JOURNAL OF COMPUTATIONAL PHYSICS **124,** 337–350 (1996)
ARTICLE NO. 0064

Numerical Considerations in Simulating a Turbulent Suspension of Finite-Volume Particles

SHIVSHANKAR SUNDARAM AND LANCE R. COLLINS*

Department of Chemical Engineering, Pennsylvania State University, University Park, Pennsylvania 16802-4400

Received March 17, 1995

Direct numerical simulation of a dilute suspension of *finite-volume* spheres requires computation of the time-varying fluid field and updating the particle momenta and positions, taking into account the effects due to particle–particle collisions. Collision calculations are inherently an order N_p^2 operation, where N_p is the number of particles in the system. Typical simulations contain 10^5–10^6 particles making the brute force computation of collisions prohibitively expensive. An alternative algorithm, based on molecular-dynamic-simulation strategies, is proposed in this paper. A second consideration in simulating a finite-volume particle suspension is how the particle forces should be coupled back into the fluid calculation (so-called reverse coupling). Careful consideration of the energy budget for the particle and fluid phases indicates that interpolation schemes for forward and reverse coupling must be symmetric in order to ensure proper behavior of the overall energy balance. Asymmetric interpolation schemes will lead to errors of one plus the error of the least accurate interpolation method per iteration. Of course, global errors may be much larger due to a cumulative effect of the systematic deviation.

1. INTRODUCTION

The emergence of direct numerical simulations as an alternative tool for investigating turbulent particle suspensions has been driven by the ever-increasing need for reliable simultaneous data on the particle and fluid phases in a particle suspension. Precise information on the two phases will enable investigations of subtle issues such as partitioning of energy transfer between phases [1], the mechanism of turbulence suppression in particle flows [2, 3], and particle collision dynamics [4], amongst many others. Such detailed information can only be found from numerical simulation because simulation yields the entire velocity field of the particle and fluid phases simultaneously, to a degree of precision limited only by the numerical method and grid resolution. The quantitative assessment of the simultaneous behavior of both velocity fields would be difficult, if not impossible, to achieve by conventional means (although it is noted that liquid-phase experiments in the near future shall be capable of providing similar information [5]). Hereafter, the term direct numerical simulation (DNS) refers to the numerical integration of the exact equations of motion for the fluid and particle phases with essentially no modeling assumptions.

For relatively dilute suspensions (i.e., volume fractions below 0.1%), there have been a number of investigations (see, for example, [6–9]) that neglect the finite volume of the particles, and rather treat the particles as point masses. In addition, we include the Lagrangian particle studies of Yeung and Pope [10] and more recently Yeung [11] in this category since the algorithms are essentially the same. The critical numerical question in these studies is the interpolation of the grid velocities to the particle positions. As shown by Yeung and Pope [10] and subsequently by Balachandar and Maxey [12], the need for a high-order interpolation (third order and higher) is essential for accurate representation of the particle positions, especially for describing the particle Lagrangian history and particle self-diffusion coefficient. Particle simulations of inhomogeneous turbulent flows have also been performed. For example, the boundary layer flow considered by Kallio and Reeks [13] and the channel studies by Kontomaris and Hanratty [14], McLaughlin [15], Brooke *et al.* [16], and Brooke *et al.* [17] neglect the particle volumes as well; although, their studies of deposition [16, 17] do assume a particle radius for the purpose of determining deposition rates. It is important to note that checking for wall collisions requires $O(N_p)$ operations (where N_p is the number of particles), whereas particle–particle collisions requires $O(N_p^2)$ operations, thus the computational effort for the deposition studies is relatively small.

A second category of studies of particle systems consisting of large particles at very high concentrations (i.e., volume fractions above 30%) is the so-called granular flow case. Here the relatively massive particles behave like a "gas phase" with a large number of particle–particle collisions. The interstitial gas, occupying such a small volume and having so little inertia, is often neglected in this regime. There have been a number of simulations of this system [18–23] in which the particle phases are tracked with an algorithm similar to those used in molecular dynamics sim-

*Author to whom inquiries should be addressed.

0021-9991/96 $18.00

ulations, augmented by a relatively sophisticated algorithm to describe the inelastic collision dynamics [24]. Indeed, these flows are essentially collision dominated; that is, collisions are very important because they are the principle way in which the solid phase loses energy. The system is somewhat simpler than the simulations performed on the dilute suspensions because the gas phase is ignored; however, the accounting of all particle–particle collisions is more sophisticated than the simple ''point-mass'' approximation.

Two earlier investigations by Sundaram and Collins [25, 26] provide motivation to perform simulations that include the effect of the gas phase *and* finite-volume particles. To summarize their results, Sundaram and Collins developed relationships for several two-point correlations in a particle–fluid system, each of which required simultaneous information about the position and velocity of the particles and fluid as an input. Two-point correlations are useful for characterizing the spatial arrangement of the particles and for incorporating the multitude of length scales present in turbulent flows in a self-consistent manner [27]. Their analysis demonstrated that the mathematically accurate method for characterizing two-point correlations or spectra of an inhomogeneous gas–solid suspension is to perform integrals over the particle, fluid, and mixed phases. They also showed that many spectral correlations can have an appreciable contribution from the particle phase, even though the suspension is ''dilute,'' because the density of the particles is much larger than that of the suspending fluid. Consequently, the most general and versatile spectral analysis of a suspension should include the contributions from the finite-volume spheres and the continuous fluid phase. This provides motivation for seeking an algorithm for simulating a fluid laden with finite-volume spheres.

In this paper, we consider the numerical issues associated with simulating *finite-volume* particles with a *participating* gas phase. Section 2 describes the governing equations and gives an overview of the numerical method used in the direct numerical simulations. We then consider two numerical consequences of finite volume spheres: (i) Finite-volume particles will collide; thus an algorithm for accounting for collisions between particles is required. (ii) The presence of a known particle volume has implications for the reverse coupling terms in the fluid equations. Each of these topics is discussed in detail in Sections 3 and 4 respectively. A summary and suggestions for future implementations is then given in Section 5.

2. OVERVIEW OF NUMERICAL SIMULATIONS

The technique of direct numerical simulation of particle-laden flows has been discussed extensively elsewhere in the literature [28, 29], therefore the purpose of this section is to summarize the governing equations and numerical method we employ. Our discussion will be limited to a periodic cube of fluid that contains N_p spherical particles, each of which may be of a different size. To achieve reasonably accurate statistics, N_p typically ranges between 10^5–10^6 particles. The particle trajectories are likewise periodic, so that particles that exit one side of the box appear instantaneously on the opposite side. The numerical algorithm updates the positions and momenta of the individual particles in a Lagrangian sense, and the fluid phase by solving the incompressible Navier–Stokes equations shown below.

A. *Governing Equations*

The governing equations for determining the nth particle position and momentum in time are respectively

$$\frac{d\mathbf{x}_p^n}{dt} = \mathbf{v}_p^n \tag{1}$$

$$m_p^n \frac{d\mathbf{v}_p^n}{dt} = m_p^n \frac{[\mathbf{u}(\mathbf{x}_p^n) - \mathbf{v}_p^n]}{\tau_p^n} + \sum_{\substack{j=1 \\ j\neq n}}^{N_p} \mathbf{F}_p^{jn}, \tag{2}$$

where $\mathbf{x}_p^n$, $\mathbf{v}_p^n$, and m_p^n are center location, velocity, and mass of the nth particle. $\mathbf{u}(\mathbf{x}_p^n)$ is the velocity of fluid ''in the neighborhood'' of the nth particle (note, technically there cannot be fluid at the center of a particle, however, in practice the fluid velocity varies slowly over distances on the order of the particle radius; hence $\mathbf{u}(\mathbf{x}_p^n)$ is a good approximation for the local fluid velocity in the neighborhood of a particle). τ_p^n is the response time of the particle given by $\tau_p^n \equiv \rho_p \sigma_n^2/(18\mu)$, where σ_n is the diameter of the nth particle, ρ_p is the density of the particles, and μ is the molecular viscosity of the fluid. $\mathbf{F}_p^{jn}$ is the impulsive force that results from particle–particle collisions. The first term on the right-hand side of Eq. (2) is the leading force term due to Stokes drag, and the second accounts for collisions. (The force has been simplified to include the Stokes drag only; however, this has no bearing on the analysis presented herein. For the complete expression, see Maxey and Riley [30].)

Assuming the particle sizes are small as compared to the length scales of the turbulence and, furthermore, that the volume fraction of particles is relatively small (i.e., less than 0.1%), the equations of motion for the fluid can be approximated by

$$\nabla \cdot \mathbf{u} = 0 \tag{3}$$

$$\rho\left[\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u})\right] + \nabla p = \mu \nabla^2 \mathbf{u} - \sum_{n=l}^{N_p} \frac{m_p^n[\mathbf{u}(\mathbf{x}) - \mathbf{v}_p^n]}{\tau_p^n} \delta(\mathbf{x} - \mathbf{x}_p^n), \tag{4}$$

where ρ is the density of the fluid (assumed constant), μ is the molecular viscosity, $\mathbf{u}$ is the fluid velocity at a point $\mathbf{x}$, p is the pressure, and δ is the three-dimensional Dirac delta function. Equations (3) and (4) are approximate in that they neglect the volume that the particle phase occupies (i.e., dilute approximation). At this point, it is convenient to draw attention to an important parameter, the particle Stokes number defined as $St \equiv \varepsilon^{1/2}\tau_p/\nu^{1/2}$, where ε is the turbulent energy dissipation rate and ν is the kinematic viscosity of the fluid. Physically, the particle Stokes number is the ratio of particle response time to the Kolmogorov time scale.

