Journal of Computational Physics **170,** 523–549 (2001)
doi:10.1006/jcph.2001.6747, available online at http://www.idealibrary.com on IDEAL®

An Incompressible Three-Dimensional Multiphase Particle-in-Cell Model for Dense Particle Flows

D. M. Snider

Flow Analysis, LLC, 13616 Ernesto Court NE, Albuquerque, New Mexico 87112
E-mail: dsnider@nmia.com

Received May 15, 2000; revised November 28, 2000

A three-dimensional, incompressible, multiphase particle-in-cell method is presented for dense particle flows. The numerical technique solves the governing equations of the fluid phase using a continuum model and those of the particle phase using a Lagrangian model. Difficulties associated with calculating interparticle interactions for dense particle flows with volume fractions above 5% have been eliminated by mapping particle properties to an Eulerian grid and then mapping back computed stress tensors to particle positions. A subgrid particle, normal stress model for discrete particles which is robust and eliminates the need for an implicit calculation of the particle normal stress on the grid is presented. Interpolation operators and their properties are defined which provide compact support, are conservative, and provide fast solution for a large particle population. The solution scheme allows for distributions of types, sizes, and density of particles, with no numerical diffusion from the Lagrangian particle calculations. Particles are implicitly coupled to the fluid phase, and the fluid momentum and pressure equations are implicitly solved, which gives a robust solution. © 2001 Academic Press

Key Words: two-phase flow; Eulerian–Lagrangian flow; particle flow; multiphase particle-in-cell (MP-PIC); three-dimensional finite volume; Rayleigh–Taylor instability.

I. INTRODUCTION

Mathematical models of separated particulate multiphase flow have used either a continuum approach for all phases [1, 2] or a continuum for the fluid and a Lagrangian model for particles [3]. The continuum–continuum model readily allows modeling of particle–particle stresses in dense particle flows using spatial gradients of particle volume fractions [2, 4]. However, modeling a distribution of types and sizes of particles complicates the continuum formulation because separate continuity and momentum equations must be solved for each size and type [4, 5]. Using a continuum model for the fluid phase and a Lagrangian model for the particle phase allows economical solution for flows with a wide range of particle types, sizes, shapes, and velocities [4, 6]. However, the collision frequency is high for volume fractions above 5% and cannot be realistically resolved by current Lagrangian collision calculations [6].

Particle-in-cell (PIC) methods have been used since the 1960s [7]. Fluids are represented by discrete mass points. The differential conservation equations of mass, momentum, and energy govern the flow, although the conservation of mass is satisfied by the summation of mass points in a computational cell. Nontransport terms are calculated from the differential equations, and the transport terms are calculated from mass points moving by a velocity weighting procedure. The initial motivation for PIC methods was probably the accuracy in following interfaces. However, the calculation of convection is very accurate because of the Lagrangian nature of the PIC methods, and PIC methods offer the promise of subgrid resolution of those properties which are not updated on the grid. O'Rourke and Amsden [8] presented particle-in-cell models applied to chemically reacting flow. Particle properties were interpolated to the grid, and the flow field and particle interactions were calculated on the grid at the advanced time step. Properties were then interpolated back to particles. The calculation of particle interactions on the grid increased computational efficiency with no significant increase in numerical error. Andrews and O'Rourke [9] extended the PIC scheme to a multiphase particle-in-cell (MP-PIC) method for one-dimensional Eulerian–Lagrangian flow. In the method, particles are treated both as particles and as a continuum. The particle stress gradient, which is difficult to calculate for each particle in dense flow, is calculated as a gradient on the grid and is then interpolated to discrete particles. Snider *et al.* [10] extended the method to be two-dimensional, with an improved grid-to-particle interpolation method.

This paper retains the basic concept of treating particles both as a continuum and as discrete particles. Particle properties which are best calculated on the grid are calculated on the grid, and other particle properties are calculated at discrete particle locations. This paper defines interpolation operators and their properties which are well suited to three-dimensional calculations. The operators are fast and are shown to be locally and globally conservative. A model for the particle normal stress force on discrete particles is presented. The solution algorithm uses a split operator on the particle momentum calculation and a subgrid particle, normal stress applied to discrete particles. The solution method eliminates the need for a costly implicit solution of the particle normal stress on the grid. Further, the particle and fluid phases are implicitly coupled which provides a robust solution.

II. GOVERNING EQUATIONS

A. *Continuum Phase*

The continuity equation for the fluid with no interphase mass transfer is

$$\frac{\partial \theta_f}{\partial t} + \nabla \cdot (\theta_f \mathbf{u}_f) = 0, \tag{1}$$

where $\mathbf{u}_f$ is the fluid velocity and θ_f is the fluid volume fraction.

The momentum equation for the fluid is

$$\frac{\partial(\theta_f \mathbf{u}_f)}{\partial t} + \nabla \cdot (\theta_f \mathbf{u}_f \mathbf{u}_f) = -\frac{1}{\rho_f} \nabla p - \frac{1}{\rho_f} \mathbf{F} + \theta_f \mathbf{g}, \tag{2}$$

where ρ_f is fluid density, p is fluid pressure, and $\mathbf{g}$ is the gravitational acceleration. $\mathbf{F}$ is the rate of momentum exchange per volume between the fluid and particle phases. The fluid phase is incompressible and fluid and particle phases are isothermal. The momentum equation presented here neglects viscous molecular diffusion in the fluid but retains the viscous drag between particles and fluid through the interphase drag force, $\mathbf{F}$. Neglecting the laminar fluid viscous terms generally has negligible effect on dense particle flow, and laminar terms can be easily included in the fluid equation set. For the more interesting turbulent flow, there are currently no suitable models for dense particle flow. Large density and size particles act as large eddies of momentum transfer while gas flow around close pack particles produces small subgrid eddies and dissipation. Further, momentum transfer at walls is complicated by particles covering walls with particle sizes on the same order or larger than the viscous sublayer. This paper does not address turbulent dense particle flow. However, the discrete particle to fluid momentum transfer (which is a turbulent closure model for subgrid momentum transfer between particles and fluid) generally produces low Reynolds numbers (based on particle diameter) and provides an excellent prediction of dense particle flows over a wide range of gas flow.

B. *Particulate Phase*

The dynamics of the particle phase is described using the particle probability distribution function $\phi(\mathbf{x}, \mathbf{u}_p, \rho_p, \Omega_p, t)$, where $\mathbf{x}$ is the particle position, $\mathbf{u}_p$ is the particle velocity, ρ_p is the particle density, and Ω_p is the particle volume. For the present it is assumed that the mass of each particle is constant in time (no mass transfer between particles or to the fluid), but particles may have a range of sizes and densities. The time evolution of ϕ is obtained by solving a Liouville equation for the particle distribution function [11],

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\phi \mathbf{u}_p) + \nabla_{\mathbf{u}_p} \cdot (\phi \mathbf{A}) = 0, \tag{3}$$

where $\nabla_{\mathbf{u}}$ is the divergence operator with respect to velocity. The discrete particle acceleration is [9]

$$\mathbf{A} = D_p(\mathbf{u}_f - \mathbf{u}_p) - \frac{1}{\rho_p} \nabla p + \mathbf{g} - \frac{1}{\theta_p \rho_p} \nabla \tau. \tag{4}$$

The terms represent acceleration due to aerodynamic drag, pressure gradient, gravity, and gradient in the interparticle stress, τ.

The probability function integrated over velocity and mass gives the probable number of particles per unit volume at $\mathbf{x}$ and t in the interval $(\mathbf{u}_p, \mathbf{u}_p + d\mathbf{u}_p)$, $(\rho_p, \rho_p + d\rho_p)$, and $(\Omega_p, \Omega_p + d\Omega_p)$. The particle volume fraction is defined from the particle distribution function as

$$\theta_p = \iiint \phi \Omega_p \, d\Omega_p \, d\rho_p \, d\mathbf{u}_p. \tag{5}$$

The interphase momentum transfer function per volume in the Eulerian momentum equation is

$$\mathbf{F} = \iiint \phi \Omega_p \rho_p \left[D_p(\mathbf{u}_f - \mathbf{u}_p) - \frac{1}{\rho_p} \nabla p \right] d\Omega_p \, d\rho_p \, d\mathbf{u}_p. \tag{6}$$

The Eulerian governing equations for the particle phase may be obtained by taking the moments of Eq. (3). By multiplying Eq. (3) by $\rho_p \Omega_p$ and $\rho_p \Omega_p \mathbf{u}_p$ and integrating over particle density, volume, and velocity coordinates, the particle conservation equations are obtained. The particle continuity equation is

$$\frac{\partial(\overline{\theta_p \rho_p})}{\partial t} + \nabla \cdot (\overline{\theta_p \rho_p} \bar{\mathbf{u}}_p) = 0 \tag{7}$$

and the particle momentum equation is

$$\begin{aligned} \frac{\partial(\overline{\theta_p \rho_p} \bar{\mathbf{u}}_p)}{\delta t} + \nabla \cdot (\overline{\theta_p \rho_p} \bar{\mathbf{u}}_p \bar{\mathbf{u}}_p) &= -\theta_p \nabla p - \nabla \tau_p + \overline{\theta_p \rho_p} \mathbf{g} \\ &+ \iiint \phi \Omega_p \rho_p \, D_p(\mathbf{u}_g - \mathbf{u}_p) \, d\Omega_p \, d\rho_p \, d\mathbf{u}_p \\ &- \nabla \cdot \left[\iiint \phi \Omega_p \rho_p (\mathbf{u}_p - \bar{\mathbf{u}}_p)(\mathbf{u}_p - \bar{\mathbf{u}}_p) \, d\Omega_p \, d\rho_p \, d\mathbf{u}_p \right], \end{aligned} \tag{8}$$

where the mean particle velocity $\bar{\mathbf{u}}_p$ is given by

$$\bar{\mathbf{u}}_p = \frac{1}{\overline{\theta_p \rho_p}} \iiint \phi \Omega_p \rho_p \mathbf{u}_p \, d\Omega_p \, d\rho_p \, d\mathbf{u}_p, \tag{9}$$

and the average particle density is given by

$$\overline{\theta_p \rho_p} = \iiint \phi \Omega_p \rho_p \, d\Omega_p \, d\rho_p \, d\mathbf{u}_p. \tag{10}$$

The sum of volume fractions of fluid and particle phases must equal unity, $\theta_p + \theta_f = 1$.

