

PARTICLE-FLUID COUPLING

C.W. Hirt
Flow Science, Inc.
June 1999

BACKGROUND

There are many important flow situations involving a dispersed liquid or solid material in a continuous gas or liquid. A few everyday examples are fuel injection in a automobile engine, rain, dust storms and all sorts of atomizers for painting, cleaning and applying medicine.

In most cases there are significant interactions between the continuous and dispersed materials that arise because of the drag experienced by the dispersed particles as they move through the continuous fluid. A secondary interaction effect is the displacement of fluid volume by particle volume.

Of the two interaction effects, i.e., volume exclusion and momentum exchange, the most important is momentum exchange because this can be significant even when the volume of particles is small. To see why this is so, consider a typical two-phase system with properties similar to that of water and air. The density ratio between water and air is about 1000. This means that a liquid fraction of only $10E-4$ translates into 10% of the mixture mass. Thus, even when the water volume is a negligible fraction of the whole, it still may account for a significant portion of the mixture momentum.

To have a computationally practical momentum exchange between a fluid and discrete mass particles, the equations describing each phase must be solved simultaneously. Of course, there are limiting cases in which a simultaneous solution is simple to obtain. For instance, if the drag resistance of the fluid is negligible, then there is no exchange of momentum between the phases and the motion of each component can be evaluated independent of the other component.

On the other hand, if the drag between particles and fluid is so great that they always move with the same speed, then it is possible to treat the fluid-particle system as a fluid mixture having a variable density.

Many computational models that incorporate both a continuous fluid and discrete particle model assume that the particles do not affect the fluid dynamics. This assumption is often made because it is easy to compute the fluid motion and then, separately, move particles.

Adding a drag resistance to particles as they move through a fluid is an easy thing to do. If the average particle mass density is small compared to the fluid density, then the loss of particle momentum by this mechanism is small and does not have to be given back to the fluid.

With increasing particle density, the momentum exchange cannot be neglected. If the momentum change of the particles remains small compared to the momentum of the fluid, then it

can be explicitly added to the fluid. This is a relatively easy thing to do, however, it's not always possible to keep the momentum exchange small, and this means the explicit coupling method is subject to unexpected failure.

For the case of liquid particles in a gas, for example, the explicit method is not very useful because of the large difference in density between the phases. In this situation it is necessary to use a numerical method that implicitly couples the momentum of the particles and fluid together.

The purpose of this Technical Note is to describe such an implicit scheme within the framework of the **FLOW-3D**[®] computational fluid dynamics program. Several examples are presented that illustrate how the new capability can be used for different types of particle-fluid systems.

A PARTICLE-FLUID COUPLING APPROACH

We shall not consider particle volume effects. This feature could be added at a later time if it appears necessary and/or desirable. The addition would involve adding a source-sink volume contribution to grid cells receiving-losing a particle. All the sources-sinks coming from particles moving from one cell to another would be accumulated during the particle-moving step and then used in the fluid continuity equation during the following time step.

Our interest in this Note is on momentum coupling. One of the numerical difficulties in accounting for momentum exchanges with discrete particles is that the particles are localized at points, and there may be many of them in a given element of fluid. Another problem is associated with the formulation of the momentum exchange as a drag resistance experienced by the particles as they flow through a surrounding fluid. Drag forces are proportional to either the relative velocity or the square of the relative velocity between the phases. The latter case involves a non-linear velocity relationship that complicates the mathematics of coupling.

This proposal addresses these difficulties in the context of the current version of the **FLOW-3D**[®] program. In this program particles may have individual masses, which are computed to move under the action of forces that include body forces (e.g., gravity), viscous and form drag, and buoyancy forces computed from the local pressure gradient. Particles may bounce or stick to rigid surfaces according to a coefficient of restitution and are transmitted or reflected from porous surfaces with a probability proportional to the fraction of open area. In addition, mass particles can move in both void (i.e., gas) or liquid regions.

The momentum change in the fluid resulting from the interaction with a particle is expressed as a drag coefficient multiplying the relative velocity between the fluid and the particle. If the momentum exchange is proportional to the square of the relative velocity, then we first linearize it, which then makes the effective drag coefficient a function of the relative velocity.

Overview of Basic Approach

Consider the momentum equations for a small control volume (i.e., grid cell) and a single particle that appears in that volume. The particle momentum equation is used to advance the particle velocity from time t to $t+\delta t$. The new velocity can be expressed in terms of all known

quantities except for the fluid, which appears in the coupling term. Next, this velocity can be algebraically substituted into the corresponding coupling term in the fluid momentum equation to produce an expression for the new fluid velocity. A similar substitution can be done for every other particle that influences the fluid element. Because the fluid velocities that appear in the drag expressions are linear, it is a trivial matter to collect all the occurrences of the new velocity together in order to obtain a solution for it in terms of the remaining known quantities. Once this has been done for the entire grid, and a solution for the new velocities is obtained, they can then be used to compute the drag force needed for updating the particle velocities.

This, then, is the strategy: (1) compute new particle velocities in terms of the as yet unknown fluid velocities, (2) substitute those velocities into the momentum exchange terms in the fluid equations and solve for the new fluid velocities, finally, (3) return to the particles and update their velocities to reflect the new fluid velocities.

To make it work we must exercise a little foresight in the selection of what particles affect what fluid elements. We must also be a little careful in how we formulate the relative-velocity drag expression. The choices made were selected to make the most efficient use of capabilities already incorporated in *FLOW-3D*[®].