B. *Numerical Approximation*

The particle positions and momenta are updated using a second-order Adams–Bashforth procedure. Because the fluid velocity is known only at discrete vertex points, the fluid velocity must be interpolated to the particle positions. Earlier investigations have compared several interpolation schemes [10, 12, 14] and shown that third-order accuracy is required at a minimum to accurately track the particle trajectories. In the present simulations, a cubic Lagrangian interpolation scheme is used. A novel feature of the present simulations is the incorporation of particle–particle collisions. In principle, for a hard-sphere system, particle collisions occur at discrete instants in time, thus the particle position algorithm must sort out the effects of all collisions that occur over each finite time interval Δt and enact the proper collision rules. Section 3 discusses in detail the strategies used to detect collisions and to carry out the collision dynamics.

The fluid velocity is updated in two separate steps. The first updates the convection, diffusion, and pressure terms using a pseudospectral algorithm similar to the one described in Canuto *et al.* [31]. Partial de-aliasing is accomplished by zeroing wavenumbers beyond 8/9 of $k_{\max}$, as originally suggested by Patterson and Orszag [32]. The particle force (so-called reverse coupling) terms are then evaluated. The numerical details associated with estimating the reverse-coupling terms at the vertex points is discussed in greater detail in Section 4. Time integration for the entire system of equations is done by an efficient fourth-order Runge Kutta routine.

3. PARTICLE–PARTICLE INTERACTIONS

There are numerous algorithms for describing particle collisions in so-called molecular dynamics simulations that may be useful in the present context. However, in a particulate turbulent flow, the problem of determining particle trajectories and, in particular, particle collisions is dependent not only on particle–particle interactions but also on time and position varying fluid turbulence. This illustrates the fundamental difference between particulate turbulence and molecular simulations of hard spheres. In turbulent flows, particles are advected by the time-varying flow field and, therefore, particle interactions cannot be projected forward in time indefinitely. This implies that collisions cannot be predicted over time intervals larger than the particle update time. While the present work borrows extensively from ideas prevalent in molecular dynamics literature [33–35], this difficulty requires us to tailor their techniques to meet our specific needs. In the following section, for the sake of completeness, we discuss core concepts of existing algorithms, while highlighting, wherever necessary, unique aspects demanded by the nature of turbulence.

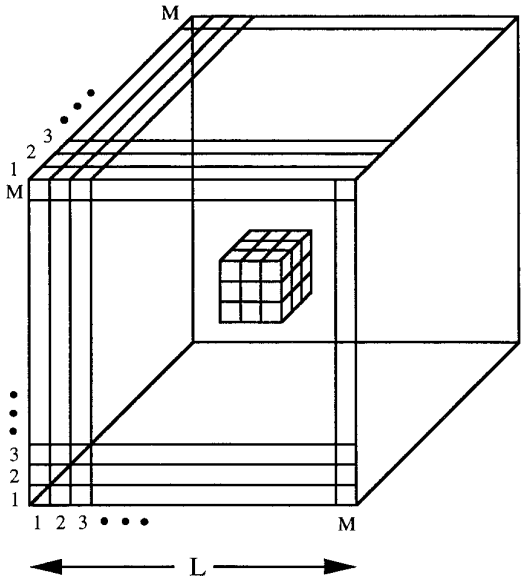

FIG. 1. Division of computational domain into M^3 cells. Potential collision partners for a particle in a given cell are found in the neighborhood comprising that cell and the 26 cells that surround the cell of interest, as shown in the center of the diagram.

Recall that we make the assumption that the particles behave as hard spheres. The term ''particle interaction'' in this system is, therefore, synonymous with hard-sphere ''collision'' while ''dynamic partners'' refers to the ''colliding particles.'' The simplicity of the hard-sphere potential renders it extremely attractive in view of the computational complexity of the other aspects of particulate turbulence. Prescribing a hard-sphere potential between particles is thus treated as a first step in including particle interactions. Comments on certain other potentials of interest and their implementation are deferred until the end of this section.

The particle motion (accurate to leading order) is described by Eqs. (1) and (2) shown in Section 2. The particle update can be broadly divided into four segments:

(i) Calculate the net force of fluid on each particle and update the particular momentum.

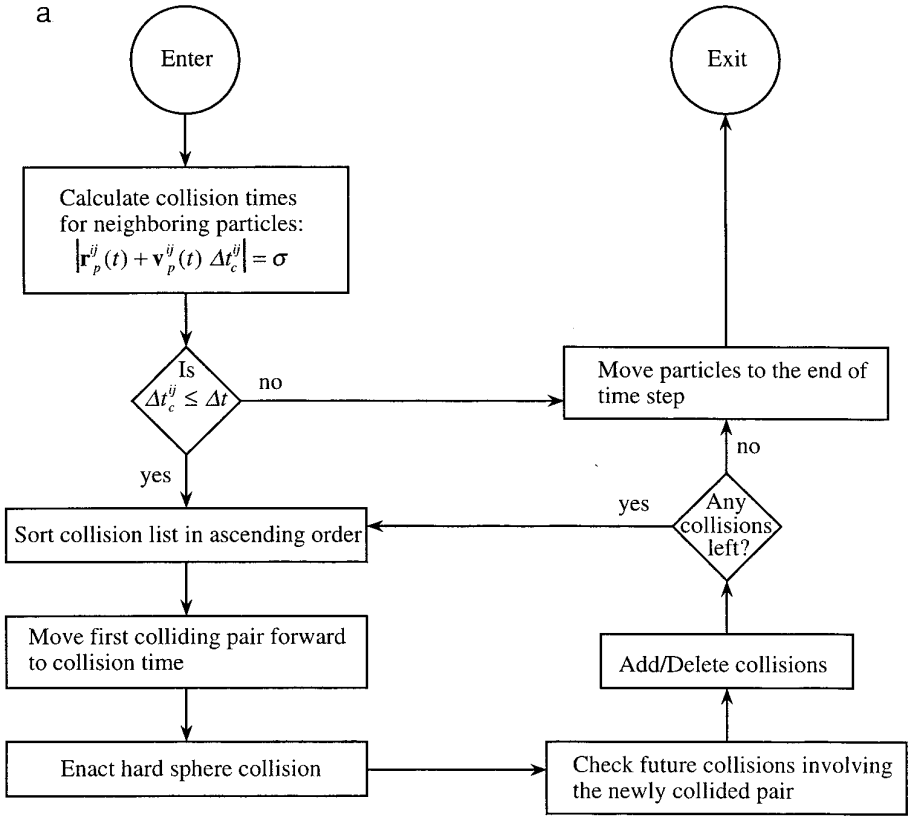

FIG. 2. (a) Algorithm for a proactive, time sequenced collision detection and enactment procedure. Note that the decision point upon entry into the module requires computation of collision times for all neighborhood particle pairs. (b) Algorithm for a retroactive, time-sequenced collision detection and enactment procedure. Note that the decision point upon entry into the module requires only a logical check for all neighborhood particle pairs.

(ii) Identify particles that will collide (or have collided) in a time interval Δt.

(iii) Enact elastic collisions between identified particles in the order they occur.

(iv) Update the particle positions using Eq. (1).

The force on a given particle (Stokes drag) depends on the interpolated fluid velocity and involves $O(N_p)$ operations, not considering the mechanics of the interpolation itself. Although this computation could rapidly escalate, based on the complexity of the interpolation, the total computational effort will still grow only linearly with N_p. For further discussion on the subject of interpolation we refer the reader to the following studies [10, 12, 14]. Implementation of impact dynamics, while certainly dependent on the nature of the particle interactions, is invoked only for the restricted subset of colliding particles and, hence, is not of concern in the computational budget. Identification of particle collisions, on the other hand, requires examining $\frac{1}{2} N_p(N_p - 1)$ particle pairs and, therefore, involves $O(N_p^2)$ operations. Thus, considering that we typically track 10^5–10^6 particles, it is clear that step (ii) is the computational "rate limiting step." We will, consequently, focus on algorithms to minimize the computational effort spent on identifying dynamic partners from among all particle pairs.