III. INTERPOLATION OPERATORS

A. *Interpolation Operators*

Particle properties are interpolated to and from the Eulerian grid in the MP-PIC scheme. Interpolation operator properties are defined which are both locally and globally conservative in mapping to and from the grid. This study uses a staggered grid where momentum properties are calculated at cell faces and scalar properties are calculated at cell centers. Both scalar and momentum particle properties are needed. The continuity equation and pressure equation are calculated at cell centers. The momentum transfer between particles and gas is calculated at cell surfaces. This requires four sets of interpolation operators in three dimensions. Scalar properties, at cell centers, are mapped with one set of interpolation functions. Momentum properties, at face centers, are mapped with three other sets. The operators have the same definition but have different support in the axis of interpolation.

This study uses linear interpolation operators, but the scheme is applicable to other interpolation operators. The three-dimensional, trilinear interpolation operators are formed from the product of directional operators in the x, y, and z directions,

$$S = S^x S^y S^z. \tag{11}$$

If a nonorthogonal grid is used, control volumes and particles are transformed to a square computational grid. The interpolation operators and gradients are calculated and then transformed back to the grid. The following discussion is for a rectangular computational grid.

For a particle located at $\mathbf{x}_p$, where $\mathbf{x}_p = (x_p, y_p, z_p)$, the cell center x-directional interpolation operator, $S_i^x(x_p)$, is an even function, independent of the y and z coordinates, and has the properties

$$S_i^x(x_p) = \begin{cases} 0 & x_{i-1} \geq x_p \geq x_{i+1} \\ 1 & x_p = x_i \end{cases} \tag{12}$$

and

$$\sum_i S_i^x(x_p) = 1 \tag{13}$$

for all nodes, i, and in particular for nodes supporting $S_i^x(x_p)$. Similarly, the y-operator, $S_j^y(y_p)$, is an even function, independent of the x and z coordinates, and the z-operator, $S_k^z(z_p)$, is an even function, independent of the x and y coordinates. In two dimensions, four cell center grid nodes support the interpolation to particle position $\mathbf{x}_p$, and in three dimensions, eight grid nodes support the interpolation.

The face center directional interpolation operators are defined similarly to cell center operators. The x-face, directional interpolation operator has the properties

$$S_{i+1/2}^x(x_p) = \begin{cases} 0 & x_{i-1/2} \geq x_p \geq x_{i+3/2} \\ 1 & x_p = x_{i+1/2} \end{cases} \tag{14}$$

and

$$\sum_\xi S_\xi^x(x_p) = 1 \tag{15}$$

for all face nodes, ξ, and in particular for nodes supporting $S_\xi^x(x_p)$. Similarly, the y-face and z-face operators are defined.

The x, y, and z cell center directional operators for particle κ at location $\mathbf{x}_p$ are abbreviated $S_{i,\kappa}^x$, $S_{j,\kappa}^y$, and $S_{k,\kappa}^z$. The x, y, and z cell face directional operators are abbreviated $S_{i/2,\kappa}^x$, $S_{j/2,\kappa}^y$, and $S_{k/2,\kappa}^z$. The interpolation operators to particle κ at location $\mathbf{x}_p$ are:

Cell center interpolation operator: $S_{i,j,k,\kappa} = S_{i,\kappa}^x S_{j,\kappa}^y S_{k,\kappa}^z$
x-face interpolation operator: $S_{i/2,j,k,\kappa} = S_{i/2,\kappa}^x S_{j,\kappa}^y S_{k,\kappa}^z$
y-face interpolation operator: $S_{i,j/2,k,\kappa} = S_{i,\kappa}^x S_{j/2,\kappa}^y S_{k,\kappa}^z$
z-face interpolation operator: $S_{i,j,k/2,\kappa} = S_{i,\kappa}^x S_{j,\kappa}^y S_{k/2,\kappa}^z$.

The interpolation operator S satisfies $\sum_\xi S_\xi(\mathbf{x}_p) = 1$ for the support nodes ξ.

The cell center, x-linear interpolation operator at $\mathbf{x}_p$ on the computational grid is

$$S_i^x = \frac{x_{i+1} - x_p}{x_{i+1} - x_i}. \tag{16}$$

Similarly, the S^y and S^z operators are defined. The face cell directional interpolation operators are defined in the same way as the cell center interpolation operators, except using face nodes.

The cell volume fraction at grid (i, j, k) is

$$\theta_{p_{i,j,k}} = \frac{1}{\Omega_{i,j,k}} \sum_{\kappa=1}^{N_p} n_{p\kappa} \Omega_{p_\kappa} S_{i,j,k,\kappa}, \tag{17}$$

where the summation is over all particles, Ω_p is a particle volume, n_p is the number of particles in a parcel (a cloud of particles all with the same properties), N_p is the number of parcels or clouds, and the grid cell volume is $\Omega_{i,j,k}$.

B. *Product of Interpolation Operators*

The numerical scheme implicitly couples particles within the continuum fluid momentum equation, which leads to the product of interpolation operators such as

$$S_\zeta(\mathbf{x}_p) \sum_{\xi}^{8} S_\xi(\mathbf{x}_p) Q_\xi = \sum_{\xi}^{8} S_\zeta(\mathbf{x}_p) S_\xi(\mathbf{x}_p) Q_\xi, \tag{18}$$

where the grid property Q was mapped to a particle location and then back from particles to the grid. The expansion of Eq. (18) produces eight products of interpolation operators and results in a 27-point stencil for the Q property, in two dimensions. The resulting large stencil in three dimensions is computationally and computer memory expensive. The properties of product of interpolation operators are defined here which give compact support for the mapping. The interpolation operator properties are less diffusive than direct expansion of Eq. (18), and the operators are shown, in Appendix A, to be conservative.

The product of interpolation operators is defined as

$$S_\zeta\left(\mathbf{x}_{p_\kappa}\right) S_\xi(\mathbf{x}_{p\iota}) = \begin{cases} S_\xi\left(\mathbf{x}_{p_\kappa}\right) & \text{if } \xi = \zeta \text{ and } \iota = \kappa \\ 0 & \text{if } \xi \neq \zeta \text{ or } \iota \neq \kappa, \end{cases} \tag{19}$$

where ζ and ξ are grid nodes and ι and κ are particle location indices.

C. *Gradient of Interpolated Properties*

A grid property, Q, mapped to particle location $\mathbf{x}_p$, is

$$Q_p = \sum_{\xi=1}^{N} S_\xi(\mathbf{x}_p) Q_\xi, \tag{20}$$

where N is the grid nodes in support of the interpolation operator S_ξ. The gradient of the particle property is

$$\nabla Q_p = \sum_{\xi=1}^{N} \nabla S_\xi(\mathbf{x}_p) Q_\xi + \sum_{\xi=1}^{N} S_\xi(\mathbf{x}_p)(\nabla Q)_\xi. \tag{21}$$

At time t, the particle position and grid are fixed, and the first term is zero. The interpolation of the gradient to the particle position becomes

$$\nabla Q_p = \sum_{\xi=1}^{N} S_\xi(\mathbf{x}_p)(\nabla Q)_\xi. \tag{22}$$

D. *Product of Interpolation Operator and Gradient of Interpolation Operator*

The product of interpolation operator and the gradient of the interpolation operator is

$$\nabla S_\zeta(\mathbf{x}_p) \cdot \sum_{\xi=1}^{8} S_\xi(\mathbf{x}_p) \nabla Q_\xi = \sum_{\xi=1}^{8} \nabla S_\zeta(\mathbf{x}_p) \cdot [S_\xi(\mathbf{x}_p) \nabla Q_\xi], \tag{23}$$

where the grid property ∇Q was mapped to a particle location and then back from particles to the grid. The product of the interpolation operator and gradient of the interpolation operator is

$$\sum_\xi S_\zeta \nabla S_\xi = \sum_\xi S_{x_\zeta} S_{y_\zeta} S_{z_\zeta} \left[\frac{\partial S_{x_\xi}}{\partial x} S_{y_\xi} S_{z_\xi} \mathbf{e}_x + \frac{\partial S_{y_\xi}}{\partial y} S_{x_\xi} S_{z_\xi} \mathbf{e}_y + \frac{\partial S_{z_\xi}}{\partial z} S_{x_\xi} S_{y_\xi} \mathbf{e}_z \right], \tag{24}$$

where the unit vector is $\mathbf{e} = (\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$.

The definition of the product of interpolation operators requires $S_\xi S_\zeta = S_\xi$ if $\zeta = \xi$, else 0. Because an interpolation operator is formed from directional operators which are functions of only one independent direction, then for the product rule Eq. (19) to be true, the product of directional operators must also follow the product rule. That is,

$$S_\zeta^x\left(x_{p_\kappa}\right) S_\xi^x(x_{p\iota}) = \begin{cases} S_\xi^x\left(\mathbf{x}_{p_\kappa}\right) & \text{if } \xi = \zeta \text{ and } \iota = \kappa \\ 0 & \text{if } \xi \neq \zeta \text{ or } \iota \neq \kappa \end{cases} \tag{25}$$

for the x-directional operator, and similarly for the y- and z-directional operators.

The product of interpolation operator and gradient of interpolation operator then reduces to

$$\nabla S_\zeta(\mathbf{x}_p) \cdot \sum_{\xi=1}^{8} S_\xi(\mathbf{x}_p) \nabla Q_\xi = S_\zeta \left[\frac{\partial S_{x_\zeta}}{\partial x} \mathbf{e}_x + \frac{\partial S_{y_\zeta}}{\partial y} \mathbf{e}_y + \frac{\partial S_{z_\zeta}}{\partial z} \mathbf{e}_z \right] \cdot \nabla Q_\zeta. \tag{26}$$

For a linear interpolation, the derivative of the directional interpolation operator is a top-hat function.

IV. NUMERICAL SOLUTION

The governing equations for fluid and particles are solved on the computer. A computational particle method is used to solve for the particle distribution rather than direct solution of the Liouville equation. Particle properties are interpolated to and from the grid using the interpolation operators. The incompressible, three-dimensional continuum equations are solved using a finite volume method. The numerical method implicitly couples phases through the interphase momentum transfer. New time values are superscript $n+1$ and old time values are superscript n.

