Macroscopic Particle Model

Tracking Big Particles in CFD

Madhusuden Agrawal[1], André Bakker
Fluent Inc., 10 Cavendish Ct., Lebanon, NH 03766, USA

Michael T. Prinkey
Fluent Inc., 3647 Collins Ferry Road, Suite A,
Morgantown, WV 26505, USA

Presented at the:
AIChE 2004 Annual Meeting
Particle Technology Forum - Paper 268b
November 7-12, Austin, Texas, USA

Abstract: Typical particle tracking models used in CFD software assume that particles are point masses that do not interact. Large particles immersed in the fluid flow cannot be modeled using this type of approach. The modeling of big (macroscopic) particles requires special treatment to take into account effects such as the blockage of fluid volume, the proper evolution of the drag force and torque experienced by the particles, particle-particle as well as particle-wall collisions, and friction dynamics. This paper discusses a novel approach to model such situations in the context of a regular finite volume CFD solver. This new macroscopic particle model (MPM) models particles that cover multiple grid cells with six degrees of freedom, including both translation and rotation. The MPM has many industrial applications, especially in the pharmaceutical, chemical, materials handling, and sports industries. A wide variety of validations has been performed. Tests have shown that large numbers of particles can be handled easily on typical desktop computers.

Keywords: particle tracking, CFD, particle dynamics, fluid flow, big particles

[1] Corresponding author; e-mail: mxa@fluent.com

Introduction

Particle tracking models in most commercial CFD software, such as the discrete phase model (DPM) in FLUENT, assume that particles are point masses that do not interact. Big particles immersed in the fluid flow cannot be modeled using this type of approach. We consider particles to be big when they are larger than the computational grid cells in the domain. The modeling of big (macroscopic) particles requires special treatment to take into account effects such as the blockage of fluid volume, the proper evolution of the drag force and torque experienced by the particles, particle-particle collisions as well as particle-wall collisions, and friction dynamics.

To account for these effects, a macroscopic particle model (MPM) has been developed using a new, pragmatic approach that is especially suitable for CFD solvers based on the finite volume method. Although this model is implemented in FLUENT 6 using user-defined functions (UDFs) and a customized graphical user interface (GUI), the approach itself is not limited to FLUENT and is usable in other CFD software also.

In this article, we first will discuss the theory behind this approach. We then will briefly describe the implementation in FLUENT, see also Agrawal *et al.* (2003). We will conclude with typical examples of the types of flows for which this model is suited.

Theory

In the MPM approach, particles are treated in a Lagrangian frame of reference. Each particle is assumed to span several computational cells. Cells that contain at least one node within the region occupied by the particle are considered as being “touched” by the particle. At every time-step, a solid body velocity that describes the particle motion is patched in these cells as shown in Figure 1. The particles have six degrees of freedom: both translational and rotational velocity components are taken into account.

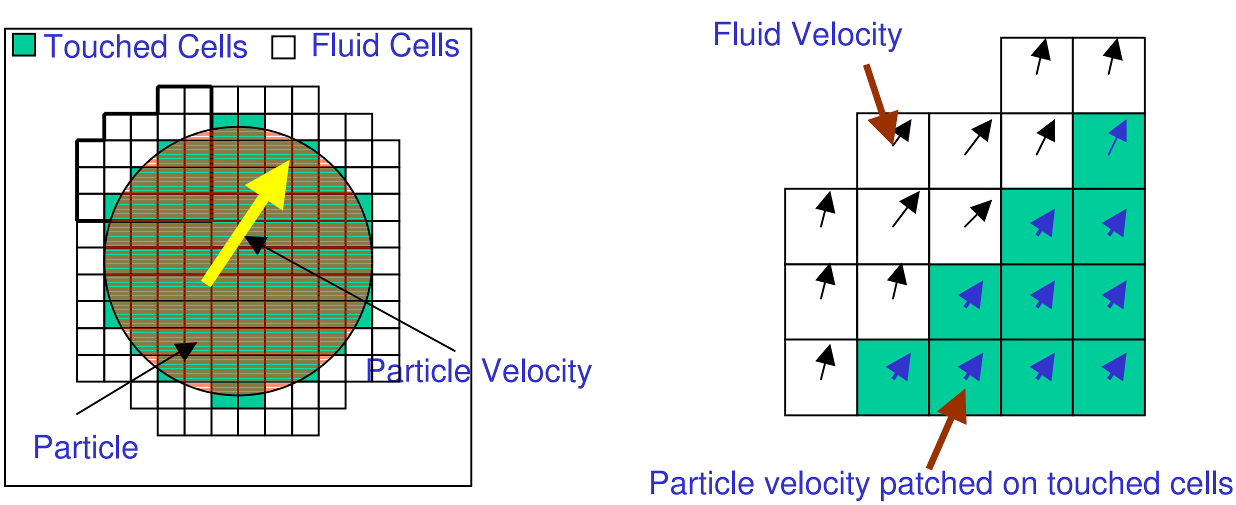

Figure 1. Fluid cells touched by particle and patching of particle velocity.