A. *Neighbor Lists*

Keeping in mind that it is terribly inefficient to have to go through the entire list of particles in order to determine the dynamic partners of a given particle, the notion of neighbor lists is introduced. Particle j is defined as a neighbor of particle i (and vice versa) if the pair separation distance r_p^{ij} is less than a defined length, R (where $r_p^{ij} \equiv |\mathbf{r}_p^{ij}|$ and $\mathbf{r}_p^{ij} \equiv \mathbf{x}_p^i - \mathbf{x}_p^j$). In the original concept [36], all of the neighbors of each particle are stored contiguously in an array which constitutes the neighbor list. The particle interaction calculation for a single particle, then, loops over only the restricted set of neighbors of that particle, as specified in the list. The list itself has to be updated every few time steps. A quick idea of the size of the neighbor

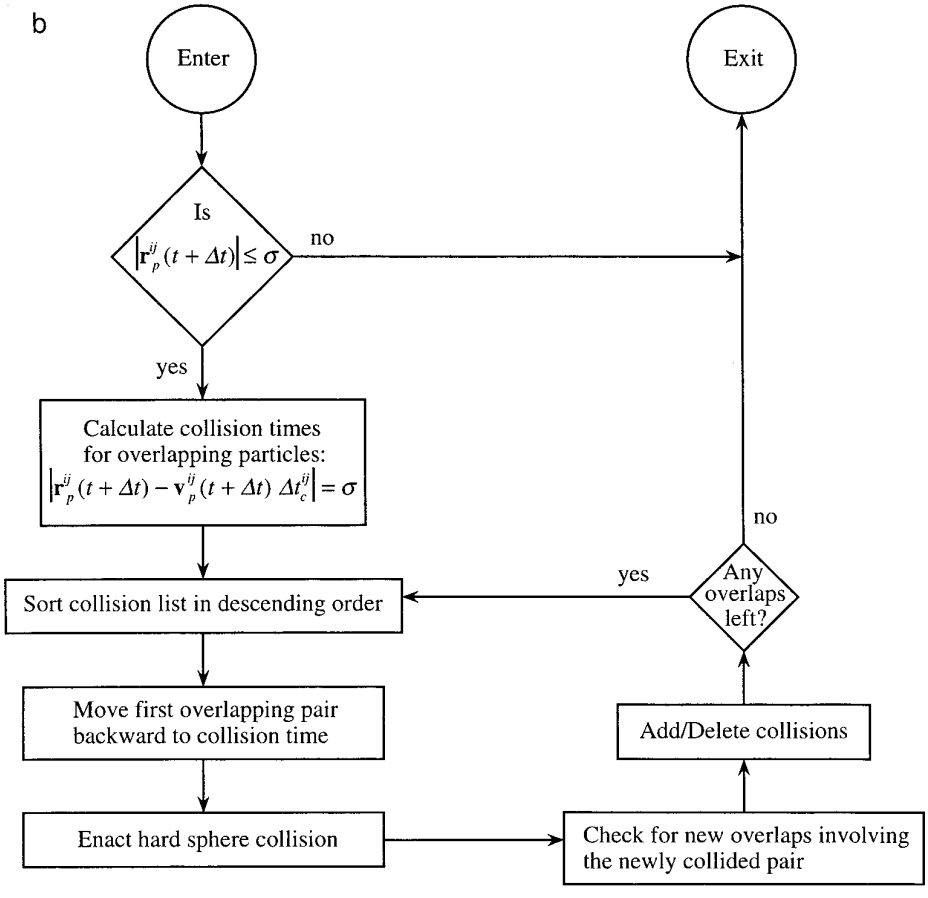

FIG. 2—*Continued*

list can be gained from the following consideration; the number of particles in the neighborhood volume is on average $\frac{4}{3}\pi R^3(N_p/L^3)$.

Summing over all the particles and avoiding double counting the same pair, the total number of elements in the neighbor list would be, approximately $\frac{2}{3}\pi R^3(N_p^2/L^3)$. The reduction in computational effort is therefore given by $\frac{4}{3}\pi(R^3/L^3)$. In principle, the smaller the value of R/L the greater the savings; however, R/L cannot be reduced indefinitely because eventually particles may interact with others that originated *outside* of the neighborhood. Such interactions would be missed by the proposed algorithm. Thus, an optimal value of R/L exists that minimizes the computational effort while not introducing significant errors due to unaccounted for collisions. We will return to this issue in the next subsection.

This illustrates the savings in compute time found by examining a reduced subset of possible dynamic partners. However, the above method is excessively memory intensive for calculations involving moderate to large numbers of particles. An alternative method that uses discrete cells in the computational domain has been found to be more efficient.

B. *Cell Index Method*

The cell-index method [34, 37], while incorporating the concept of neighbor lists, presents a more efficient way of organizing the updating neighbor information. The cubic simulation box is divided into a regular lattice of M^3 cells (see Fig. 1). The potential dynamic partners of a given particle in any cell are found from the other particles in the cell plus the particles in all the neighboring cells. This is illustrated in Fig. 1 by the central cube highlighting the 27 neighboring cells. Considering that there are on average N_p/M^3 particles in each cell and 27 neighboring cells to be searched per particle, this procedure (avoiding double counting) requires $O(\frac{27}{2}N_p^2/M^3)$ operations. The list of particles in each cell is set up and updated using linked lists. While there is some initial overhead in setting up the cell structure, the sorting of particles is fairly rapid ($O(N_p)$). Additonal details can be found in the excellent text by Allen and Tilldesley [33]. As noted above, M cannot be increased indefinitely, owing to particle dynamics considerations. Since we seek to identify colliding particle pairs only from the particles in the present cell and neighboring cells, this method is accurate only in the limit that $\nu_p^{\max}\Delta t \ll L/M$; i.e., the acceptable range of M is given by

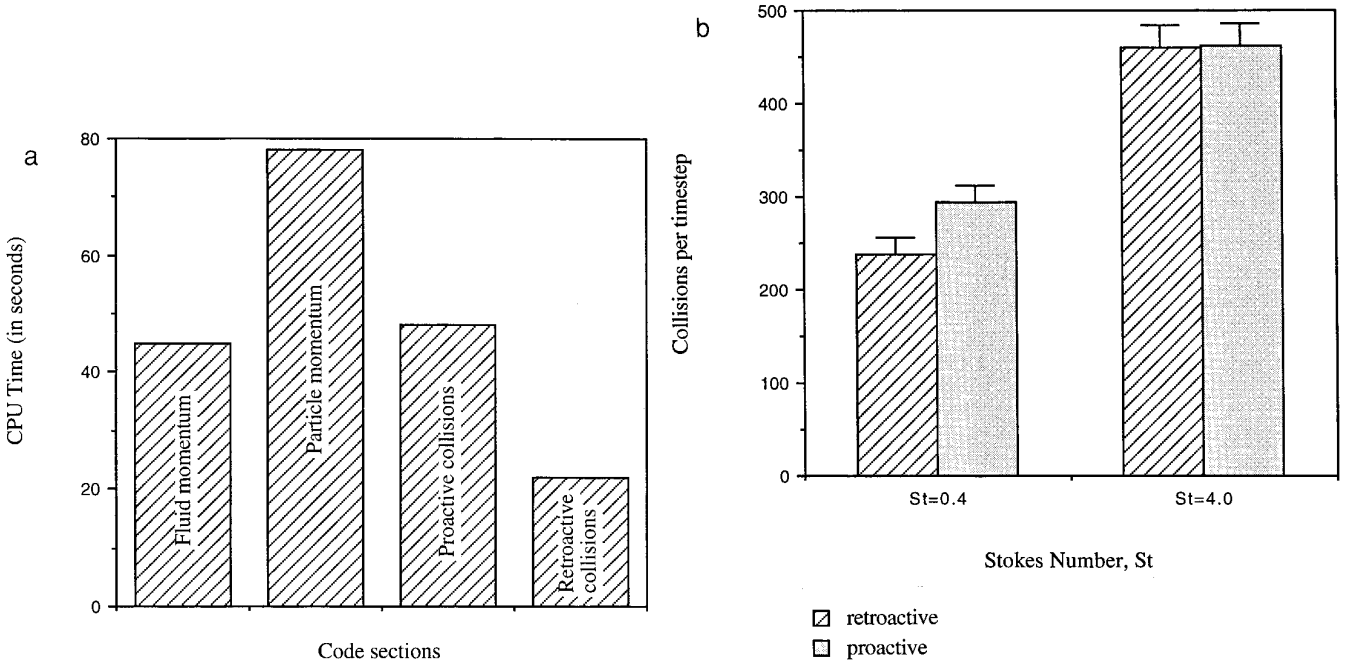

FIG. 3. (a) Comparison of CPU timings for a numerical simulation with a 64^3 fluid grid points and 64^3 particles. Both collision modules require CPU times that are of the same order as the particle and fluid momentum updates. However, the proactive method of collision enactment is twice as computationally intensive as the retroactive method. (b) Comparison of collisions per timestep obtained from both proactive and retroactive procedures. For $St = 4.0$, both methods yield almost identical results. However at $St = 0.4$, the retroactive procedure significantly underpredicts the collision frequency (by about 20%) indicating the presence of interpenetration and multiple collisions within a timestep.

$$1 \leq M \ll L/\nu_p^{\max}\,\Delta t. \tag{5}$$

C. *Identifying Collisions*

For a hard sphere system, we can check for dynamic interactions between particles i.e., collisions, within the timestep Δt, in two ways:

(a) *Proactive method.* The algorithm is schematically outlined in Fig. 2a. We calculate the collision time, Δt_c^{ij}, for all the neighbor pairs ($O(\frac{27}{2} N_p^2/M^3)$) by solving the quadratic equation

$$|\mathbf{r}_p^{ij}(t + \Delta t_c^{ij})| = |\mathbf{r}_p^{ij}(t) + \mathbf{v}_p^{ij}\Delta t_c^{ij}| = (\sigma_i + \sigma_j)/2, \tag{6}$$

where

$$\mathbf{v}_p^{ij} \equiv \mathbf{v}_p^i - \mathbf{v}_p^j.$$

For particles that are not on a collision course, the solution for Δt_c^{ij} found from Eq. (6) will be either complex or negative. In either case, the value is rejected and the particle pair is eliminated from the list of potential colliding partners. Most particle pairs will fall into this category. The quadratic solution for the remaining pairs will yield two positive values, the lesser of which constitutes the time-to-collision. A collision is scheduled for the pair within a timestep Δt if $0 \leq \Delta t_c^{ij} \leq \Delta t$ (where again Δt_c^{ij} refers to the minimum positive value). The actual enactment of collisions involves the following steps. The scheduled collisions are sorted in ascending order. The pair with the smallest Δt_c^{ij} is collided first, and the clock is advanced to the time $t + \Delta t_c^{ij}$. Note that each collision can potentially affect future collisions involving either of the two particles, thereby creating new, unscheduled collisions or eliminating others that were previously scheduled. Hence we redo the collision time computation for the two newly collided particles with all the other particles in the neighborhood. New collisions (within the same timestep) are added to the schedule list and/or old ones that can no longer occur are deleted. The schedule list is resorted in ascending order and the implementation begins again with the next smallest collision time. This continues until there are no more collisions to be implemented in the schedule list. In principle, the particle collisions could be enacted in a random order (or the order in which they were scheduled), which would eliminate the need for sorting and recalculation; however, such a procedure may yield erroneous results in the case of multiple collisions of a particle within the same timestep.

(b) *Retroactive method.* Alternatively, we can advance the particles to the end of the present timestep and then

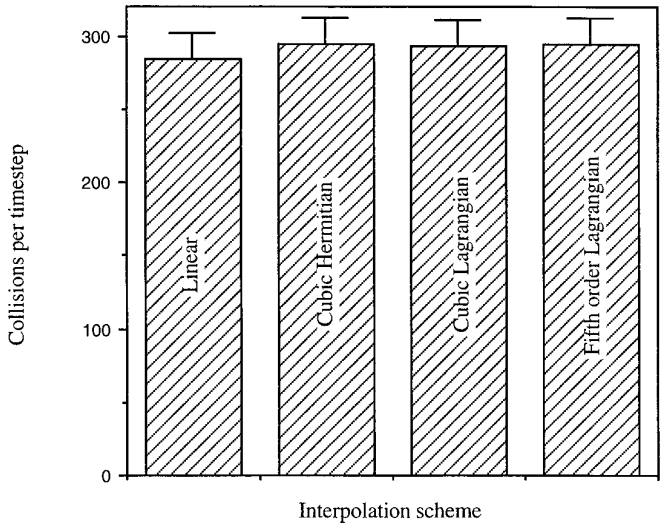

FIG. 4. Influence of fluid velocity interpolation scheme on the collision frequency obtained from the simulation. All schemes used are in agreement within error bounds.

locate collisions through the presence of overlap. This involves a logical test for each of the neighboring pairs. Particle collisions have occurred when

$$r_p^{ij}(t + \Delta t) \leq (\sigma_i + \sigma_j)/2. \tag{7}$$

Once the overlapping particles are identified the following steps are carried out. First, the exact time of collision for each colliding pair is determined from the following expression:

$$|\mathbf{r}_p^{ij}(t + \Delta t) - \mathbf{v}_p^{ij}\Delta t_c^{ij}| = (\sigma_i + \sigma_j)/2. \tag{8}$$

The scheduled collisions are then implemented in either a time-sequenced manner (*descending* order of the calculated collision times) or a random order, under the same penalty of erroneous treatment of multiple collisions discussed above. In the time-sequenced implementation, the particles are moved backward in time to the point of collision, the elastic collision between the pair is affected, and the particles are again moved forward in time. Overlap computations are redone for collided particles. New overlaps are added to the schedule list while non-existent ones are deleted, after which the entire list is resorted in descending order. These instructions are carried out until there are no remaining overlaps to consider. A schematic of the algorithm is shown in Fig. 2b.

The advantage of the retroactive test is that it has a relatively simple test for determining when a collision has occurred (Eq. (7)). Thus, the quadratic formula (Eq. (8)) is only applied to the *subset* of particle pairs that have collided. Unfortunately, this method suffers from limitation that it will not detect a collision if complete interpenetration occurs within a particular timestep. One can treat this as an external limitation on the particle timestep. Normally the particle timestep in particle-laden DNS is fixed by the fluid calculation considerations. However, if we are to minimize the occurrences of complete interpenetration we have to maintain $\nu_p^{\max}\Delta t/\sigma \ll 1$ (a particle "Courant number"). It should be pointed out that the Courant number is dependent on the particle Stokes number. Particles with low Stokes numbers, i.e., lighter, more energetic particles, will have a higher Courant number.

To obtain an objective time and precision comparison between the two methods, we performed direct numerical simulations using: (a) proactive strategy with sorting (best case) and (b) retroactive strategy without sorting (worst case). The simulations used 64^3 fluid grid points and 64^3 particles with Stokes numbers (Courant numbers) that were either 0.4 (0.6) or 4.0 (0.4). Simulations were carried out on an IBM RS 6000 Model 370 workstation. The statistical frequency of collision between particles was chosen to compare the two strategies. The results are summarized in Figure 3.

It can be seen from Fig. 3a that while both methods of collision implementation are as expensive as other aspects of the calculation, the proactive method required a factor of two times more CPU time than the retroactive method. However, as seen from Fig. 3b, the difference in collision frequencies obtained by the two methods for the case of $St = 0.4$ is quite significant (~20%). For the heavier particles $St = 4.0$, both strategies yield almost identical collision frequencies. Recall that any difference in the results from these two methods is directly attributable to multiple collisions and complete particle interpenetrations. We therefore conclude that particle interpenetrations and multiple collisions account for a significant percentage of all collisions at low Stokes numbers, even though we are in the dilute limit. Results obtained from a retroactive collision counting strategy become increasingly erroneous as the Stokes number is lowered. Obviously, the situation will worsen as the particle concentration increases.

D. *Influence of Interpolation Scheme on Collisions*

Given a precise representation of the fluid flow field, the accuracy of computed particle trajectories is dictated by the interpolation scheme used to calculate fluid velocities at the various particle locations. As mentioned earlier, many investigations have been carried out in the past to determine the computational cost and accuracy of different interpolation schemes [10, 12, 14, 38]. It is generally agreed that linear interpolation of velocities is inadequate and schemes of order three or higher are required for reliable particle trajectories. One can formulate these higher order

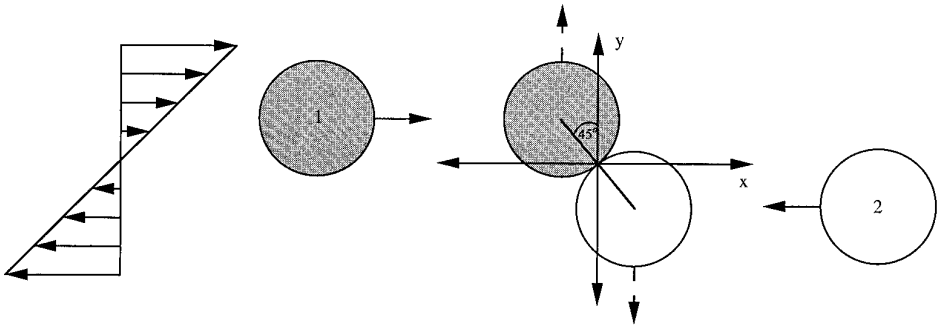

FIG. 5. Schematic of two spherical particles in a pure shear flow coming in contact at an angle of 45° to the velocity gradient.

schemes using either more fluid grid velocities (Lagrangian schemes) or velocity derivatives at the nearest grid points (Hermitian schemes), taking advantage of the spectral representation of the fluid flow field. This problem of velocity interpolation is especially relevant in Lagrangian studies of diffusion of fluid or other particles [38–40]. However, as it pertains to the present study, it is observed that the interpolation scheme does not appear to play a significant role in determining the collision frequency. This is apparent in Fig. 4 which shows the collision frequency for several different interpolation schemes (linear, cubic Lagrangian, cubic Hermitian, and fifth-order Lagrangian). Note the agreement within calculated error bounds. The choice of an interpolation scheme therefore should be determined from other considerations, some of which are discussed in Section 4.