The conservation equations are approximated by finite volumes with staggered scalar and momentum nodes. The basic numerical scheme presented was also implemented on a collocated grid where both momentum and scalar variables are defined at a cell center. An advantage in using the collocated scheme is the use of only cell center interpolation operators for mapping particle properties to and from the grid. In the initial development, the collocated grid solution was not as robust as the natural staggered grid, and the collocated solution method was set aside.

A. *Particle Equations Finite Difference Approximation*

Particles are grouped into computational parcels (clouds) each containing n_p particles with identical mass density, ρ_p, volume, Ω_p, and velocity, $\mathbf{u}_p$, located at position, $\mathbf{x}_p$. The Liouville Eq. (3) is the mathematical expression of conservation of particle numbers in volumes moving along dynamic trajectories in particle phase space. Thus the number of particles, n_p, associated with a parcel is constant in time. Because there is no mass exchange between particles, a particle's mass, m_p, is also constant. Parcel positions are updated by

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t \mathbf{u}_p^{n+1} \tag{27}$$

and the particle velocity is updated from integration of Eq. (4).

$$\mathbf{u}_p^{n+1} = \frac{\mathbf{u}_p^n + \Delta t \left[D_p \mathbf{u}_{f,p}^{n+1} - \frac{1}{\rho_p} \nabla p_p^{n+1} - \frac{1}{\rho_p \theta_p} \nabla \tau_p^{n+1} + \mathbf{g} \right]}{1 + \Delta t \ D_p}, \tag{28}$$

where $\mathbf{u}_{f,p}^{n+1}$ is the interpolated implicit fluid velocity at the particle location, ∇p_p^{n+1} is the interpolated implicit pressure gradient at the particle location, $\nabla \tau_p^{n+1}$ is the interpolated particle stress gradient at the particle location, $\boldsymbol{g}$ is gravity acceleration, and D_p is the drag coefficient.

B. *Interphase Drag Model*

The interphase drag model used here is [4]

$$D_p = C_d \frac{3}{8} \frac{\rho_f}{\rho_p} \frac{|\mathbf{u}_f - \mathbf{u}_p|}{r}, \tag{29}$$

where

$$\begin{aligned} C_d &= \frac{24}{\text{Re}} \theta_f^{-2.65} (1 + 0.5\text{Re}^{0.687}) && \text{Re} < 1000 \\ C_d &= 0.44 \theta_f^{-2.65} && \text{Re} \geq 1000. \end{aligned} \tag{30}$$

The Reynolds number is defined as

$$\mathrm{Re} = \frac{2\rho_f|\mathbf{u}_f - \mathbf{u}_p|r}{\mu_f}, \tag{31}$$

where μ_f is the gas viscosity and the particle radius is

$$r = \left(\frac{3\Omega_p}{4\pi}\right)^{1/3}. \tag{32}$$

C. *Particle Normal Stress Model*

The particle velocity given by Eq. (28) can be solved directly at each time step using fluid properties updated from the current time step (new-time fluid velocities and pressure fields) and old-time properties for the particle normal stress tensor. However, because the particle normal stress is highly nonlinear, simply using old-time values for the particle normal stress and applying the gradient to particles in a cell does not work when particles are near close pack. Over a finite time interval, particles not unduly restricted by the old-time particle normal stress can enter a control volume (or within its interpolation range) and push the particle volume fraction to close pack. This, in turn, gives a very large particle normal stress which forces all particles out of the cell. One solution is to reduce the calculation time step or use subtime intervals for the particle momentum equation solution and volume fraction. Then the particle stress builds over each time step as particles approach the close pack volume fraction, and the particles stop before close pack occurs. If the particle has significant momentum, the time step needed may be very small, and in any case the method is computationally expensive. Another approach used by Snider *et al.* [10] was an implicit scheme for calculating the volume fraction (particle normal stress) on the Eulerian grid and then applying the implicit new-time, particle normal stress gradient to particles. The implicit scheme worked well for a wide range of flow conditions [12].

There are drawbacks to the implicit calculation of the particle volume fraction method used by Snider *et al.* [10]. The Eulerian implicit calculation of volume fraction is used solely to calculate the particle normal stress and is discarded at the end of a calculation step. The final volume fraction is calculated from mapping particle volumes to the grid. The implicit volume fraction equations are derived from a Taylor series expansion of Eq. (17), which includes the particle velocities, which in turn, includes the volume fraction through the nonlinear particle normal stress. The nonlinear volume fraction equations are solved by linearizing the particle normal stress and then iteratively solving the resulting linear equations, updating the linearized coefficients each iteration step [15]. The highly nonlinear volume fraction equations are difficult to solve and can be computationally expensive.

A second problem with the implicit calculation of the volume fraction is accuracy. The constant coefficients in the linearized volume fraction equations contain gradients of the particle interpolation operators. The problem arises because the interpolation gradient for discrete particles changes sign as particles move from one interpolation support to another. The fixed interpolation gradients in the Eulerian volume fraction coefficients do not reflect the change of gradients as particles move. The implicit solution can predict a nonphysical, negative volume fraction. The implicit calculation of volume fraction conserves volume, even if the local volume fraction is negative, and a false decrease in volume fraction at one node has an associated false increase in volume fraction at its neighbor node. Limiting the minimum value of the Eulerian calculated volume fraction to zero does not pose a problem in the solution scheme. A small volume fraction results in a trivial particle normal stress in the particle velocity calculation. On the other hand, a small error in volume fraction near close pack can give a very large particle normal stress and catastrophic results.

A new particle stress model is presented here. The particle normal stress is modeled by a continuum calculation of the particle pressure and the subsequent normal stress force is applied to discrete particles. The new Lagrangian discrete particle stress model which uses the stress gradient, eliminates the need for an implicit solution for the particle normal stress (particle volume fraction). The model is robust and fast which makes it well suited for three-dimensional calculations. Calculations using the new particle normal stress model compare well with experimental data.

Collisions between particles, where particles are modeled as a continuum, are estimated by an isotropic interparticle stress where the off-diagonal elements of the stress tensor are neglected. The continuum particle stress model used in this study is an extension of the model from Harris and Crighton [13]

$$\tau = \frac{P_s \theta_p^\beta}{\max[\theta_{cp} - \theta_p, \varepsilon(1-\theta_p)]}. \tag{33}$$

The constant P_s has units of pressure, and θ_{CP} is the particle volume fraction at close packing. For the constant β, Auzerais *et al.* [14] recommend $2 \leq \beta \leq 5$. The original expression by Harris and Crighton was modified to remove the singularity at close pack by adding the ε expression in the denominator. The ε is a small number on the order of 10^{-7}. The particle stress is unaffected by the modification except when the volume fraction approaches or exceeds close pack. The close pack limit is somewhat arbitrary and depends on the size, shape, and ordering of the particles. Therefore allowing the particle volume fraction to reach or slightly exceed close pack is physically possible considering that shifting or rearranging of granular materials may occur.

The particle stress Eq. (33) depends only on the concentration of particles and neglects the size and velocity of particles. A more complex, continuum particle normal stress model based on dense phase kinetic gas theory has been developed [15–17]. The model given by Lun *et al.* [16] is

$$\tau = \left[\theta_p \bar{\rho}_p + \theta_p^2 \bar{\rho}_p (1+\gamma) g_o\right] \Theta, \tag{34}$$

where γ is a restitution coefficient, and $\bar{\rho}_p$ is an average particle density. The granular temperature, Θ is given by

$$\Theta = \frac{1}{3}\langle C^2 \rangle, \tag{35}$$

where C is the instantaneous minus hydrodynamic velocity of the particle averaged over the velocity space. The radial distribution function is

$$g_o = \frac{3}{5}\left[1 - \left(\frac{\theta_p}{\theta_{cp}}\right)^{1/3}\right]^{-1}. \tag{36}$$

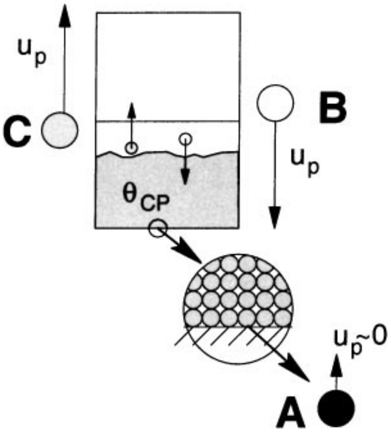


FIG. 1. Illustration of particles in a close pack volume.

The simple model, given by Eq. (33), contains the most important aspect of the kinetic theory model which is the dependency of normal stress on the particle concentration (particle volume fraction).

The Lagrangian or discrete particle solution allows subgrid modeling of the force. A model for the particle normal stress gradient applied to discrete particles is presented. The continuum particle pressure gradient from Eq. (33) gives the force on particles due to motion and inelastic collision of particles. The pressure gradient is assumed to be applied to particles in three general physical states near close pack. The three particles are shown in Fig. 1. For the initial discussion the mean velocity of particles (particles moving within a moving frame of reference) is zero but is later included in the model.

Consider the particle (cloud) ***A*** shown near the wall in Fig. 1. The particle is surrounded by neighbor particles or a wall and has zero mean velocity. For a large particle normal stress gradient, the resulting particle velocity and distance traveled based on only the particle normal stress may be unrealistically high. In the packed bed, the particle movement is limited to the particle collision mean free path and stress transfer to the wall. The effect of the large particle normal stress on particles deep within the packed bed is to slightly expand the bed.

Consider particle (cloud) ***B***, shown in Fig. 1, which rapidly approaches a close pack cell. The particle may be traveling through a low particle volume fraction region where the mean free path between particles is large, and its movement is not unduly restricted. The particle also may be within a stream of densely packed particles (particles moving within a moving frame of reference) and moving at the stream velocity. As the particle approaches the packed cell, the large particle normal stress from the close pack bed begins to reduce the particle's velocity. As the particle reaches the close packed region, the large particle normal stress stops the particle from penetrating into the cell and moves it away from the volume center. The physical process is collision with other particles and an elastic restitution.

Consider a particle leaving a near close pack bed, as shown by particle ***C*** in Fig. 1. The particle may have been forced from the bed by the particle normal stress or by other forces. The collision between particles diminishes as the particle moves away from the close pack volume, and the particle normal stress decreases. The particle mean free path increases and the particle has less restriction from collision with neighbor particles.