By patching the rigid body motion of the particle, momentum effectively is added to the fluid. The integral of the momentum change, linear as well as angular, gives the drag force and torque experienced by each particle. These forces are used to compute the new velocities and positions of the particles at the next timestep. Additional forces, such as body forces, also can be included in the model. The drag force on the particle is calculated as follows:

$$Drag = \left(\sum_{\substack{touched \\ cells}} m_f \left(\bar{V}_f - \bar{V}_p \right) \alpha_p \right) \frac{1}{\Delta t} \qquad \ldots(1)$$

The torque experienced by the particle is obtained as follows:

$$Torque = \left(\sum_{\substack{touched \\ cells}} m_f \left(\bar{V}_f - \bar{V}_p \right) . \bar{r}\, \alpha_p \right) \frac{1}{\Delta t} \qquad \ldots(2)$$

Here, m_f is fluid mass, $\bar{V}_f$ is the fluid velocity, $\bar{V}_p$ is the particle velocity, $\bar{r}$ is the radius vector from the fluid cell center to the particle center and α_p is particle volume fraction. For the fluid cells that are completely within the particle volume, the particle volume fraction is one.

For partially filled cells, the particle volume fraction is calculated based on the effective cell nodes inside the particle volume. Note that when particles are continuously injected into the domain, a mass source term, based on the total volume of fluid displaced per unit time by the injections, is included in the model to maintain an overall mass balance.

To detect a particle-wall collision, the model identifies the boundary faces (wall surfaces) that the particle intersected during the previous time-step, if any. If a collision with a wall is detected, the incoming particle velocity is projected onto the normal and tangential components of the reflected particle velocity, applying the coefficient of restitution and friction factor, as appropriate. Rolling friction for particles rolling along a wall surface also can be included in the friction factor. In the same way, the model detects particle-particle collisions, and applies the principle of conservation of momentum to obtain the final velocities of both particles.

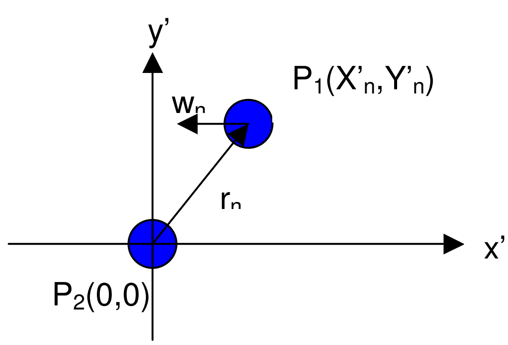

Figure 2. Coordinate system for the relative motion of two particles.

Absence of particle acceleration within a time step leads to constant particle velocities before collision and allows collision detection from purely kinematic and geometric consideration. Figure 2 depicts a pair of particles at time t_n from known positions and with known velocities. It is convenient to describe the relative particle motion in a frame of reference, (x',y'), with its origin on one of the two approaching particles and its x'-axis anti-parallel to their relative velocity $\overline{w}_n$:

$$\overline{w}_n = \overline{V}_{1n} - \overline{V}_{2n} \quad \ldots(3)$$

Here $\overline{V}_{1n}$ and $\overline{V}_{2n}$ are the velocities in the laboratory reference frame of particle-1 and particle-2 in the n^{th} time step. The particle separation vector $\overline{r}_n$ is defined by:

$$\overline{r}_n = \overline{X}_{1n} - \overline{X}_{2n} \quad \ldots(4)$$

Where $\overline{X}_{1n}$ and $\overline{X}_{2n}$ are the coordinates of the particle-1 and particle-2 at the end of n^{th} time step in the laboratory frame.