E. *Multiple Collisions of a Single Pair*

Another important consideration in the dynamics of colliding particles as compared to point masses is the question of multiple collisions between a single pair of particles. It has been observed that particle pairs with rapid response times can collide several times during a single "interaction." This is an apparent artifact of the imprecise dynamics assumed for the particle pair during collision; that is, the microscopic model used to estimate the force on a particle does not take into account the presence of the second particle. Consequently, what would likely be a single collision physically will manifest itself as multiple collisions in the simulation under certain circumstances. Multiple collisions can be considered a crude approximation to the true two-particle dynamics. Fortunately this artifact is insignificant with respect to overall particle dynamics because the fraction of particles that are colliding at any given time is very small. However, caution must be used in determining *collision frequencies* from simulation data because multiple collisions may produce spuriously large collision frequencies.

To demonstrate the multiple collision effect, we consider two spheres moving relative to each other as a result of a uniform velocity gradient of magnitude $\dot{\gamma}$ (see Fig. 5). Without loss of generality, we take the particles to be moving in the x-direction, with particle one moving in the positive x-direction and particle two in the negative x-direction. Based on these assumptions the non-dimensional equations of motion for sphere "m" ("m" refers to either particle 1 or 2—see Fig. 5) are given by

$$\frac{d\nu_{p_x}^m}{dt} + \frac{\nu_{p_x}^m}{St^*} = \frac{y_p^m}{St^*}, \quad \frac{dx_p^m}{dt} = \nu_{p_x}^m, \tag{9}$$

$$\frac{d\nu_{p_y}^m}{dt} + \frac{\nu_{p_x}^m}{St^*} = 0, \quad \frac{dy_p^m}{dt} = \nu_{p_y}^m, \tag{10}$$

where position, time, and velocity are made dimensionless, based on the particle diameter σ, the characteristic flow time $1/\dot{\gamma}$, and characteristic approach velocity $\dot{\gamma}\sigma$, respectively. For the purpose of this discussion, the particle Stokes number is defined as $St^* \equiv \dot{\gamma}\tau_p$. Furthermore, for the sake of simplicity, we assume the particles hit at an angle of incidence of 45° to the perpendicular (see Fig. 5). Then, the initial conditions *immediately following* the collision are

$$\begin{aligned} t = 0^+: \quad \nu_{p_x}^m = 0, \quad \nu_{p_y}^m = \frac{(-1)^{m-1}}{2\sqrt{2}}, \\ x_p^m = \frac{(-1)^m}{2\sqrt{2}}, \quad y_p^m = \frac{(-1)^{m-1}}{2\sqrt{2}}. \end{aligned} \tag{11}$$

The exact solution to Eqs. (9) and (10) with the above initial conditions is

$$\nu_{p_x}^m = \frac{(-1)^{m-1}}{2\sqrt{2}}\left[(1 + St^*)(1 - e^{-t/St^*}) - te^{-t/St^*}\right] \tag{12}$$

$$\nu_{p_x}^m = \frac{(-1)^{m-1}}{2\sqrt{2}}\, e^{-t/St^*} \tag{13}$$

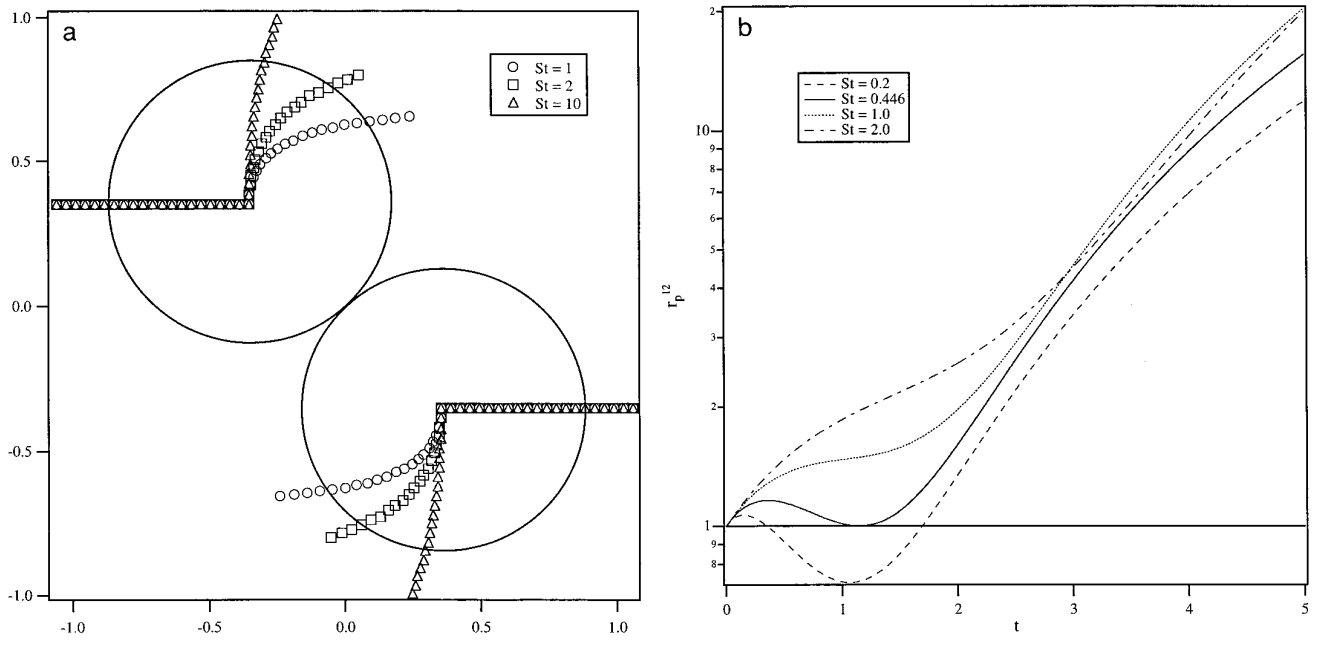

FIG. 6. (a) Particle center trajectories for various values of St^*. Notice that with increasing St^*, particles travel greater distances before acquiring appreciable momentum from the fluid. (b) Variation of the particle separation with time for various St^*. Note that for $St^* \leq 0.45$ the particle separation crosses unity indicating that particles will undergo secondary collisions.

$$x_p^m = \frac{(-1)^{m-1}}{2\sqrt{2}} St^* \left[\frac{t-1}{St^*} + (1+2St^*)(e^{-t/St^*}-1) + t(e^{-t/St^*}+1) \right] \tag{14}$$

$$y_p^m = \frac{(-1)^{m-1}}{2\sqrt{2}} [1 + St^*(1-e^{-t/St^*})]. \tag{15}$$

Particle trajectories, as depicted by the motion of the particle centers, are shown in Fig. 6a for several values of the parameter St^*. Immediately following the collision event ($t = 0$), the particle velocities are pointed in the y direction, however, in time, they are deflected by the external flow field. The rate of re-alignment of the particle velocity with the flow field is controlled by the parameter St^*. For large St^*, the ''persistence'' of the y-velocity is long and the particle easily escapes without a second collision. For smaller St^*, the particles will more quickly regain x-momentum from the flow field, thus creating the possibility for a second (third, fourth) collision. This can be seen more easily if we consider the magnitude of the relative coordinate vector, r_p^{12} defined by

$$\begin{aligned} r_p^{12} &= [(x_p^1 - x_p^2)^2 + (y_p^1 - y_p^2)^2]^{1/2} \\ &= [(2x_p^m)^2 + (2y_p^m)^2]^{1/2}. \end{aligned} \tag{16}$$

Recall that a collision has occurred whenever $r_p^{12} = 1$. A plot of r_p^{12} versus time is shown in Fig. 6b for several values of the parameter St^*. As the particle Stokes number decreases, the particles pass each other by an increasingly narrow margin, colliding for a second time for $St^* \leq 0.45$.

Once again, the presence of multiple collisions is not very significant in terms of the overall dynamics of the particles and fluid because at any given time there are only a small number of particles that are colliding as compared to non-colliding particles. Nevertheless, this effect is important when counting collisions because it can lead to spuriously high collision frequencies, particularly at low Stokes numbers. Indeed, in a companion study of collision frequencies in particle laden flows [4], it has been observed that reliable results could not be obtained for particles with $St < 0.4$. (Recall, for fully developed turbulence, the Stokes number is usually defined in terms of the Kolmogorov eddy time scale, i.e., $St = \varepsilon^{1/2}\tau_p/\nu^{1/2}$.) This is attributed to the appearance of multiple collisions at the lower values of the Stokes number. Remarkably, the theory outlined above captures the critical value of the Stokes number reasonably well ($St = 0.4$), despite the simplified flow field considered. Apparently, the *local* flow field around colliding particles in fully developed turbulence is effectively represented by a linear shear flow, as originally suggested by Saffman and Turner [41]. More realistic collision rates for particles with Stokes numbers in the range $0 \leq St \leq$

0.4 can be obtained by either substituting more precise collision dynamics that include the effect of the second particle on the flow field, or by eliminating multiple collisions from the counting scheme.