A computational algorithm is developed to model the proposed particle behavior described above. The particle velocity calculation is split into two parts, where the velocity is the sum of the particle velocity from the particle normal stress, and the velocity from all other forces. For a discrete particle

$$\mathbf{u}_p^{n+1} = \tilde{\mathbf{u}}_p + \mathbf{u}_{p\tau}. \tag{37}$$

The numerically integrated particle velocity without the particle normal stress force is

$$\tilde{\mathbf{u}}_p = \frac{\mathbf{u}_p^n + \Delta t D_p \mathbf{u}_{f,p}^{n+1} - \dfrac{\Delta t}{\rho_p} \nabla p_p^{n+1} + \Delta t \mathbf{g}}{1 + \Delta t D_p}. \tag{38}$$

The estimated discrete particle velocity from the continuum particle normal stress gradient is

$$\delta \mathbf{u}_{p\tau} = -\frac{\Delta t \nabla \tau_p}{\rho_p \theta_p (1 + \Delta t D_p)}. \tag{39}$$

An intermediate particle volume fraction calculation could be made at the end of the first-step velocity calculation, using the particle velocity from Eq. (38), which includes all forces on the particle except the particle normal stress. Intermediate particle locations could be calculated, and the particle volume interpolated to the grid giving an intermediate particle volume fraction, which, in turn, can be used to estimate the new-time continuum particle normal stress. The intermediate particle normal stress gradient would then be used in the second-step velocity calculation using Eq. (39). The calculation of an intermediate volume fraction is the same process as that at the end of the time step to get the new-time volume fraction. Because particles are numerically contained within the grid which is a ray tracing process, the intermediate calculation of volume fraction from interpolating particle volume to the grid would add a computationally expensive step for complex three-dimensional geometries. An alternative intermediate estimate of the volume fraction could be calculated on the grid using the Taylor series expansion of Eq. (17) as described by Snider *et al.* [12]. While this approach has merit, the extra computational time was found not to be warranted, and this intermediate particle stress calculation is not included in this model.

The particle normal stress, as given by Eq. [33], is only important near close pack and has little effect elsewhere. As an example of the magnitude of the particle normal stress, consider dense particle flow of sand ($\rho_p = 2500$ kg/m^3) at 92% close pack. For a close pack volume fraction of $\theta_p = 0.6$ and a steep volume fraction gradient from 0 to 0.55, the particle acceleration from particle normal stress is 0.048 m/s^2 using $P_s = 10$ Pa and $\beta = 3$ in Eq. [33]. Using gravity as a reference, the particle normal stress is small. As the particle volume fraction approaches close pack, the particle stress becomes extremely large. The continuum particle normal stress gradient applied to discrete particles is limited by the parcel velocity relative to mean flow velocity in the orthogonal directions $\mathbf{e}_\kappa = (\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$. The new time velocity is

$$\begin{aligned} \nabla\tau &\le 0 \\ u'_{p\tau_\kappa} &= \min(\mathbf{e}_\kappa \cdot \delta\mathbf{u}_{p\tau}, (1+\gamma)(\bar{\mathbf{U}}_p - \tilde{\mathbf{u}}_p)\cdot \mathbf{e}_\kappa) \\ u_{p\tau_\kappa} &= \max\left(u'_{p\tau_\kappa}, 0\right) \end{aligned} \tag{40}$$

$$\begin{aligned} &\nabla\tau > 0 \\ u'_{p\tau_\kappa} &= \max(\mathbf{e}_\kappa \cdot \delta\mathbf{u}_{p\tau}, (1+\gamma)(\bar{\mathbf{U}}_p - \tilde{\mathbf{u}}_p)\cdot\mathbf{e}_\kappa) \\ u_{p\tau_\kappa} &= \min\left(u'_{p\tau_\kappa}, 0\right), \end{aligned} \tag{41}$$

where the particle velocity is $\mathbf{u}_{p_\kappa} = (u_{p_x}, u_{p_y}, u_{p_z})$, γ is an elastic restitution factor, and the mean particle velocity, $\bar{\mathbf{U}}_p$, is the base or mean free flow of the particles where particles are moving within a moving frame of reference. The choice of variables in Eqs. (40) and (41) is to integrate the forces on the particle and to work with velocities.

The described particle normal stress model for discrete particles is designed for fast computational speed while having a physical basis for applying the particle normal stress gradient. The model, given by Eqs. (40) and (41), is applied to the three particles shown in Fig. 1. The particle normal stress gradient for the packed bed is large and negative, which gives a large positive velocity from the particle normal stress alone (Eq. (39)).

Particle ***A***, at the bottom of the stagnant bed, has a gravity force and possible fluid drag giving a negative velocity. The resulting particle normal stress velocity from Eq. (40) matches the velocity from gravity and fluid drag from Eq. (38). The resulting combined velocity from Eq. (40) is zero or a small positive velocity from elastic restitution. The particle may move slightly in the direction of the particle normal stress force, but, for the most part, the particle remains stationary (relative to the particle base flow).

Particle ***B*** rapidly approaches the packed bed. Away from the packed bed, the particle velocity from Eq. (38) is greater than the velocity from the particle normal stress and the particle will slow from the particle normal stress, but the particle will continue to move toward the packed region. As the particle moves closer to the packed bed, the particle velocity is less than the velocity resulting from a large particle normal stress, and the particle is stopped and bounces back depending on the elastic restitution factor.

Particle ***C*** is moving away from the packed region and is at the fringe of the packed bed where the volume fraction is low. The resulting normal stress particle velocity from Eq. (40) is zero, and the particle ***C*** velocity is not restricted by the particle normal stress and its velocity is given by Eq. (38).

The model needs one additional property which is the combination of properties for particles ***A*** and ***C***. The particle is moving in the direction of the particle pressure force but is within the packed bed. The particle motion will be limited by particle collisions. The model limits the motion of particles in close pack region to a collision mean free path. A mean free path is defined as

$$\ell = f\bar{r}_p, \tag{42}$$

where $\bar{r}_p$ is an effective particle radius. The formula for the mean free path given by Gidaspow [4],

$$\ell = \frac{1}{3\sqrt{2}}\frac{\bar{r}_p}{\theta_p}, \tag{43}$$

provides guidance in defining f.

The particle normal stress model given in this paper gives a natural limiting of the particle volume fraction to close pack. There are no other restrictions to force particles to less than close pack besides those applied in the particle stress model. The particle normal stress gradient as given in the particle momentum Eq. (4) is conservative [9]. The particle stress model described in this paper conserves the interpolation of the particle stress gradient to and from the grid except in the narrow volume fraction region at close pack. This lack of conservation of interpolating the stress gradient to and from the grid at close pack is inconsequential for a number of reasons. First, numerically, the grid particle normal stress is not used in an Eulerian solved conservation equation. If the gradient is needed for an Eulerian conservation calculation, the particle stress mapped to the grid will give the consistent particle normal stress gradient. Secondly, because the grid particle normal stress models (either the simple model Eq. (33) or the more complex model based on kinetic theory of gases [15]) has such an extreme nonlinear variation in force with only a small change in volume fraction at close pack, the adjustment in particle stress gradient for discrete particles gives very close to the same calculated volume fraction from either the particle or the Eulerian solution (both give a calculated volume fraction near close pack).

V. EULERIAN NUMERICAL APPROXIMATIONS

The Eulerian momentum equations and pressure equation (which is derived from volume continuity) are approximated by finite volumes with staggered scalar and momentum nodes. The conservation of volume is calculated in a Lagrangian frame of reference. The finite volume equations are developed in Cartesian coordinates for an orthogonal grid. The numerical representation gives linear algebraic equations with explicit coefficients multiplied times the velocity and pressure. The coefficients are built up by adding the contribution from each part of the numerical approximation.

A. *Continuity and Pressure Equation*

Conservation of volume is calculated explicitly in the Lagrangian frame of reference. After grid momentum equations are solved, which implicitly couple the particles and fluid, grid properties are interpolated back to particle (parcel) positions and used in a final explicit calculation of parcel velocities. Unlike most single fluid PIC methods, in the MP-PIC method, particles move with their own velocity rather than a velocity interpolated from the grid. Following the particle velocity calculation, the particle positions are updated. The new volume fraction is calculated by interpolating particle volumes to the grid using Eq. (17). The new-time fluid volume fraction is calculated from the conservation of volume relation $\theta_f = 1 - \theta_p$.

For incompressible flow the Eulerian continuity equation reduces to conservation of volume. A pressure field is constructed which guarantees satisfaction of the fluid continuity equation. The pressure equation is formed from combination of the fluid continuity equation and fluid momentum equation. Unlike single-phase incompressible flow, the divergence of individual phase velocities (continuity equation) is not zero but is balanced by the time rate of change of the phase volume. For an Eulerian solution of all phases, the phases can be added to eliminate the time rate of change of volume. In the Eulerian–Lagrangian solution, a pressure field is calculated which maintains fluid phase continuity. The time rate of change of fluid volume, in the continuity equation, is the negative of the rate of change of the particle volume interpolated to the grid. A pressure-correction scheme similar to SIMPLE is used to adjust the pressure and fluid velocity fields which satisfy continuity. The pressure correction method follows those for single phase flow, and details of pressure-correction algorithms are given by Ferziger and Peric’ [18].

B. *Momentum*

The finite volume approximation to the u-momentum Eq. (2) for staggered node ξ can be written as

$$c_\xi u_{f_\xi}^{n+1} = \sum_\eta c_\eta u_{f_\eta}^{n+1} + \sum_\eta a_\eta p_\eta^{n+1} + \Omega_\xi (F_x)_\xi^{n+1} + C'_\xi, \tag{44}$$

where the summations, η, are over neighbor nodes, F_x is the interphase momentum transfer, Ω is the volume, and C' is a constant “source.” The explicit constant coefficients c and a are from the numerical approximation to the momentum equation using numeric methods such as given by Ferziger and Peric’ [18] or Anderson *et al.* [19]. In this study, the time derivative is approximated by a backward difference, and the convective term is a blended central difference and upwind approximation. For a momentum cell, the convective contribution from the right neighbor to the constant coefficient is

$$c_r = c'_r + \begin{cases} \rho_f (\theta_f u_f)_r^n A_r & \text{if } u_{f_r}^n \geq 0 \\ 0 & \text{if } u_{f_r}^n < 0, \end{cases} \tag{45}$$

where c'_r, on the right side of Eq. (45), represents the building of a coefficient from multiple numerical steps. The momentum cell right face flux, $(\theta_f u_f)_r^n$, is the interpolated value between the product of the velocity and volume fraction at face node ξ and its right neighbor value, and A_r is the right area of the momentum cell. Similar terms result for the other five faces and the center momentum node. The blend of upwind and central difference results in a blending factor multiplying terms in Eq. (45) and an explicit central difference momentum term added to the equation source term. The left and right pressure coefficients in Eq. (44) are

$$a_l = A_l \quad a_r = -A_r, \tag{46}$$

where A_l and A_r are the momentum cell left and right faces, respectively.