If the motion of the two particles was allowed to continue uninterrupted, the variation of their separation vector can be expressed analytically in terms of their constant relative velocity as:

$$\overline{r}(t) = \overline{r}_n + (t - t_n)\overline{w}_n \quad \ldots(5)$$

Therefore, the particle separation distance would vary as:

$$r(t) = \sqrt{r_n^2 + 2\left[\overline{r}_n \cdot \overline{w}_n\right](t - t_n) + w_n^2 (t - t_n)^2} \quad \ldots(6)$$

Contacts between two neighboring particles within a time step are detected by determining whether their minimum separation distance becomes less than or equal to the sum of their radii. For pairs that collide, the exact moment of contact, t_c, is determined by solving the following equation:

$$r(t_c) = d_{12} = \frac{d_1 + d_2}{2} \quad \ldots(7)$$

The coordinates of the centers of the particles at the instant of collision are determined from:

$$X_c = \overline{X}_n + (t_c - t_n)\overline{V}_n \quad \ldots(8)$$

A similar algorithm is used to detect particle-wall collisions. The effect of wall motion also has been taken into account in particle-wall collision detection.

In many particle-particle systems, field forces besides gravity are important. For example, particles might agglomerate because of electrostatic or magnetic forces. Such forces are implemented using a potential force model, in which inter-particle field forces are given by:

$$F_i = \sum_j \frac{G_p M_i^{n1} M_j^{n2}}{R_{i-j}^{n3}} \qquad \ldots(9)$$

Here F_i is the force on particle i resulting from field forces exerted by all other particles; M_i is the particle mass; R_{i-j} is the inter-particle distance; G_p, n_1, n_2, and n_3 are model constants specified by the user for each individual particle describing the system. The sign of the model constant G_p determines if the field force is an attraction or a repulsion force. Particle-wall field forces are similarly given by:

$$F_{iw} = \sum_{walls} \frac{G_w M_i^{n4}}{R_i^{n5}} \qquad \ldots(10)$$

Here F_{iw} is the field force on particle i from all walls; M_i is the particle mass; R_i is the particle-wall distance; G_w, n_4, and n_5 are model constants specified by the user.

Implementation

A customized Graphical User Interface (GUI) was created using the SCHEME programming language to define all user inputs for the macroscopic particle model. Particle properties for each particle stream, such as initial position, initial velocity, particle density and particle radius, can be defined through easy to use GUI panels or can be read from a formatted ASCII file. Figure 3 shows snapshots of some of these GUI panels.

The post processing for the macroscopic particle model has been coupled with FLUENT's discrete phase model visualization tools, which allow particles to be displayed as shaded spheres with a defined radius. This post processing mostly has been automated with few user inputs required. Transient particle data also can be saved in a Fieldview data file in binary format. At each time step, the position of all particles also can be stored in an ASCII file in comma separated variable (CSV) format, which can be viewed in EXCEL or other standard text file readers.

The macroscopic particle model has been programmed in such a way that it works well with the FLUENT parallel solver. A Binary Space Partitioning (BSP) Tree algorithm has been used to enhance the speed to search cells and faces, which are in the neighborhood of particle coordinates. The tests have shown that a large number of interacting particles (10,000 and more) can be handled easily without the need for excessive computation time.

Other main features of the macroscopic particle model as implemented in FLUENT 6 are summarized in Table 1.