F. *Generalization to Other Potentials*

The main feature in this particle calculation was the idea of determining dynamic partners from a restricted subset of particles, i.e., the neighbor lists. More complicated potentials, in principle, may include interactions amongst all particles in the system. We can reconcile the two ideas, if the potential under consideration is short-range and therefore can be truncated at a finite radius, r_c, smaller than the computational domain, i.e. ($r_c < L$). For example, a short-range repulsive force due to a lubrication layer between the particles may be necessary for describing more viscous flows. A second example is the possible presence of electrostatic charges on the particles that may cause a longer-range repulsive force (typically particles will be charged negatively), as can hydrodynamic forces for larger particles [25, 42]. For the computation of dynamic partners to be accurate we simply require that the neighborhood scale is larger than r_c ($L/M > r_c$). Recognizing the numerical advantages of small neighborhoods, it is apparent that the shorter the range of the postulated potential, the numerically faster the calculation will be. Since the hard sphere potential represents the shortest range of interaction possible ($r_c = \sigma$), the results obtained in our study can be interpreted as a best case analysis.

4. TWO-WAY COUPLING

Experimental studies report both augmentation and attenuation of the fluid phase turbulence due to the presence of particles [43–46]. Augmentation of fluid turbulence seems contrary to conventional wisdom, which indicates that the particles are an additional source of dissipation. However, when one considers the *total* energy in the system (fluid and dispersed phases), it rapidly becomes clear that the particles are indeed dissipative, and modulation of the turbulence depends on aspects of spectral transfer due to the particle source term in the fluid equation (see Eq. (4)). Nevertheless, several critical questions regarding the precise nature of energy interchange between the fluid and particle phases remain unanswered, and perhaps numerical simulation will provide an important, if not the only, means for obtaining that information.

In the past, numerous numerical studies have attempted to describe the phenomenon of turbulence modification by a dispersed phase due to so-called reverse coupling [7, 8]. Since the particles in these simulations were ''point masses,'' there was no basis for estimating the disturbance velocity from the particles, thus ad hoc approximations for the reverse coupling terms were used. In the present study, the presence of a *finite-volume* particle will produce a disturbance velocity in the fluid that in principle can be determined from the Navier–Stokes equations [47]. Unfortunately, the exact solution for the disturbance velocity at arbitrary Reynolds number is presently unknown. Furthermore, there remains the difficulty of reverse-interpolating the disturbance velocity resulting from the particles back to the fluid grid points. That is, there are important issues of conservation and discretization that must be resolved to make numerical simulations of finite-volume particles with two-way coupling precise. In this section, we develop conservation laws within the framework of a discrete simulation. Furthermore, we address issues such as the effect of interpolations and reverse interpolations on the mechanism of energy transfer between the phases. The analysis is similar in nature to the one proposed by Burgess *et al.* [48] in a different context.

A. *Exact Equations of Kinetic Energy Transfer*

The governing equations for the fluid and particle phases have been summarized in Section 2 (Eqs. (1)–(4)). If we define the kinetic energy per unit volume of fluid as $e \equiv \frac{1}{2}\rho u_i^2$, then the governing equation for e is

$$\frac{\partial e}{\partial t} + \nabla \cdot [(e+p)\mathbf{u}] = \nu \nabla^2 e - \varepsilon - \sum_{n=1}^{N} \frac{m_p^n \mathbf{u}(\mathbf{x}) \cdot [\mathbf{u}(\mathbf{x}) - \mathbf{v}_p^n(\mathbf{x}_p^n)]}{\tau_p^n} \delta(\mathbf{x} - \mathbf{x}_p^n), \tag{17}$$

where ε is the local rate of dissipation of energy (per unit volume) defined by

$$\varepsilon = \mu \nabla \mathbf{u} : \nabla \mathbf{u}; \tag{18}$$

Eq. (17) is an exact expression for the kinetic energy of the fluid phase. If we integrate the result over the volume of fluid in the periodic box we obtain the dynamic equation for the total kinetic energy in the fluid phase. The result is

$$\frac{dE}{dt} = -\Phi_\nu - \sum_{n=1}^{N} \frac{m_p^n \mathbf{u}(\mathbf{x}_p^n) \cdot [\mathbf{u}(\mathbf{x}_p^n) - \mathbf{v}_p^n(\mathbf{x}_p^n)]}{\tau_p^n} + \iint_S [\nu \nabla e - (e+p)\mathbf{u}] \cdot \mathbf{n}\, dS, \tag{19}$$

where

$$E \equiv \iiint_V e\, dV \tag{20}$$

$$\Phi_\nu \equiv \iiint_V \varepsilon\, dV. \tag{21}$$

The surface integral shown in Eq. (19) accounts for contributions to the kinetic energy due to pressure work at the boundary and external fluxes across the boundary of the box, neither of which contributes in the periodic system under consideration. Thus, the surface integral is identically zero, and the total kinetic energy of the fluid phase is affected by only two terms: (i) a viscous loss term which accounts for degradation of kinetic energy into thermal energy; (ii) an energy exchange term which accounts for gains or losses to the particle phase.

It is informative to conduct a similar analysis on the particle phase so that a total energy balance can be constructed. The dynamic equation for the kinetic energy of a single particle is found by dotting Eq. (2) with the particle velocity. Summing over the total number of particles then yields

$$\frac{dE_p}{dt} = \sum_{n=1}^{N} \frac{m_p^n \mathbf{v}_p^n(\mathbf{x}_p^n) \cdot [\mathbf{u}(\mathbf{x}_p^n) - \mathbf{v}_p^n(\mathbf{x}_p^n)]}{\tau_p^n}, \tag{22}$$

where E_p is the total kinetic energy of the particle phase. (Since collisions are assumed to be elastic, the collision impulse force, $\mathbf{F}_{jn}$, conserves the kinetic energy of the particles, and therefore does not contribute to Eq. (22).) It is immediately apparent from Eq. (22) that the energy exchange between the two phases is *not conservative* because the ''weighting'' of the velocity difference in the two expressions (Eqs. (19) and (22)) is different in each case. This is more clearly evident if we sum Eqs. (19) and (22) to obtain the dynamic equation for the total kinetic energy in the system:

$$\frac{dE}{dt} + \frac{dE_p}{dt} = -\Phi_\nu - \sum_{n=1}^{N} \frac{m_p^n \,|\, \mathbf{u}(\mathbf{x}_p^n) - \mathbf{v}_p^n(\mathbf{x}_p^n)|^2}{\tau_p^n}. \tag{23}$$

The total energy of the system is dissipated by two mechanisms, a homogeneous term due to viscous losses throughout the system and a term that results from the drag imparted by one phase on the other. The drag forces, although *conservative* in their momentum exchange, are *dissipative* in terms of the total kinetic energy.

B. *Consequences of Spatial Discretization*

The exact equations of motions presented in Section 2 cannot be solved numerically in the form they are shown because of the discrete representation of the fluid velocity field in the numerical method. The fluid velocity at discrete vertex points shall be designated by $\mathbf{u}^\nu$, where the superscript ''ν'' identifies a particular vertex point. Numerical approximation of the fluid velocity at a particle position $\mathbf{x}_p^n$ is determined by interpolating the fluid velocities at the vertex points. In principle, we could use the complete Fourier decomposition to obtain an excellent interpolated value for the fluid velocity; however, that would require a complete Fourier transform per particle, or $O(N_p N_\nu^3)$ operations, a computation that would be prohibitively expensive (note, N_ν is the total number of vertices). Instead, we choose to use interpolation functions similar to those proposed in the literature [48, 49]. We shall limit the discussion to interpolations that use a weighted sum of velocities in the neighborhood of the point of interest. For example, the velocity at a point $\mathbf{x}_p^n$ can be defined by

$$\mathbf{u}(\mathbf{x}_p^n) = \sum_{\nu=1}^{N_\nu} S(\mathbf{x}_p^n, \mathbf{x}^\nu)\mathbf{u}_\nu, \tag{24}$$

where the interpolation function $S(\mathbf{x}_p^n, \mathbf{x}^\nu)$ is constrained to satisfy

$$\sum_{\nu=1}^{N_\nu} S(\mathbf{x}_p^n, \mathbf{x}^\nu) = 1 \tag{25}$$

for arbitrary values of $\mathbf{x}_p^n$. Interpolation functions are often characterized by the order of accuracy of the interpolation (based on a local Taylor series expansion). In general, higher orders can be achieved by incorporating more vertex points. We have written the expression in its most general form so that any-order interpolation scheme is represented.