C. *Interphase Momentum Transfer*

Up to this point, a momentum equation contains adjacent cell velocities and pressures. An implicit-coupled, interphase momentum transfer between particles and fluid contributes to existing terms and can add new terms. The implicit interphase momentum transfer at momentum node ξ is

$$\mathbf{F}_\xi^{n+1} = \frac{1}{\Omega_\xi} \sum_{\kappa=1}^{N_p} S_{\xi_\kappa} \left[D_{p_\kappa} \left(\mathbf{u}_{f,p_\kappa}^{n+1} - \mathbf{u}_{p_\kappa}^{n+1} \right) - \frac{1}{\rho_{p_\kappa}} \nabla p_{p_\kappa}^{n+1} \right] n_{p_\kappa} m_{p_\kappa}, \tag{47}$$

where m_p is the mass of a particle and n_p is the number of particles in a parcel. Putting the particle velocity given by Eq. (28) into the interphase drag and putting the interphase drag into the fluid momentum equation, the numerical approximated momentum equation is

$$\begin{aligned}
&\frac{\Omega_\zeta \rho_f (\theta_f \mathbf{u}_f)_\zeta^{n+1}}{\Delta t} + \mathbf{F}_{c_\zeta}^{n+1} = \frac{\Omega_\zeta \rho_f (\theta_f \mathbf{u}_f)_\zeta^{n}}{\Delta t} - \Omega_\zeta \nabla p_\zeta^{n+1} + \Omega_\zeta \rho_f (\theta_f)_\zeta \mathbf{g} \\
&\quad - \sum_\kappa \frac{m_{p_\kappa} n_{p_\kappa} S_{\zeta_\kappa} D_{p_\kappa}}{\left(1 + D_{p_\kappa} \Delta t\right)} \sum_\xi S_{\xi_\kappa} (\mathbf{u}_f)_\xi^{n+1} + \sum_\kappa \frac{m_{p_\kappa} n_{p_\kappa} S_{\zeta_\kappa}}{\left(1 + D_{p_\kappa} \Delta t\right) \rho_{p_\kappa}} \sum_\xi S_{\xi_\kappa} \nabla p_\xi^{n+1} \\
&\quad + \sum_\kappa \frac{m_{p_\kappa} n_{p_\kappa} S_{\zeta_\kappa} D_{p_\kappa}}{\left(1 + D_{p_\kappa} \Delta t\right)} \left(\mathbf{u}_{p_\kappa}^n + \Delta t \mathbf{g}\right) - \sum_\kappa \frac{m_{p_\kappa} n_{p_\kappa} S_{\zeta_\kappa} \Delta t D_{p_\kappa}}{\left(1 + D_{p_\kappa} \Delta t\right) \rho_p} \sum_\xi S_{\xi_\kappa} \frac{\nabla \tau_\xi}{\theta_p^n},
\end{aligned} \tag{48}$$

where $\mathbf{F}_\mathrm{c}$ is the advective terms. The summation, κ, is over all particles. Abbreviated subscripts are used where ζ is the momentum node, and ξ is the summation of the cell grid nodes supporting the interpolation to a particle location.

If a top-hat interpolation is used, the fluid velocity at particle position $\mathbf{x}_p$ is either zero or the node velocity. If a trilinear or other interpolation is used, the fluid velocity at particle position $\mathbf{x}_p$ includes the node velocities in support of the interpolation. Considering only the fluid velocity term in the gas to particle drag Eq. (47), the interphase momentum in the fluid equation is

$$\mathbf{F}_\zeta^{n+1} = \cdots + \sum_{\kappa=1}^{N_p} S_{\zeta_\kappa} D_{p_\kappa} n_{p_\kappa} m_{p_\kappa} \sum_{\xi=1}^{N} S_{\xi_\kappa} \mathbf{u}_{f_\xi} + \cdots, \tag{49}$$

where the first summation is over all particles, N_p, interpolating the particle momentum to the grid. The second summation is over nodes, N, supporting the interpolation of the fluid velocity to particle locations. In two dimensions with bilinear interpolation, the above interphase momentum introduces the upper left and right and the bottom left and right neighbor velocities which gives a 9-point velocity stencil. Using the interpolation product definition Eq. (19), the velocity term in Eq. (47) reduces to

$$\mathbf{F}_\zeta^{n+1} = \cdots + u_{f_\zeta} \sum_{\kappa=1}^{N_p} S_{\zeta_\kappa} D_{p_\kappa} n_{p_\kappa} m_{p_\kappa} + \cdots, \tag{50}$$

which maintains the tight support of only neighbor velocities. For a staggered grid, the implicit interphase momentum can increase the two-dimensional pressure stencil from 5 to 12. Using the interpolation product definition, Eq. (19), no additional pressure terms are added to the stencil. Using the definition of the interpolation operator product, double summation terms in the momentum Eq. (48) are reduced to single summations over particles.

Equation (48) can be cast into linear algebraic equations similar to Eq. (44) with new coefficients and constants. If the fourth term on the right side of Eq. (48) was expanded, additional neighbor velocities would be included in the fluid equation. However, applying the definition for multiplied interpolation operators, only the advective term, $\mathbf{F}_\mathrm{c}$, gives neighbor velocities which have the same coefficients as in Eq. (44). The diagonal coefficient of the fluid velocity will have an additional contribution, and the pressure coefficients will have additional contributions. The other terms are lumped into the constant. The equations are solved using a conjugate gradient solver [20].

VI. TEST PROBLEMS

Four three-dimensional test problems are given. The first two problems demonstrate the MP-PIC method for calculating particle flow where the particle normal stress is not a significant force. The second two problems demonstrate the MP-PIC method for particle flow at close pack where the particle normal stress is a major force. The first problem is calculating one-dimensional settling of particles for which there is an analytical solution. The three-dimensional calculation of one-dimensional settling of heavy particles above a lighter fluid requires care in initial conditions. Small perturbations in the initial interface will lead to three-dimensional Rayleigh–Taylor-type plumes. The second test problem is a Rayleigh–Taylor problem where the interphase particle drag is high. Small perturbations in interface volume fraction grow into large structures. The growth of the three-dimensional buoyancy driven mixing layer is compared with measured data. The third problem is gravity sedimentation where heavy particles settle to close pack. The last test problem is the blowing of a jet of particles onto a plate. The problem is three-dimensional and nonlinear, with particles close packing under the jet.

A. *One-Dimensional Layered Sedimentation*

The one-dimensional sedimentation of a dense particle–fluid mixture above a lighter fluid is calculated. The problem considers gravity-driven particles falling through gas and depositing at the bottom of the container. The three-dimensional solution requires that there are no initial perturbations. Small perturbations quickly grow to three-dimensional structures. The problem also requires a high level of convergence so that numerical perturbations do not lead to three-dimensional structures. The convergence criterion was a maximum residual less than 10^{-15} on all equation sets. Eventually numerical instabilities will lead to three-dimensional flow in this inherently unstable problem.

Andrews and O'Rourke [9] provided an analytical solution for the one-dimensional sedimentation and only the rudimentary details are given here. A granular mixture of single size particles occupies 20% of the upper half of the container. Table I gives details of the problem. The interphase drag is truncated from Eq. (30) to $C_d = \frac{24}{\mathrm{Re}}\theta_f^{-2.65}$ to allow an analytical solution. Using a drift flux approximation, a nonlinear differential equation results. Particles at the bottom of the expanding mixing layer, where the particle volume

TABLE I
One-Dimensional Sedimentation Calculation

Number of parcels	52,000
Particle radius	0.001 m
Particle density	1000 kg/m^3
Fluid density	1 kg/m^3
Fluid viscosity	0.02 kg/ms
P_s	5 Pa
θ_{cp}	0.7
Gravity	−9,0,0 m/s^2
Number cells	50 × 5 × 5
Domain	1 × 0.5 × 0.5 m
D_{Stokes}	90 s^{-1}
Calculation time step	0.001 s

fraction is near zero, fall at near the Stoke's free-fall velocity of 0.1 m/s. At the top of the mixture, which has a constant volume fraction, the expansion the particle volume fraction has the characteristic solution

$$\theta = \begin{cases} v^{-1}(x/t) & v(\theta_{-\infty})t \le x \le v(\theta_{+\infty})t \\ \theta_{-\infty} & x < v(\theta_{-\infty})t \\ \theta_{+\infty} & x > v(\theta_{+\infty})t, \end{cases} \tag{48}$$

where

$$v = \frac{g(\theta_f)^n}{D_p}[(2+n)\theta_p - 1]. \tag{49}$$

The upper part of the slug will remain a step function because of the crossing of characteristics. For Stoke's flow $n = 1$, and for Stoke's flow with the hindrance function, $n = 3.65$.

Figure 2 shows the calculated particle volume fraction at times during the transient. Particles at position ***C*** fall at near the Stoke's free-fall velocity 0.1 m/s and reach the bottom in 5 s. The numeric calculation predicts this expanding mixing layer well. At point ***B***, the particle volume fraction is 0.2, and, from the characteristic, the mixture edge rises at 0.00576 m/s. The numerical solution also calculates a rise in the mixture edge at ***B***, but the numeric resolution is too low to provide a quantitative comparison with the analytic value. At the top edge of the mixture, ***A***, the particles fall at $0.0443\theta_f$ s. The calculation predicts accurately this kinematic shock. There is also calculated, high wave number harmonics on the particle volume fraction which are not in the analytic solution. Because of the dependence on particle volume fraction between ***A*** and ***B***, the linear solution does not apply. Calculating a one-dimensional, heavy-above-light fluid problem using a three-dimensional solution is difficult. Perturbations whether physical or numerical lead to instabilities. The one-dimensional test problem shows that with care and high convergence tolerances, the solution scheme can predict a stable solution for an unstable problem without perturbations (for a while). Note that eventually perturbations form in the calculation. The next test problem shows that the numerical scheme can equally well calculate an unstable problem starting from small perturbations.