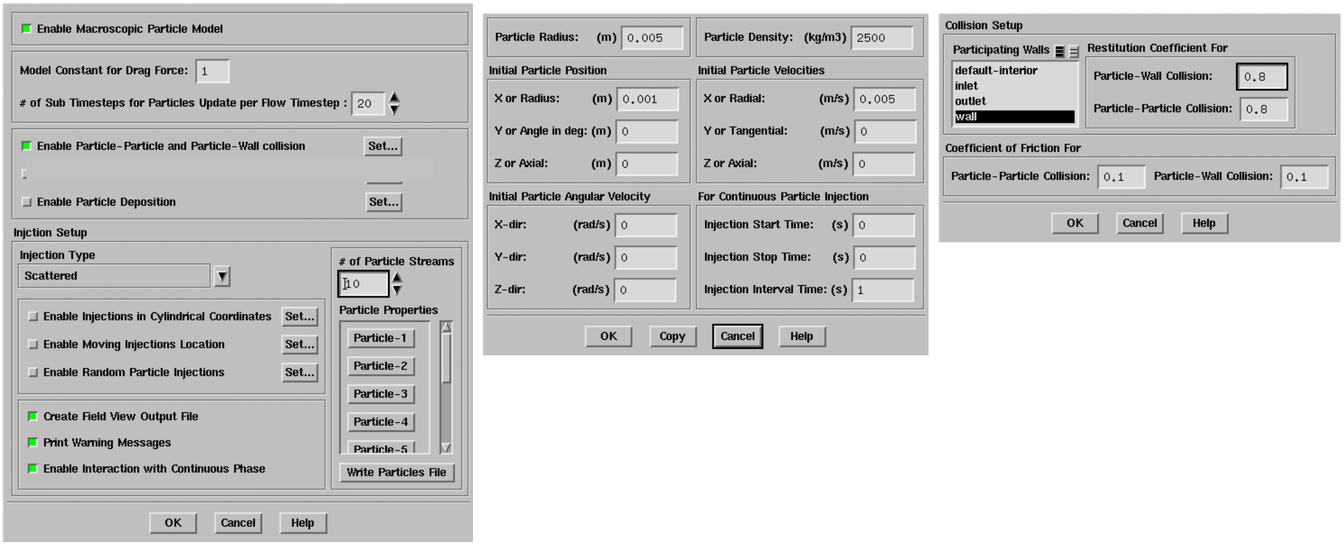

Figure 3. Snapshots of typical GUI panels.

Table 1. Other features of the macroscopic particle model.

Continuous particle injection	User can define start time, stop time and interval time for each injection. The macroscopic particle model will inject new particles at regular intervals from the start time to the stop time.
Particle injection in a cylindrical coordinate system	User can define initial particle position and velocity in a cylindrical coordinate system.
Random particle injection	With this option, particles will be injected from random locations between specified lower and upper coordinates.
Moving/rotating injection location	User can define rotational and/or translational velocity for the injection location. The model calculates a new injection location based on these velocities at each time step and releases particles from new positions. This feature allows the injection to be at a location that is fixed relative to a moving object, such as a rotating fan blade.
Particle deposition and buildup on defined surfaces	User can define wall zones for deposition. Any particle that collides with these wall zones or with any previously deposited particle will be deposited.
Particle interaction with continuous phase	User can have either one way coupling or two way coupling of particle motion with fluid flow.

Results

The MPM has many industrial applications, especially in the pharmaceutical, chemical, materials handling, and sports industries. A wide variety of test cases have been performed using the macroscopic particle model. These test cases validate the robustness, convergence properties and compatibility of this model with other physics models (i.e. sliding mesh, moving deforming mesh, multiphase, laminar flow, turbulence, etc). These test cases also validate individual features of the model (i.e. friction, collision, continuous injection, moving injection location, post processing, etc). Figures 4 through 11 depict plots of typical test cases performed using the macroscopic particle model in various applications.

Figure 4 shows motion of five particles in a box. Particles are of a different radius but of the same density. Air enters from one side of the box and leaves from the diagonally opposite side. The first image shows the velocity vectors on a middle vertical plane in the absence of any particle in the domain. It also shows initial position of the particles. The second image shows velocity vectors and particle position after some time. Flow blockage due to the presence of the macroscopic particles clearly can be noticed in this figure.

Figure 5 shows the motion of five balls and the resulting velocity field on sloping surfaces. The balls are of the same radius but of different mass. The red ball is the heaviest one and the blue ball is the lightest one. Balls are falling and rolling under gravity and generate the expected flow field. This test qualitatively validates the implementation of rolling friction.

Figure 6 shows the motion of balls falling through a hopper. Balls of two different radii continuously are injected and falling through a hopper under gravity. No airflow field is solved in this test example and only calculations for particle dynamics are performed.

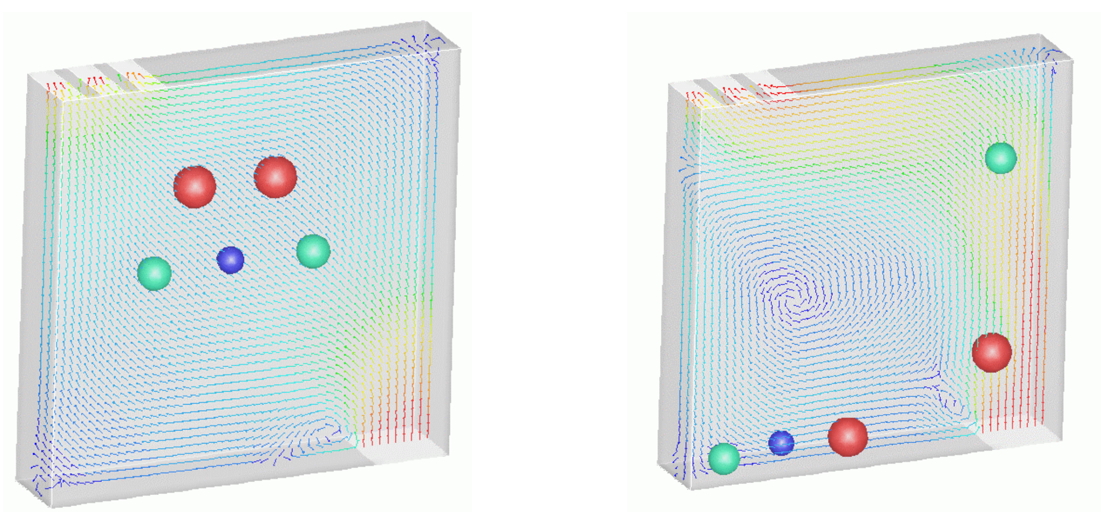

Figure 4. Particle motion in a box. Airflow is from bottom right to top left.