The spatially discrete update formula for the particle velocity can then be expressed as (note, for the sake of simplicity we do not consider the discretization in time, although it would have no effect on the outcome)

$$m_p^n \frac{d\mathbf{v}_p^n}{dt} = m_p^n \frac{\left[\sum_{\nu=1}^{N_\nu} S(\mathbf{x}_p^n, \mathbf{x}^\nu)\mathbf{u}^\nu - \mathbf{v}_p^n\right]}{\tau_p^n}. \tag{26}$$

As shown previously for the continuum, the equation for the total particle energy is then found by dotting Eq. (26) with the particle velocity. Summing the result over the entire particle population results in the following expression for the total particle kinetic energy:

$$\frac{dE_p}{dt} = \sum_{n=1}^{N_p} m_p^n \frac{\left[\sum_{\nu=1}^{N_\nu} S(\mathbf{x}_p^n, \mathbf{x}^\nu)\mathbf{u}^\nu - \mathbf{v}_p^n\right]}{\tau_p^n} \mathbf{v}_p^n. \tag{27}$$

The fluid phase is updated using the pseudospectral method outlined in Section 2. However, the forces from the particle phase must be reverse-interpolated to the vertex points of the grid since that is where the fluid phase is updated. If we define the total kinetic energy of the fluid

phase as the sum of the kinetic energy over all the vertex points, then its dynamic equation is

$$\frac{dE}{dt} = -\Phi_\nu - \sum_{\nu=1}^{N_\nu}\left\{\sum_{n=1}^{N_p}\frac{m_p^n\left[\sum_{\nu'=1}^{N_\nu} S(\mathbf{x}_p^n, \mathbf{x}^{\nu'})\mathbf{u}^{\nu'} - \mathbf{v}_p^n\right]}{\tau_p^n} S^*(\mathbf{x}_p^n, \mathbf{x}^\nu)\mathbf{u}^\nu\right\}, \tag{28}$$

where the reverse interpolation function, $S^*(\mathbf{x}_p, \mathbf{x}_\nu)$, is *not* necessarily the same as the forward function. As long as $S^*(\mathbf{x}_p, \mathbf{x}_\nu)$ satisfies the constraint shown in Eq. (25), the combined *momentum* of the fluid and particle phases will automatically be conserved. Furthermore, if the reverse interpolation scheme is the same as the forward, i.e., $S^*(\mathbf{x}_p, \mathbf{x}_\nu) = S(\mathbf{x}_p, \mathbf{x}_\nu)$, then the total *kinetic energy* (fluid plus particulates) will satisfy

$$\frac{dE}{dt} + \frac{dE_p}{dt} = -\Phi_\nu - \sum_{n=1}^{N}\frac{m_p^n\left|\sum_{\nu=1}^{N_\nu} S(\mathbf{x}_p^n, \mathbf{x}^\nu)\mathbf{u}^\nu - \mathbf{v}_p^n\right|^2}{\tau_p^n} \tag{29}$$

which is the discrete equivalent of Eq. (23). If different orders of interpolation are used, an error of the following form will result:

$$\text{Error} = \sum_{n=1}^{N_p}\frac{m_p^n\left[\sum_{\nu'=1}^{N_\nu} S(\mathbf{x}_p^n, \mathbf{x}^{\nu'})\mathbf{u}^{\nu'} - \mathbf{v}_p^n\right]}{\tau_p^n} \times \sum_{\nu=1}^{N_\nu}[S(\mathbf{x}_p^n, \mathbf{x}^\nu) - S^*(\mathbf{x}_p^n, \mathbf{x}^\nu)]\mathbf{u}^\nu. \tag{30}$$

The order of the error associated with inconsistent interpolations will be one plus the order of the scheme with maximum error. Of course, the cumulative error will be much larger. Thus, it appears that a necessary condition to satisfy the global kinetic energy balance for the particle–fluid system is to use the same interpolation function for forward and reverse coupling. This is imperative for simulation studies that investigate sensitive questions such as the rate of exchange of energy between the phases [7, 8].

There is an important dilemma introduced by the above result. In general, it is advantageous to use a high-order interpolation scheme for forward interpolation because it will improve the accuracy of the estimated value of velocity. However, raising the order of the reverse-interpolation increases the volume of influence of each individual particle. Given that the particle sizes are manifestly small as compared to the grid spacing (see Section 2), the spreading of the volume of influence of an individual particle over several grid volumes appears to be unphysical. Thus, one is tempted to use a high-order interpolation scheme for forward interpolations and a low-order one for reverse interpolations, resulting in the error shown in Eq. (30).

The importance of high-order accuracy on the forward interpolation has been well documented in the interpolation studies referred to earlier. We suggest that the reverse interpolation should use the same high-order scheme. Given the possible unphysical ''spreading'' that results from high-order reverse interpolation, we felt it was necessary to test how significant that effect was on the numerical simulation. The natural variable for characterizing this effect is the turbulent energy spectrum since it represents the distribution of energy on the basis of length scales. It has been observed in previous simulations of particle-laden flows (with two-way coupling) that the fluid energy spectrum at low wavenumbers is suppressed, as compared to the particle-free case, while the energy spectrum is augmented at higher wavenumbers [7, 8]. That is, the energy spectrum is observed to ''pivot'' around a characteristic wavenumber. It should be noted that the observed pivot point corresponds to a length scale that is much larger than the particle size and Kolmogorov scale; therefore the effect is assumed to be a collective one involving clusters of particles; however, this is speculative.

To test the effect of the reverse-coupling interpolation scheme, we have performed two simulations with identical parameter values but different grid resolutions. The low-resolution case used a 48^3 grid and the high-resolution case used a 96^3 grid. The fluid viscosity was fixed at a value such that the energy spectrum was well resolved in the low-resolution runs. The interpolation scheme was fixed in the two runs, thus the physical volume of influence of an individual particle was reduced by 1/8 in the high-resolution run. Therefore, if an appreciable error due to the unphysical spreading of the particle volume of influence was incurred, we would expect a significant disparity between the low and high resolution calculations. The particle–fluid simulations were performed for a sufficient time to observe the previously described ''pivoting'' of the spectrum. It can be seen from Fig. 7a that the decay of fluid turbulent energies for both resolutions is identical. Looking for any spectral redistribution, the energy spectra are shown collectively in Fig. 7b. Remarkably, the results from the two resolutions appear to be essentially the same over the first 24 wavenumbers. Indeed, the pivot point remained at $k = 6$ for both runs. We therefore conclude that the effect of the spreading due to reverse coupling is relatively minor, at least over the scales that are resolved by the fluid simulation. Furthermore, since high-order accuracy is essential for proper updating of the particle locations, we recommend using the same order in the reverse interpolation, thereby eliminating the error shown in Eq. (30).

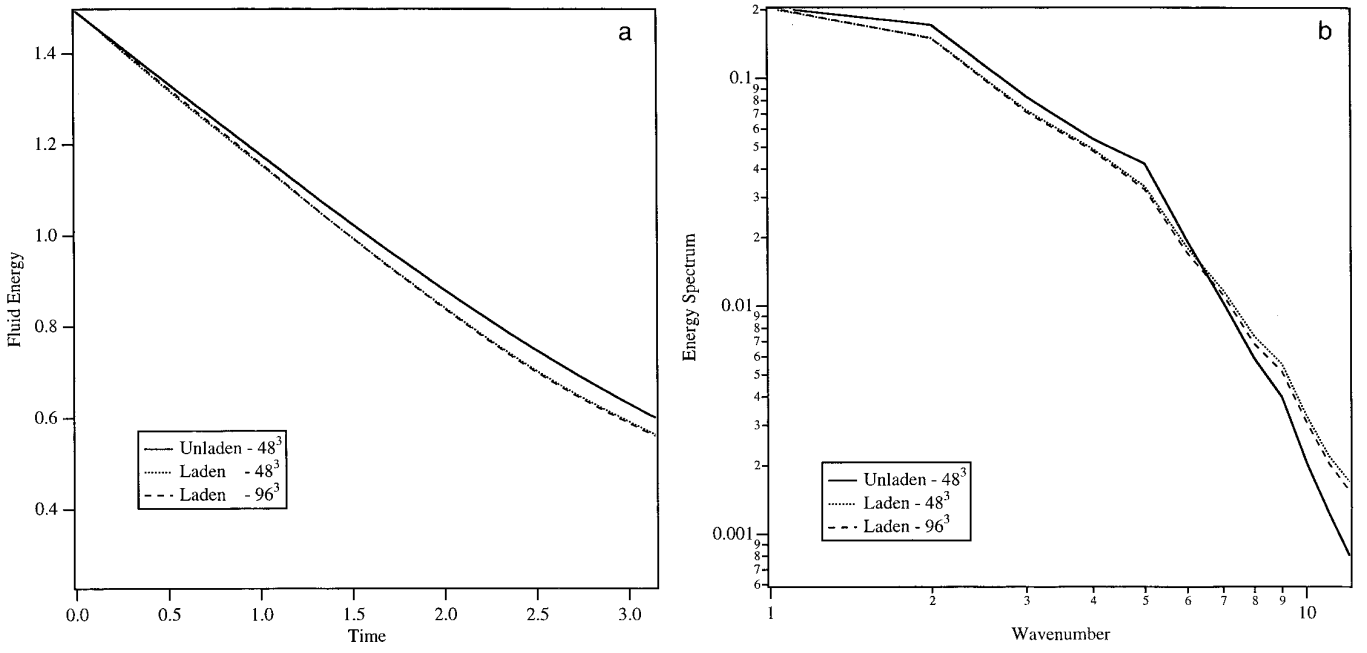

FIG. 7. Turbulent fluid energy in unladen (48^3) and particle laden (48^3 and 96^3) systems. The fluid energies for particle-laden fluids at both resolutions are nearly identical. (b) Comparison of fluid and particle energy spectra for 48^3 and 96^3 runs. Note that the pivot or crossover point remains stationary at $k = 6$ for both resolutions.

