 177:467–483, 1987.

[6] U. Ghia, K. N. Ghia, and C. T. Shin. High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method. *J. Comput. Phys*, 48:387–411, 1982.

[7] S. Hou and Q. Zou. Simulation of cavity flow by the lattice boltzmann method. *J. Comp. Phys.*, 118:329–347, 1995.

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

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

[10] H.H. Hu, N.A. Patankar, and M.Y. Zhu. Direct numerical simulations of fluid-solid systems using the arbitrary lagrangian-eulerian technique. *J. Comput. Phys.*, 169:427–462, 2001.

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

[12] P.K. Kundu and I.M. Cohen. *Fluid Mechanics*. Elsevier Academic Press, 3rd edition, 2004. pp. 686-687.

[13] A.J.C. Ladd. Short-time motion of colloidal particles: Numerical simulation via a fluctuating lattice-boltzmann equation. *Phys. Rev. Lett.*, 70(9):1339–1342, 1993.

[14] R. W. MacCormack. The effect of viscosity in hypervelocity impact cratering. Technical Report 69-354, AIAA, 1969.

[15] J. S. McNown, H. M. Lee, M. B. McPherson, and S. M. Ebgez. In *Proc. 7th Int. Congr. Appl. Mech.*, volume 2, page 17, 1948.

[16] R.R. Nourgaliev, T.N. Dinh, and T.G. Theofanous. A pseudocompressibility method for the numerical simulation of incompressible multifluid flows. *Intl. J. Multiphase Flow*, 30:901–937, 2004.

[17] R.R. Nourgaliev, T.N. Dinh, T.G. Theofanous, and D. Joseph. The lattice boltzmann equation method: theoretical interpretation, numerics and implications. *Intl. J. Multiphase Flow*, 29:117–169, 2003.

[18] D. Qi. Lattice-boltzmann simulations of particles in non-zero-reynolds-number flows. *J. Fluid Mech.*, 385:41–62, 1999.

[19] J.C. Tannehill, D.A. Anderson, and R.H. Pletcher. *Computational Fluid Mechanics and Heat Transfer*. Taylor & Francis, Washington, DC, 2nd. edition, 1997.

[20] H.S. Udaykumar, R. Mittal, and W. Shyy. Computation of solid-liquid phase fronts in the sharp interface limit on fixed grids. *J. Comput. Phys.*, 153:535–574, 1999.