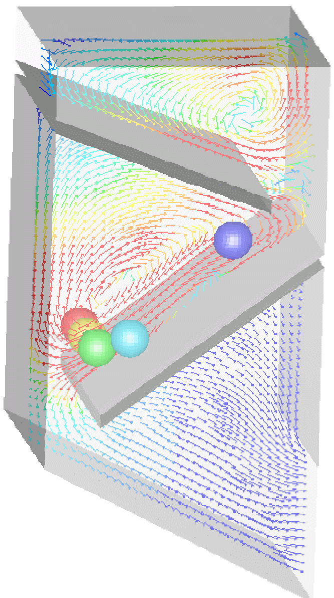

Figure 5. Particles rolling on slopes.

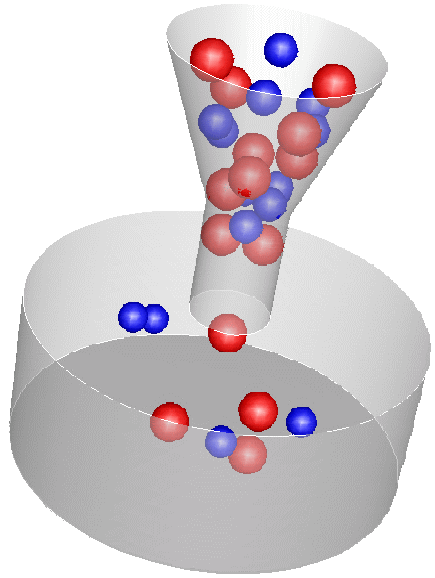

Figure 6. Particles falling through a hopper.

Figure 7 demonstrates the compatibility of the macroscopic particle model with FLUENT's Volume of Fluid (VOF) multiphase model. A spinning, heavy ball is dropped in the stationary water; the ball disrupts the water surface, which results in splashing. The figure shows the free surface shape and the velocity vectors at the free surface.

Figure 8 demonstrates the interaction of big particles with a rotating paddle. Two different instances in time are shown. The paddle rotates counter-clockwise as seen from the top of the vessel.

Figure 9 shows the motion of two different particle sizes flowing through a filter element. One set of particles has a size smaller than that of the filter opening, while the other set of particles is bigger. Smaller particles can pass through the filter element while the filter will block the bigger particles. This is a very effective way to model particle separation through a filter. The gravity effect, which allows particles to settle in the bottom section of the pipe, was considered in this example.

Figure 10 demonstrates the ability to model a large number of interacting particles (10,000 and more). The test mimics a fluidized bed simulation using the MPM. This simulation was performed on a four-processor computer in less than a day. The first picture shows fluidizing particles, and the second picture shows velocity vectors on a middle vertical plane.

Figure 11 shows a pool table simulation using the macroscopic particle model. The picture on the left shows path lines of the airflow (including recirculation) generated by the cue ball as it rolls towards the rack on a pool table, and the picture on the right shows path lines just after cue ball strikes the rack and disperses the balls.

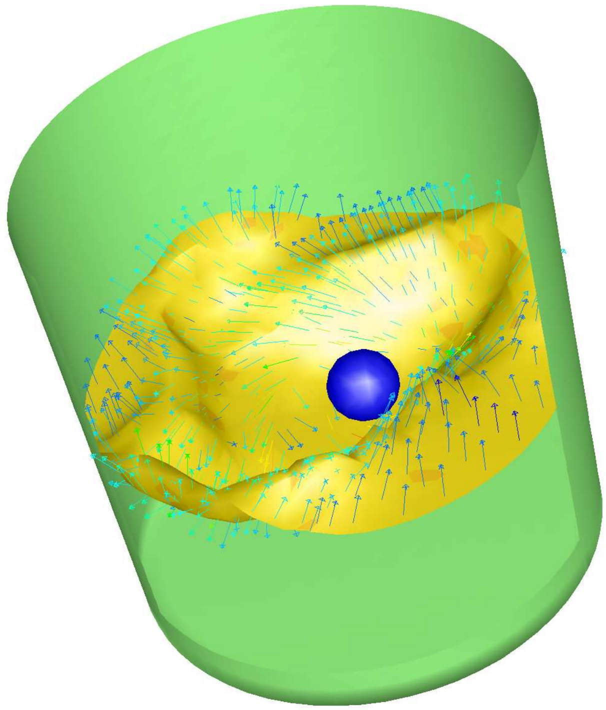

Figure 7. Water splashing due to particle motion.