5. CONCLUSIONS

An algorithm for simulating finite-volume particles has been discussed. The introduction of more realistic particles produces several new numerical challenges, two of which are discussed in detail in this paper. They are: (i) numerical algorithm for handling particle–particle collisions; (ii) the proper formulation for reverse-interpolating the particle forces back to the computational grid. The critical issue associated with the former is reducing the computational cost of identifying the colliding pairs, an inherently $O(N_p^2)$ operation. This was addressed by subdividing the computational domain into smaller "neighborhoods." Potential partners for collision are selected only from particles within the neighborhood, substantially reducing the computational budget (from $O(N_p^2)$ to $O(N_p^2/M^3)$ operations). Two different methods of testing for collisions have been presented. The first (proactive) anticipates all the collisions that will occur within a time increment Δt, and then enacts them in the order that they occur. The second (retroactive) looks for particle–particle overlaps after the time increment has been completed, and then resolves those overlaps by enacting particle collisions retroactively in the order that they occurred until all overlaps have been accounted for. A comparison of the proactive method with sorting (best case) with the retroactive method without sorting (worst case) showed that the latter approach produced significant errors in the collision frequency, particularly at low values of the Stokes number. Furthermore, the computational cost of the former approach is roughly twice that of the latter.

One consequence of incorporating collisions into the algorithm is evidence of multiple collisions between a single pair of particles. This effect appears to be an artifact of the model used to estimate the forces on particles that are undergoing collision. For example, two particles that have just collided will be driven apart by the impulse force of the collision, however, if the particle response time is sufficiently small the fluid may redirect them towards each other causing a second collision. This process may occur several times until the particles "free" themselves from the path of the other. Physically, you would expect the particles to undergo a single interaction, altering the local flow field so as not to produce multiple contacts. That physics is not built into the model presented in this study. Fortunately, multiple collisions are unlikely to seriously affect the overall particle dynamics, however, it was pointed out that determining particle collision frequencies from simulation data requires care not to include the multiple collisions in your counting scheme. This is particularly relevant for particles with low Stokes numbers (i.e., $St < 0.4$).

There has remained some controversy associated with the proper algorithm for interpolating the reverse coupling effects back onto the grid. The controversy arises from the fact that the disturbance velocity created by the relative

motion of a particle and fluid scales with the particle diameter, a length scale that is much smaller than is resolved by the fluid equations. The outstanding question is then how should the reverse coupling be incorporated into the fluid equations that are numerically integrated on a much coarser grid. Error analysis of the particle and fluid equations with both forward and reverse coupling demonstrates that although momentum is properly conserved independently of the interpolation method, overall energy is biased if the forward and reverse interpolation schemes are not symmetric. That is, there is a systematic error associated with the difference between the two interpolation schemes. Earlier studies of forward interpolation have shown that a minimum of a third-order scheme is required to obtain accurate results. Consequently, we suggest that an equivalent interpolation scheme be used in the reverse direction, despite the minor objection that reverse coupling with a higher-order scheme tends to spread the influence of the particle over a larger volume. Numerical simulations with coarse and fine grid resolution demonstrated that the effect of spreading does not appear to significantly affect the dynamics of the mean energy and spectral pivot point in a particle-laden fluid.

ACKNOWLEDGMENTS

The authors thank Dr. J. U. Brackbill for several helpful suggestions on the analysis presented in Section 4. Financial support from Dow Chemical through the Young Minority Investigator Award (awarded to LRC) is also gratefully acknowledged.

REFERENCES

1. S. Sundaram and L. R. Collins, *in preparation*.
2. R. Gore and C. T. Crowe, *Int. J. Multiphase Flow* **15,** 279 (1989).
3. G. Hetsroni, *Int. J. Multiphase Flow* **15,** 735 (1989).
4. S. Sundaram and L. R. Collins, *J. Fluid Mech., in review* (1995).
5. M. Placnik, L. Klaboch, S. J. Haam, U. Utojo, and R. S. Brodkey, "The Measurement of Turbulence Parameters in Dispersive Flow of Solids in Pipes: An Index of Refraction Matching Technique," in *AIChE Annual Meeting, St. Louis, Missouri, 1993* (unpublished).
6. K. D. Squires and J. K. Eaton, *Phys. Fluids A* **3,** 1169 (1991).
7. K. D. Squires and J. K. Eaton, *Phys. Fluids A* **2,** 1191 (1990).
8. S. Elghobashi and J. C. Truesdell, *Phys. Fluids A* **5,** 1790 (1993).
9. L. P. Wang and M. R. Maxey, *J. Fluid Mech.* **256,** 27 (1993).
10. P. K. Yeung and S. B. Pope, *J. Comput. Phys.* **79,** 373 (1988).
11. P. K. Yeung, *Phys. Fluids* **6,** 3416 (1994).
12. S. Balachandar and M. R. Maxey, *J. Comput. Phys.* **83,** 96 (1989).
13. G. A. Kallio and M. W. Reeks, *Int. J. Multiphase Flow* **15,** 433 (1989).
14. K. Kontomaris, T. J. Hanratty, and J. B. McLaughlin, *J. Comput. Phys.* **103,** 231 (1992).
15. J. B. McLaughlin, *Phys. Fluids A* **1,** 1211 (1989).
16. J. W. Brooke, K. Kontomaris, T. J. Hanratty, and J. B. McLaughlin, *Phys. Fluids A* **4,** 825 (1992).
17. J. W. Brooke, T. J. Hanratty, and J. B. McLaughlin, *Phys. Fluids* **6,** 3404 (1994).
18. O. R. Walton, *Mech. Mat.* **16,** 239 (1993).
19. O. R. Walton, in *Particulate Two-Phase Flow*, edited by M. C. Roco (Butterworth–Heinemann, Boston, 1993), p. 885.
20. O. R. Walton and R. L. Braun, *Acta Mech.* **63,** 73 (1986).
21. M. Y. Louge, J. T. Jenkins, and M. A. Hopkins, *Phys. Fluids A* **2,** 1042 (1990).
22. C. S. Campbell and C. E. Brennen, *J. Fluid Mech.* **151,** 167 (1985).
23. C. S. Campbell and C. E. Brennen, *J. Appl. Mech.* **52,** 172 (1985).
24. S. F. Foerster, M. Y. Louge, C. Hongder, and K. Allia, *Phys. Fluids* **6,** 1108 (1994).
25. S. Sundaram and L. R. Collins, *Int. J. Multiphase Flow* **20,** 1021 (1994).
26. S. Sundaram and L. R. Collins, *Int. J. Multiphase Flow* **20,** 1039 (1994).
27. J. O. Hinze, *Turbulence* (McGraw–Hill, New York, 1975), p. 790.
28. J. J. Riley and G. S. Paterson, *Phys. Fluids* **17,** 292 (1974).
29. Elghobashi, *Appl. Sci. Res.* **48,** 301 (1991).
30. M. R. Maxey and J. J. Riley, *Phys. Fluids* **26,** 883 (1983).
31. C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics* (Springer-Verlag, New York, 1988).
32. S. A. Orszag and G. S. Patterson, Jr., *Numerical simulation of turbulence* (Springer-Verlag, New York, 1972), p. 127.
33. M. R. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Oxford Univ. Press, Oxford, 1987).
34. R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles* (McGraw–Hill, New York, 1981).
35. J. M. Haile, *Molecular Dynamics Simulation—Elementary Methods* (Wiley, New York, 1992).
36. L. Verlet, *Phys. Rev.* **159,** 98 (1967).
37. B. Quentrec and C. Brot, *J. Comput. Phys.* **13,** 430 (1975).
38. P. K. Yeung and S. B. Pope, *J. Fluid Mech.* **207,** 531 (1989).
39. S. E. Elghobashi and G. C. Truesdell, *J. Fluid Mech.* **242,** 655 (1992).
40. K. D. Squires and J. K. Eaton, *J. Fluid Mech.* **226,** 1 (1991).
41. P. G. Saffman and J. S. Turner, *J. Fluid Mech.* **1,** 16 (1956).
42. I. Kim, S. Elghobashi, and W. Sirignano, *J. Fluid Mech.* **246,** 465 (1993).
43. R. N. Parthasarathy and G. M. Faeth, *J. Fluid Mech.* **220,** 485 (1990).
44. S. Schreck and S. J. Kleis, *J. Fluid Mech.* **249,** 665 (1993).
45. Y. Tsuji, Y. Morikawa, and H. Shiomi, *J. Fluid. Mech.* **139,** 417 (1984).
46. Y. Tsuji, Y. Morikawa, T. Tanaka, and K. Karimine, *Int. J. Multiphase Flow* **14,** 565 (1988).
47. E. J. Chang and M. R. Maxey, *J. Fluid. Mech.* **277,** 347 (1994).
48. D. Burgess, D. Sulsky, and J. U. Brackbill, *J. Comput. Phys.* **103,** 1 (1992).
49. J. U. Brackbill and H. M. Ruppel, *J. Comput. Phys.* **65,** 314 (1986).