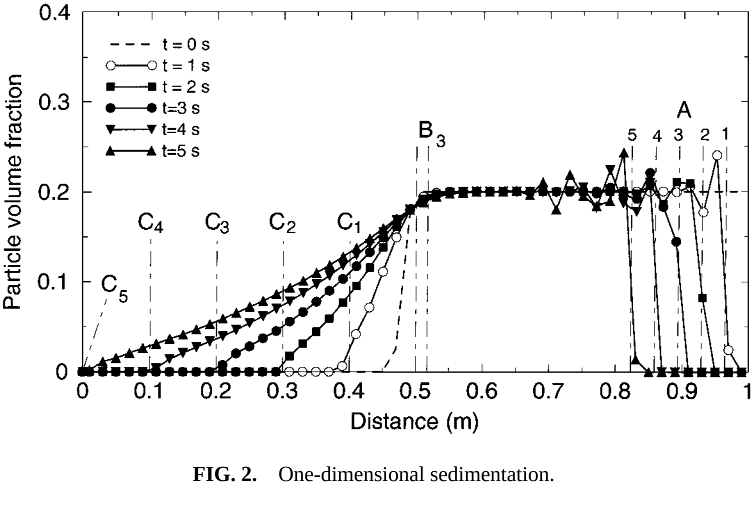


FIG. 2. One-dimensional sedimentation.

B. *Rayleigh–Taylor Problem*

Rayleigh–Taylor heavy fluid, above light fluid is inherently an unstable problem. Small perturbations grow and form into three-dimensional structures. There has been substantial measured data on the growth of Rayleigh–Taylor mixing layers. Following the initial exponential instability growth period, the growth of the mixing layer is well described by

$$h = \alpha A g t^2,$$

where h is the half width of the mixing layer, g is gravity, t is time, and the Atwood number is $A = (\rho_1 - \rho_2)/(\rho_1 + \rho_2)$. The value of the constant α is still undergoing scrutiny, but the wealth of measured data suggests that the constant α lies between 0.05 to 0.07. [21–24].

The heavy fluid, above light fluid (Rayleigh–Taylor) problem is calculated using the multiphase particle-in-cell numerical scheme. The calculation is three-dimensional and is in a brick-shaped container. The heavy fluid is a mixture of fluid and particles. The interphase drag is large producing the mixture fluid. The particles are a single size and density and are uniformly, randomly distributed in the upper section of the container. No initial perturbation is placed on the interface. Small perturbations result from the random distribution of particles. Table II gives the calculation parameters.

Six calculations were made at four Atwood numbers. Two calculations were at the same Atwood number and a different grid density. Two calculations were at the same Atwood number but different levels of perturbation magnitude. Figure 3 shows the calculated developing mixing layer for $A = 0.1667$, looking at an angle from the bottom. Figure 4 shows a slice in the middle of the container.

The growth rate of the mixing layer is shown in Fig. 5. The curves turn over at the end time as the mixing layer reaches the bottom wall. All calculations fit well with h using $\alpha = 0.07$, which, in turn, compares well with the measured data.

C. *Gravity-Dominated Particle Flow*

A uniform, well-mixed suspension of sand particles and air are left to settle to close pack. The calculation parameters are given in Table III. Particles are initially motionless and are

TABLE II
Rayleigh–Taylor Calculation

Particle radius	0.001 (μm)	0.001 (μm)	0.001 (μm)	0.001 (μm)
Particle density	10 (kg/m^3)	5 (kg/m^3)	3 (kg/m^3)	11 (kg/m^3)
Fluid density	1 (kg/m^3)	1 (kg/m^3)	1 (kg/m^3)	1 (kg/m^3)
Initial particle volume fraction	0.2	0.2	0.2	0.2
Particle mixture density	2.8 (kg/m^3)	1.8 (kg/m^3)	1.4 (kg/m^3)	3 (kg/m^3)
Interphase drag	10^5 s^{-1}	10^5 s^{-1}	10^5 s^{-1}	10^5 s^{-1}
x, *y*, *z* gravity	0, −9.8, 0 (m/s^2)	0, −9.8, 0 (m/s^2)	0, −9.8, 0 (m/s^2)	0, −2, 0 (m/s^2)
Number *x*, *y*, and *z* cells	40 × 40 × 80	40 × 40 × 80	40 × 40 × 80	50 × 50 × 90
x-range	10 cm	10 cm	10 cm	10 cm
y-range	10 cm	10 cm	10 cm	10 cm
z-range	20 cm	20 cm	20 cm	20 cm
Atwood number	0.4737	0.2857	0.1667	0.5
Number particles	5.3×10^5	5.3×10^5	5.3×10^5	2×10^5
Calculation time step	0.002 s	0.002 s	0.002 s	0.002 s

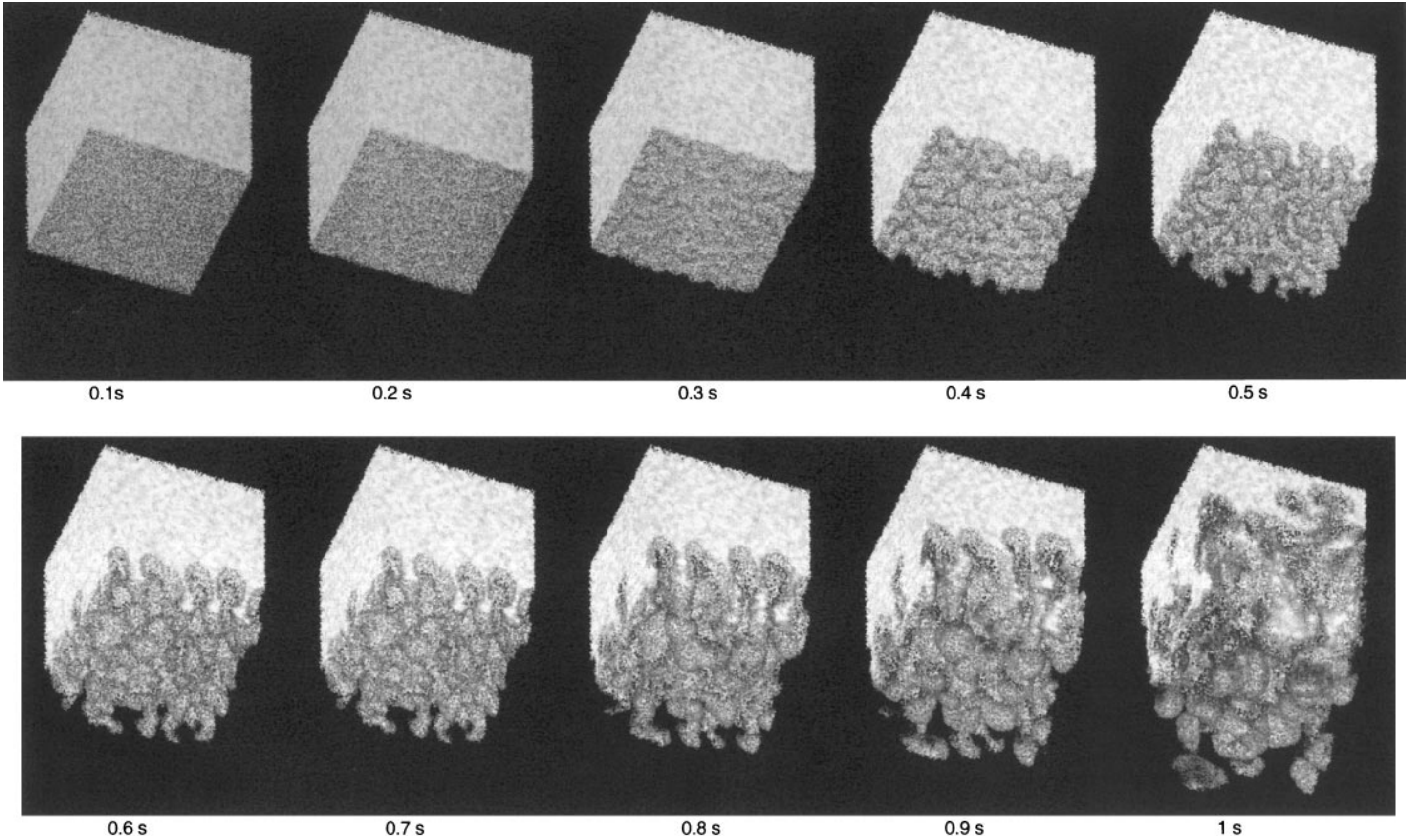


FIG. 3. Perspective view of three-dimensionally growing plumes for $A = 0.1667$.

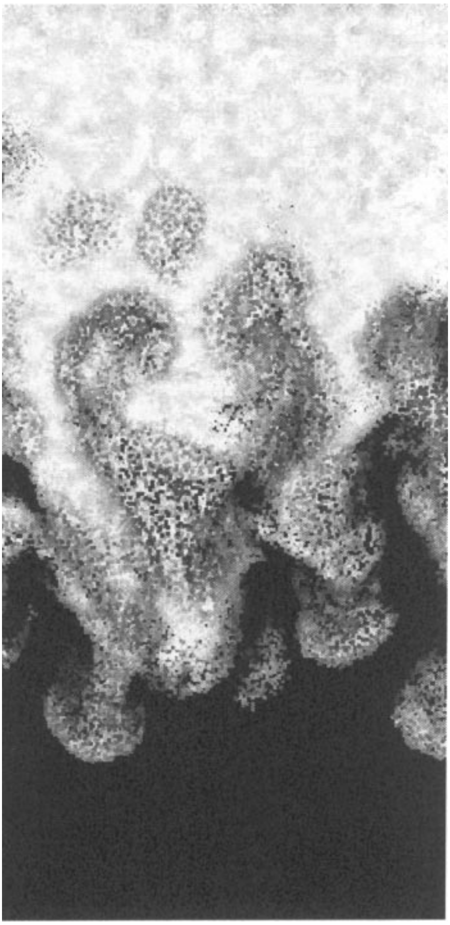


FIG. 4. Side view and cut through center of three-dimensionally growing plumes for $A = 0.1667$ at $t = 0.9$ s.