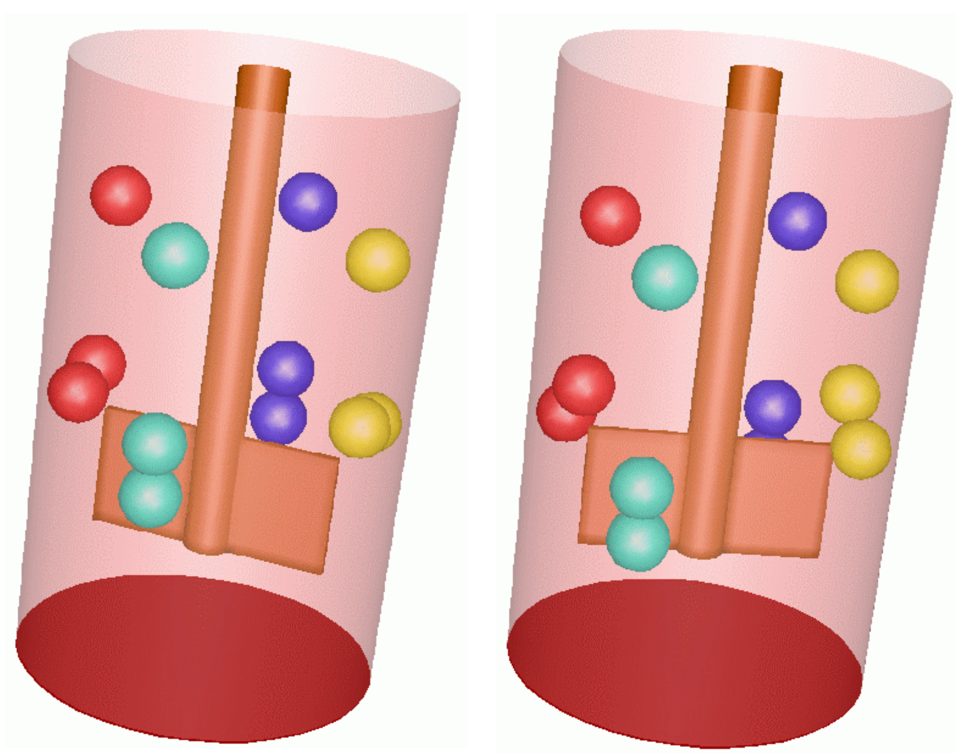

Figure 8. Interaction of falling big particles with a rotating paddle. The two lower cyan particles are pushed out of the way by the blade.