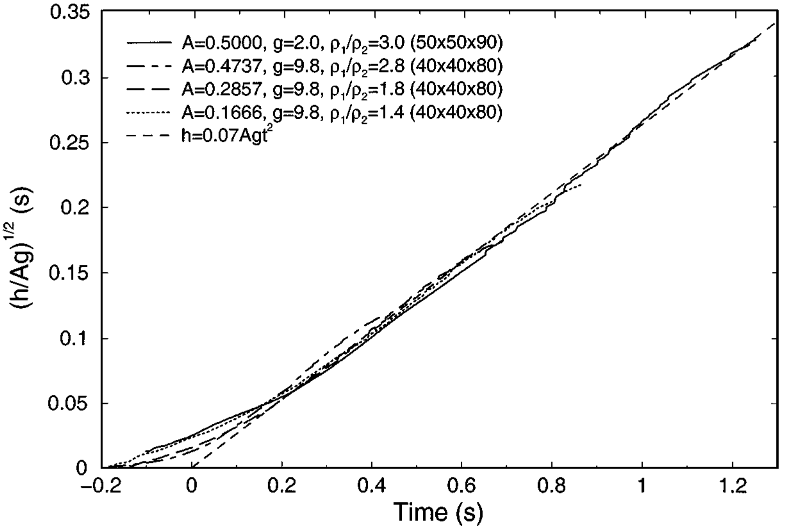


FIG. 5. The growth rate of the mixing width.

TABLE III
Gravity-Dominated Flow Calculation

Number of parcels	162,057
Particle radius	300 μm
Particle density	2500 kg/m^3
Fluid density	1.093 kg/m^3
Fluid viscosity	1.95×10^{-5} kg/ms
Initial particle volume fraction	0.3
P_s	10 Pa
β	2
θ_{cp}	0.6
Gravity	0, 0, -9.8 m/s^2
Number cells	$15 \times 15 \times 40$
Size of container	$13.8 \times 13.8 \times 30$ cm
Calculation time step	0.002 s

Note. Particles are initially uniformly randomly distributed in a container.

uniformly, randomly distributed which give initial small fluctuations in volume fraction about the average 0.3 volume fraction, as shown in Fig. 6. The driving force is from gravity. The particles are heavy with a constant, relatively large size, and the fluid is light, which gives gravity-dominated flow in the 0.3-m deep container. The upper mixture interface between suspended particles and clarified fluid is approximated well by $h = gt^2/2$. Figure 6 shows the particle volume fractions, and Fig. 7 shows the particle distributions at stages of filling. The interface between clarified fluid and mixture at 0.1 and 0.15 s matches well with the gravity-dominated flow values of 0.25 and 0.19 m from the bottom, respectively. The MP-PIC solution, with the particle normal stress model presented in this text, gives a natural settling to close pack. Figure 6 shows that at 0.2 s the entire particle mixture is close pack, and at 8 s no further settling beyond close pack has occurred.

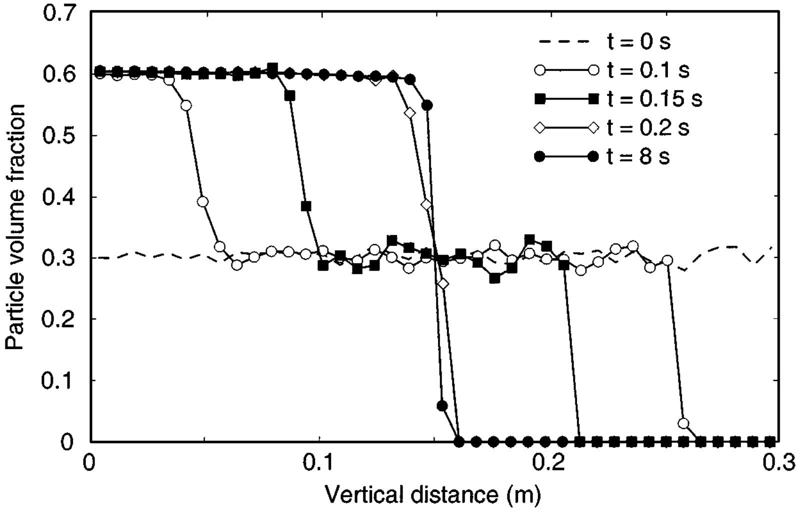


FIG. 6. Volume fraction at times during gravity-dominated sedimentation.

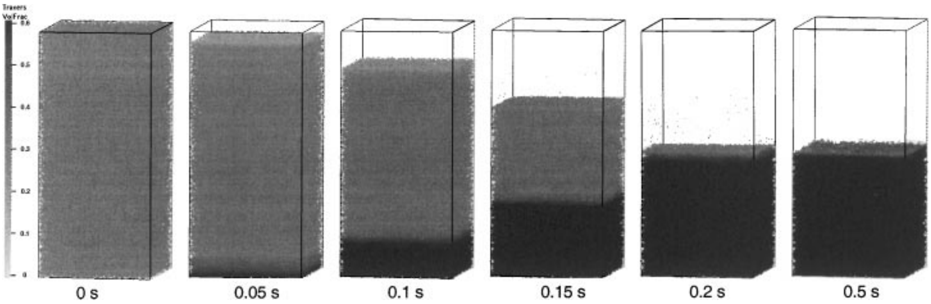


FIG. 7. Particle distribution at times during gravity-dominated sedimentation. Particles are colored based on local particle volume fraction.

D. *A Three-Dimensional Jet of Particles*

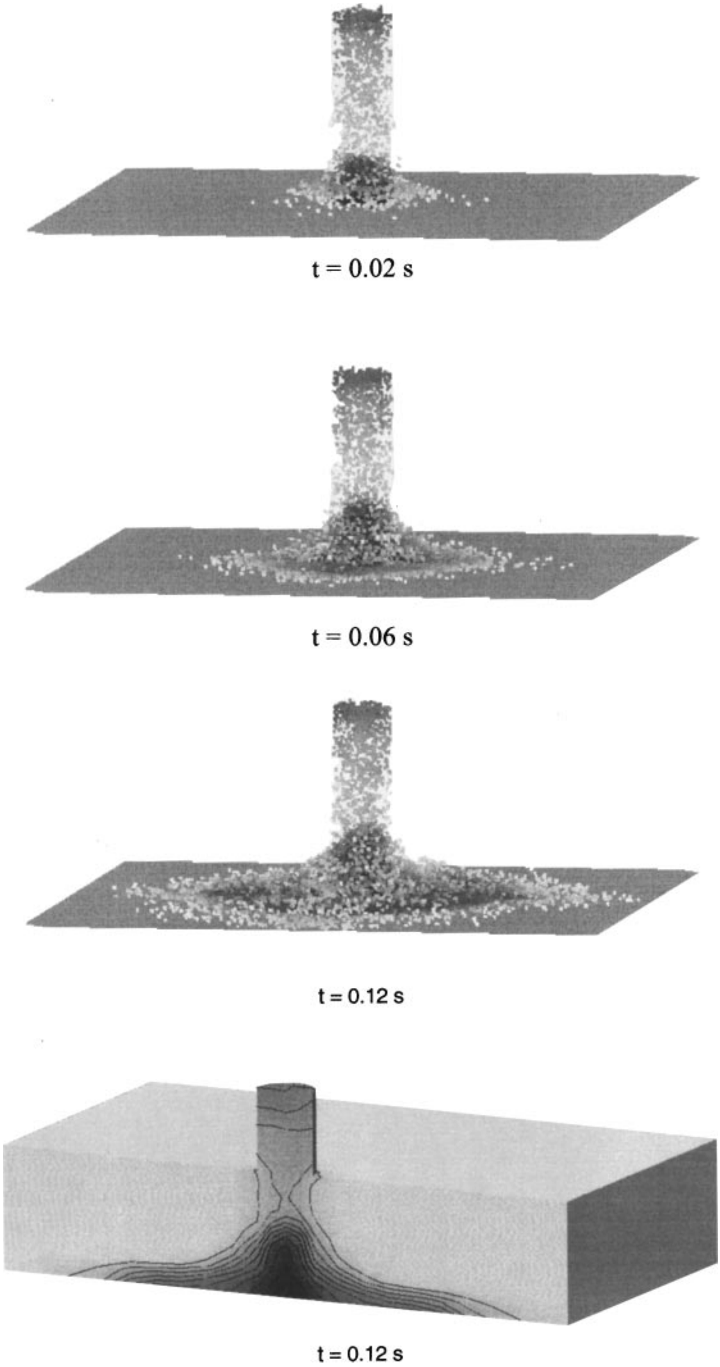
In the first two test problems, the particle normal stress is not a big player. The robust solution method for a dynamic problem where particles are jammed to close pack is demonstrated in the jet test problem. A jet of particles from a 1.5-cm tube is directed onto a flat plate at high velocity. The inlet blowtube pressure boundary is 10 kPa, and the boundary conditions at the periphery of the calculation domain except for the bottom flat plate are 0 kPa. The gas velocity in the blowtube is on the order of 25 m/s. Particles are fed at a particle volume fraction of 0.3. Details of the calculation are given in Table IV.

Figure 8 shows particles impacting the plate. The particles hit the plate and spread circularly from the point of impact. As the vertically traveling particles slow and turn out horizontally, the particles pack to near close pack under the jet. The color of particles corresponds to the particle volume fraction. Black is near close pack, and Fig. 8 shows the dark core of close pack particles under the jet. Figure 9 shows the time history of the volume fraction at the wall under the jet. The particle volume fraction quickly approaches close pack and then stays slightly below the close pack value. There is a small high frequency

TABLE IV
Particle Jet Calculation

Number of parcels (at $t = 0.8$s)	70,797
Particle radius	74 to 180 μm
Particle density	2760 kg/m^3
Fluid density	1 kg/m^3
Fluid viscosity	0.02 kg/ms
P_s	100 Pa
β	3
θ_{cp}	0.5
Gravity	0, 0, -9.8 m/s^2
Number cells	24 × 24 × 14
Domain	27 × 27 × 17 cm
Blow tube diameter	1.5 cm
Particle feed particle volume fraction	0.3
Feed pressure	10 kPa
Boundary pressure	0 kPa
Calculation time step	0.00005 s

Note. Particle sizes fed in a Gaussian distribution about the mean radius.



t = 0.02 s

t = 0.06 s

t = 0.12 s

t = 0.12 s

FIG. 8. Particle jet impacting a flat plate. Particle color is based on local volume fraction.

fluctuation riding on the particle volume fraction. The ability to calculate close pack in this highly dynamic problem of particles jamming onto a wall illustrates the robustness of the calculation method. The presented model and its associated algorithm for the particle normal stress force provides excellent performance.