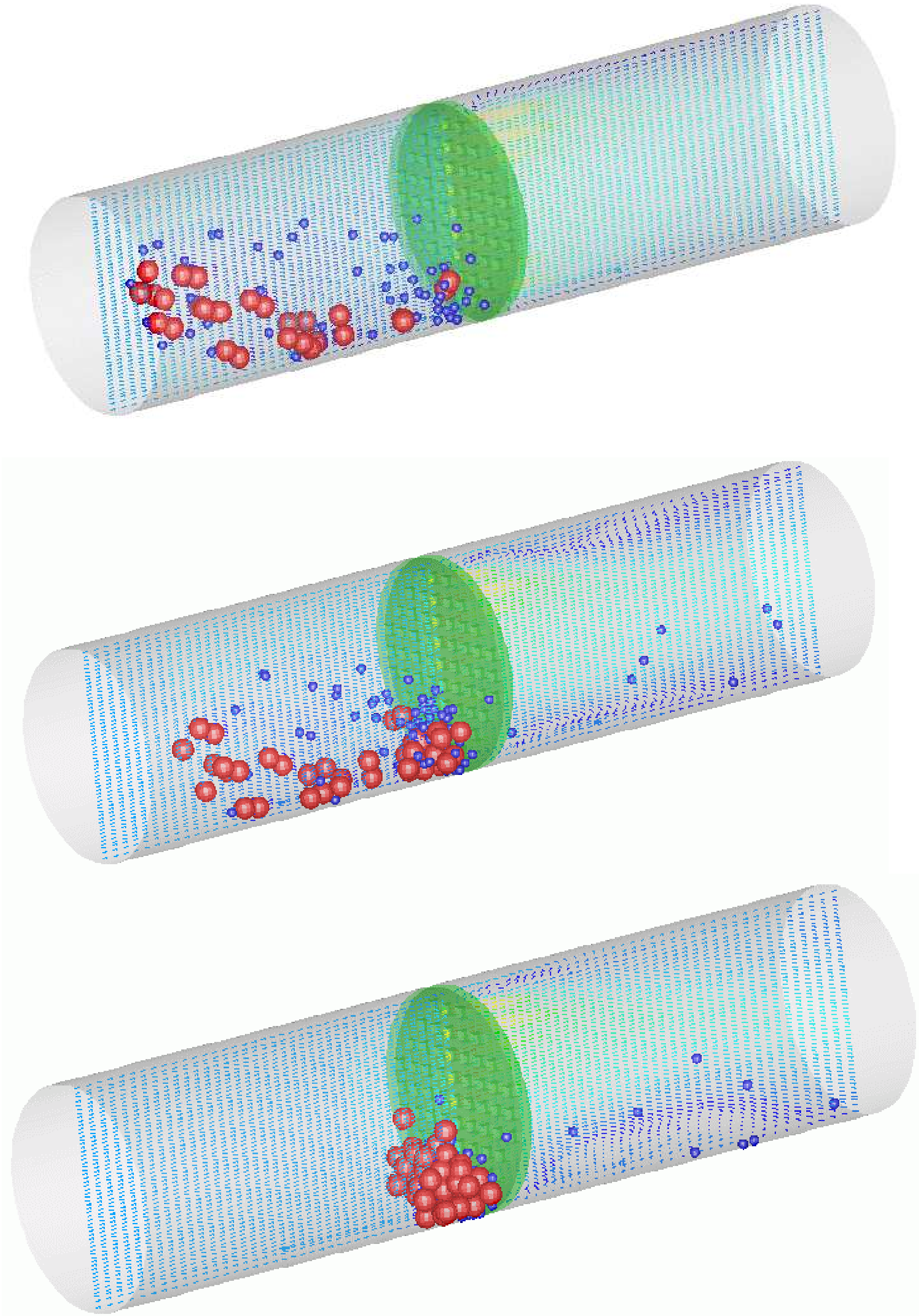

Figure 9. Particle separation through a filter element at three instances in time. The flow is from left to right. The small particles flow through the holes in the perforated plate and exit the pipe on the right. The plate blocks the bigger particles.

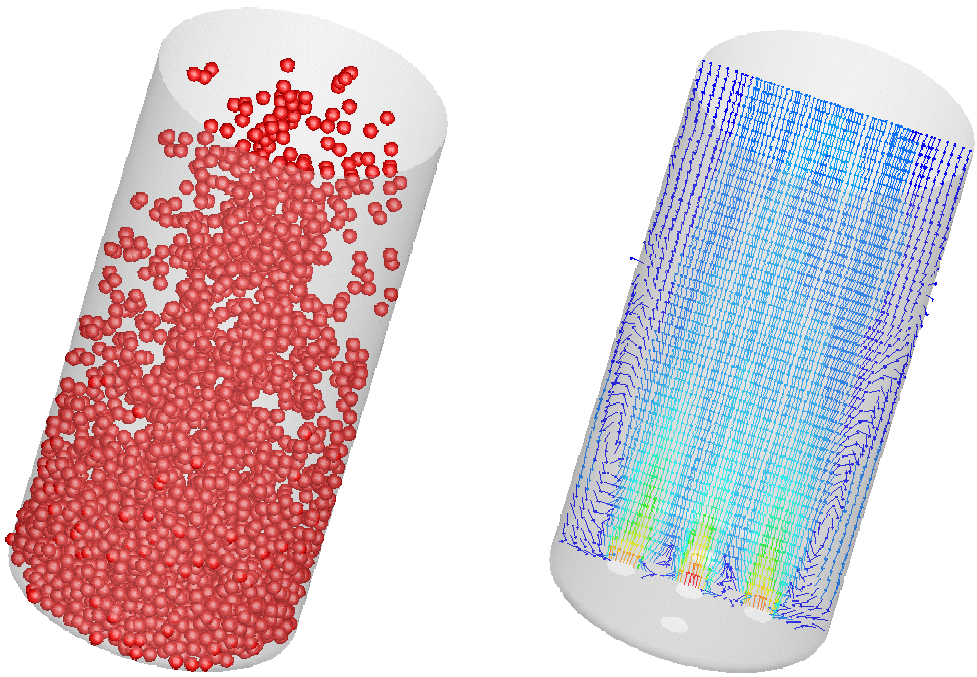

Figure 10. A large number of particles in a fluidized bed application.