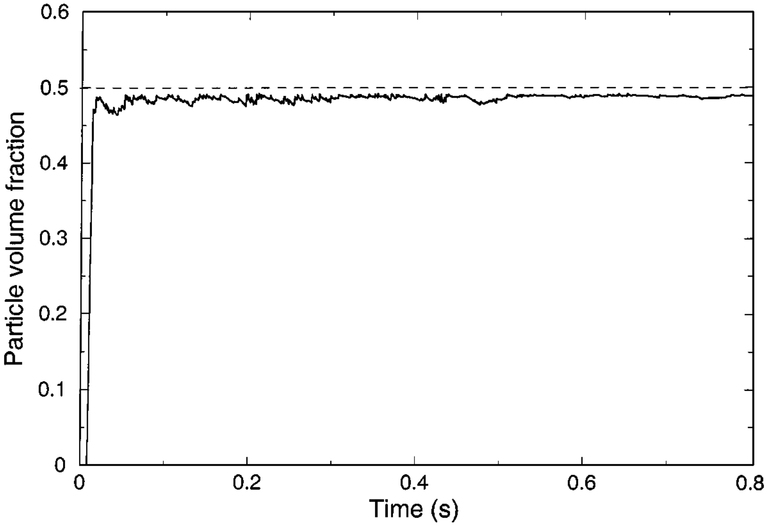


FIG. 9. The particle volume fraction at the wall under the jet.

VII. CONCLUDING REMARKS

The MP-PIC Eulerian–Lagrangian numerical scheme for particle flow has been extended to three dimensions. While not discussed in this paper, the three-dimensional solution addresses the problem of gridless particle flow in intricate geometries specified by CAD drawings. The numerical scheme has been applied and can solve a wide variety of problems in complex geometries.

This paper defines and formalizes interpolation operators and their properties which provide compact support, are both locally and globally conservative, and are suitable for three-dimensional calculations. These operators are essential to robust and fast solution methods.

A subgrid model was presented for modeling the particle normal stress applied to discrete particles. The algorithm implementing the model is fast and robust. Using the new particle normal stress model eliminates implicit solutions or fractional time step methods previously required by the highly nonlinear, particle normal stress. The calculated results show excellent performance from the model.

APPENDIX A

The interpolation operator defined above with the property given by Eq. (19) is conservative. This is shown by mapping the particle property m_p to grid node ζ, which gives the grid particle property

$$M_{p\zeta} = \sum_{\kappa=1}^{N_p} S_\zeta\left(\mathbf{x}_{p_\kappa}\right) m_{p_\kappa}, \tag{a1}$$

where the total number of particles is N_p.

The grid property, $M_{p\zeta}$, mapped back to particle position $\mathbf{x}_{p\kappa}$ is

$$m_{p_\kappa} = \sum_{\xi=1}^{N} S_\xi \left(\mathbf{x}_{p_\kappa}\right) M_{p\xi}, \tag{a2}$$

where N is the number of nodes, which includes all nodes supporting the interpolation operators.

Putting (a1) into (a2) and reordering the summations gives

$$m_{p_\kappa} = \sum_{\kappa=1}^{N_p} m_{p_\kappa} \sum_{\xi=1}^{N} S_\zeta \left(\mathbf{x}_{p_\kappa}\right) S_\xi \left(\mathbf{x}_{p_\kappa}\right). \tag{a3}$$

Using the definition of the product of interpolation operators, Eq. (19),

$$m_{p_\kappa} = m_{p_\kappa} \sum_{\xi=1}^{N} S_\xi \left(\mathbf{x}_{p_\kappa}\right), \tag{a4}$$

and because $\sum_{\xi=1}^{N} S_\xi (\mathbf{x}_{p_\kappa}) = 1$, the particle property, m_{p_κ}, for the particle at $\mathbf{x}_{p_\kappa}$ is recovered. The mapping of the particle property to the grid and the subsequent mapping of the property to a particle is conservative.

Now examine the mapping of a grid property to particles and the remapping of particle properties back to the grid. Start with the mapping of grid property, M_{p_ζ}, to particle positions $\mathbf{x}_{p_\kappa}$ for all particles as given by (a2). The particle property is interpolated back to the grid using (a1). Putting (a2) into (a1) gives

$$\Psi_{p\zeta} = \sum_{\kappa=1}^{N_p} \sum_{\xi=1}^{N} S_\zeta \left(\mathbf{x}_{p_\kappa}\right) S_\xi \left(\mathbf{x}_{p_\kappa}\right) M_{p\xi}. \tag{a5}$$

The product of the interpolation operators is defined by Eq. (19). Only one combination of the products is not zero, and the summation over nodes reduces to a single value.

$$\Psi_{p\zeta} = \sum_{\kappa=1}^{N_p} S_\zeta \left(\mathbf{x}_{p_\kappa}\right) M_{p\zeta}. \tag{a6}$$

The grid property, $M_{p\zeta}$, does not depend on particle position and can be removed from the summation, giving

$$\Psi_{p\zeta} = M_{p\zeta} \sum_{\kappa=1}^{N_p} S_\zeta \left(\mathbf{x}_{p_\kappa}\right). \tag{a7}$$

The portion of the grid property $M_{p\zeta}$, originally mapped to a particle positions, is mapped back from particles to grid node ζ. The process is conservative.

REFERENCES

1. D. Gidaspow, Hydrodynamics of fluidization and heat transfer supercomputer modeling, *Appl. Mech. Rev.* **39**, 1 (1986).
2. G. K. Batchelor, A new theory of the instability of a uniform fluidized bed, *J. Fluid Mech.* **193**, 75 (1988).
3. A. A. Amsden, P. J. O'Rourke, and T. D. Butler, *KIVA-II: A Computer Program for Chemically Reactive Flows with Sprays*, LA-11560-MS (Los Alamos National Laboratories, Los Alamos, NM, 1989).
4. D. Gidaspow, *Multiphase Flow and Fluidization Continuum and Kinetic Theory Description*, (Academic Press, Boston, 1994).
5. M. A. Risk, Mathematical modeling of densely loaded, particle laden turbulent flows, *At. Sprays* **3**, 1 (1993).
6. P. J. O'Rourke, *Collective Drop Effects on Vaporizing Liquid Sprays*, Ph.D. thesis, (Princeton University, 1981).
7. F. H. Harlow and A. A. Amsden, *Fluid Dynamics, A LASL Monograph*, LA-4700 (Los Alamos National Laboratories, Los Alamos, NM, 1971).
8. P. J. O'Rourke and A. A. Amsden, On particle-grid interpolation and calculating chemistry in particle-in-cell methods, *J. Comput. Phys.* **109**, 37 (1983).
9. M. J. Andrews and P. J. O'Rourke, The multiphase particle-in-cell (MP-PIC) method for dense particle flow, *Int. J. Multiphase Flow* **22**, 379 (1996).
10. D. M. Snider, P. J. O'Rourke, and M. J. Andrews, *An Incompressible Two-Dimensional Multiphase Particle-in-Cell Model for Dense Particle Flows*, NM, LA-13280-MS (Los Alamos National Laboratories, Los Alamos, NM, 1997).
11. F. A. Williams, *Combustion Theory, 2nd ed.* (Benjamin–Cummings, Menlo Park, CA, 1985).
12. D. M. Snider, P. J. O'Rourke, and M. J. Andrews, Sediment flow in inclined vessels calculated using multiphase particle-in-cell model for dense particle flow, *Int. J. Multiphase Flow* **24**, 1359 (1998).
13. S. E. Harris and D. G. Crighton, Solutions, solitary waves and voidage disturbances in gas-fluidized beds, *J. Fluid Mech.* **266**, 243 (1994).
14. F. M. Auzerais, R. Jackson, and W. B. Russel, The resolution of shocks and the effects of compressible sediments in transient settling, *J. Fluid Mech.* **195**, 437 (1988).
15. J. T. Jenkins and S. B. Savage, A theory for the rapid flow of identical smooth, nearly elastic spherical particles, *J. Fluid Mech.* **130**, 67 (1983).
16. C. K. K. Lun, S. B. Savage, D. J. Jeffrey, and N. Chepurniy, Kinetic theories for granular flow: Inelastic particles in Couette flow and singly inelastic particles in a general flow field, *J. Fluid Mech.* **140**, 223 (1984).
17. J. Ding and D Gidaspow, A bubbling fluidization model using kinetic theory of granular flow, *AIChE J.* **36**, 523 (1990).
18. J. H. Ferziger and M. Peric', *Computational Methods for Fluid Dynamics* (Springer-Verlag, Berlin/Heidelberg, 1996).
19. D. A. Anderson, J. C. Tannehill, and R. H. Pletcher, *Computational Fluid Mechanics and Heat Transfer* (Hemisphere, New York, 1984).
20. P. J. O'Rourke and A. A. Amsden, *Implementation of a Conjugate Residual Iteration in the KIVA Computer Program*, Los Alamos Report LA-10849-MS (Los Alamos National Laboratories, Los Alamos, NM, 1986).
21. D. L. Youngs, Numerical simulation of turbulent mixing by Rayleigh-Taylor instability, *Physica* **12D**, 32 (1984).
22. D. L. Youngs, Modeling turbulent mixing by Rayleigh-Taylor instabilities, *Physica* **37D**, 270 (1989).
23. P. F. Linden, J. M. Redondo, and C. P. Caulfield, Molecular mixing in Rayleigh-Taylor instability, in *Advances in Compressible Turbulent Mixing*, edited by W. P. Dannevik, A. C. Buckingham, and C. E. Leith, (1992), p. 95.
24. D. M. Snider and M. J. Andrews, The structure of shear driven mixing with an unstable thermal stratification, *J. Fluids Eng.* **118**, 55 (1996).