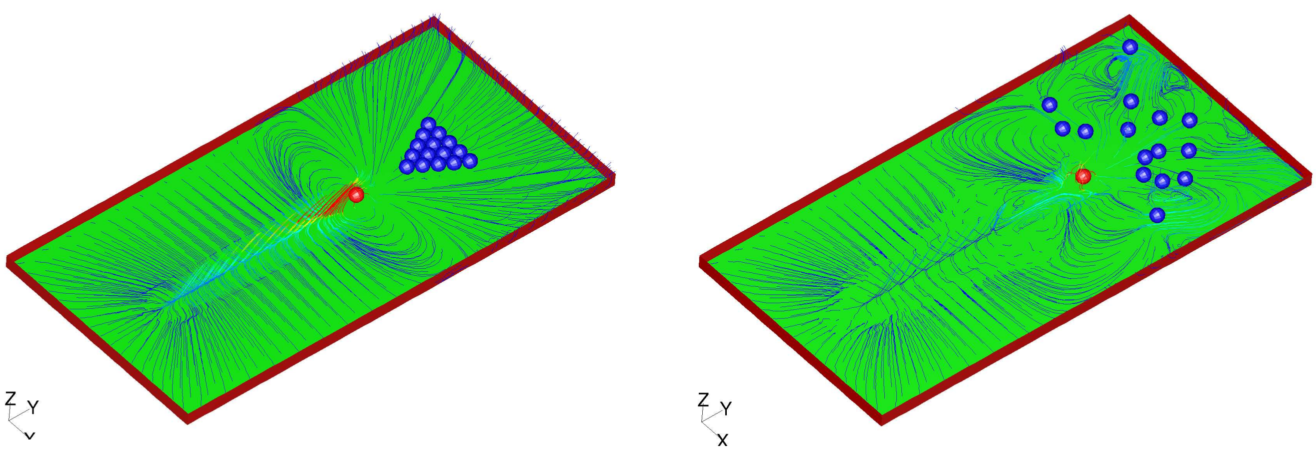

Figure 11. A pool table simulation using the macroscopic particle model. The streamlines indicate the airflow induced by the balls near the pool table.

Conclusion

The macroscopic particle model is an innovative approach to model big spherical particles for situations where traditional discrete phase models are not applicable. It was implemented in FLUENT, but in principle the method is suited for any a finite volume CFD solver.

This model has many industrial applications. A wide variety of test cases has demonstrated the robustness of this model and provided qualitative validations. Detailed quantitative validations are in progress. Additional improvements and enhancements in the current macroscopic particle model are being developed. The present formulation can be extended from spherical shaped particles to other regular shapes (cylindrical, ellipsoid and brick). Heat and mass transfer and chemical reactions between two particles, as well as between particle and fluid, also can be taken into account. Particle breakup and agglomeration also can be included in the MPM.

Acknowledgment

The authors wish to thank Birendra Kumar David and Joe Maruszewski for their contributions to the implementation of this model.

References

Agrawal M., Bakker A., Prinkey M.T. (2003)
Tracking Big Particles, Fluent Newsletter, Fall 2003, page 11.

Symbols

D_1	Diameter of particle-1
D_2	Diameter of particle-2
D_{12}	Sum of radius of two particles
F_i , F_{iw}	Field forces on particle i
G_p , G_w	Field force model constants
m_f	Fluid mass
M_i	Particle mass
n_1 , n_2 , n_3 , n_4 , n_5	Field force model constants
$\bar{r}$	Radius vector from fluid cell center to particle center
r(t)	Particle separation distance at time t
$r(t_c)$	Minimum separation distance to detect collision
$\bar{r}_n$	Particle separation vector
R_i , R_{i-j}	Particle distances
Δt	Time step
t_n	Flow time at n^{th} time step
T	Current flow time
$\bar{X}_{1n}$	Particle-1 coordinates
$\bar{X}_{2n}$	Particle-2 coordinates
$\bar{X}_c$	Coordinates of particle at the instant of collision
$\bar{X}_n$	Coordinates of particle at n^{th} time step
$\bar{V}_n$	Particle velocity at n^{th} time step
$\bar{V}_f$	Fluid velocity
$\bar{V}_p$	Particle velocity
$\bar{V}_{1n}$	Velocity vector for particle-1
$\bar{V}_{2n}$	Velocity vector for particle-2
$\bar{w}_n$	Relative velocity vector between two particles
α_p	Particle volume fraction