Details of the Coupling Procedure

As previously mentioned, the momentum exchange is expressed as a force that is equal to a drag coefficient times the relative velocity between the phases. This expression can be interpreted in another way, in which the fluid and particles may be thought of as forming a composite fluid. In particular, the drag coefficient times the fluid velocity can be thought of as a mass added to the fluid, while the drag coefficient times the particle velocity can be interpreted as momentum added to the fluid. Exactly how this works out can be illustrated by considering the coupling with a single particle.

The basic idea we will implement can be described in terms of a single velocity component, say U , for a typical grid (fluid) control volume. Further, we shall assume there is one discrete particle located in the control volume moving with velocity u_p (the addition of more particles is easy once the coupling concept is understood).

Discrete momentum equations for U and u_p can be written in the form:

$$\begin{aligned} M_f U^{n+1} &= M_f \tilde{U} + m_p \delta t k (u_p^{n+1} - U^{n+1}) \\ m_p u_p^{n+1} &= m_p \tilde{u}_p - m_p \delta t k (u_p^{n+1} - U^{n+1}) \end{aligned} \quad (1)$$

In these equations M_f and m_p are the mass of the s in the control volume and particle mass respectively. The superscript $n+1$ refers to the new time level values that are being computed as an advancement from step n to $n+1$ with step size δt . A tilde over the velocities in the first term on the right hand sides of the equations indicates an estimate for the $n+1$ level velocities that contain all forces except for particle drag. These tilde velocities are computed explicitly in terms of known, time level n quantities. If there is no drag resistance, the tilde and $n+1$ level quantities are identical, and no further computations are necessary.

To solve Eq.1 simultaneously we begin by first expressing u_p^{n+1} in terms of U^{n+1} , then we can substitute it into the equation for U^{n+1} ,

$$u_p^{n+1} = (1 - \omega)\tilde{u}_p + \omega U^{n+1}, \text{ where } \omega = \frac{\delta tk}{1.0 + \delta tk} \quad (2)$$

Then the U^{n+1} equation can be rewritten as,

$$\begin{aligned} M_f U^{n+1} &= M_f \tilde{U} + m_p \omega (\tilde{u}_p - U^{n+1}), \text{ or} \\ U^{n+1} &= \frac{M_f \tilde{U} + m_p \omega \tilde{u}_p}{M_f + m_p \omega} \end{aligned} \quad (3)$$

When expressed this way, we can see very clearly how the particles are coupled with the fluid. The coupling efficiency is expressed by the factor ω . If the drag coefficient k is zero, indicating no coupling, then ω is zero and Eq.3 reduces to the identity,

$$U^{n+1} = \tilde{U}.$$

The alternative limit of infinitely strong drag, in which the particles and fluid move together, has the coupling value $\omega=1$ and Eq.3 then becomes,

$$(M_g + m_p)U^{n+1} = M_g \tilde{U} + m_p \tilde{u}.$$

This result is exactly what would be expected for a homogeneous mixture. The mass on the left side is the sum of the fluid and particle mass. On the right side of the equation there is the sum of the momentum of the fluid and particle at time step $n+1$, which contains contributions from all the forces acting on both phases. In effect, the weighting factor ω measures that portion of the particle mass and momentum that is tied to the fluid, while the remaining $(1-\omega)$ portion is free.

Adding more particles is now easy. All one must do is add an additional term to the right side of the first equation in Eq.3 for the momentum exchange with each particle.

If the fluid is incompressible, we use the Marker-And-Cell (MAC) scheme to insure incompressibility. This means that the pressure gradient acceleration contained in U -tilde must be updated to have time level $n+1$ pressures, which are computed from the continuity equation. The only change that particle coupling will make to the present scheme used in **FLOW-3D**[®] is that the denominator in Eq.3 must contain contributions from all the particles in the control volume.

Now we can summarize the implicit coupling scheme. At the start of a cycle the particle accelerations are computed explicitly except for the drag term. At the same time the ω factors are computed and stored for later use. Next the fluid momentum equations are solved with the inclusion of the particle drag effects as indicated by Eq.3. At the end of the cycle the particles

are updated to account for the drag, which can now be computed using the newly computed fluid velocities.

With this program there can be any amount of momentum exchange between the particles and the fluid. The extra computational effort needed to implicitly couple the two phases is small and should not significantly increase computational times.

Further Details

The addition of particle drag contributions to the fluid momentum equation has two parts. One contribution contributes to a D'Arcy like drag, while the other is simply an additional momentum source term. For D'Arcy drag contributions *FLOW-3D*[®] uses a cell centered quantity, DRG(ijk), that is defined in terms of the drag coefficient k of the cell as,

$$DRG = \frac{1.0}{1.0 + \delta t \cdot k}.$$

To make use of the existing DRG formulation in *FLOW-3D*[®] we must define the fluid velocity used in the drag term as a mass-weighted average. That is, for the x-direction velocity $U = (M_r U_r + M_l U_l) / (2M_c)$, where the subscripts in the numerator refer to the right and left sides of a grid cell and M_c is the mass of the cell. This deviates from the marker particle scheme in the code, which interpolates the edge velocities to the actual location of the particle within the cell.

This change in definition of fluid velocity at a particle is necessary to be consistent with the use of DRG. If we insisted on using the interpolated velocity, it would then be necessary to compute and store equivalent DRG values for every velocity component; i.e., we would need three new arrays instead of the one existing three-dimensional array (DRG).

In *FLOW-3D*[®] when a value of DRG is needed at a cell boundary location, it is computed as a reciprocal average of the cell centered values of DRG lying on either side of the boundary. Because DRG is itself a reciprocal, this is equivalent to using a simple average of the drag coefficients (k) in the neighboring cells. This fact, along with the use of average fluid velocities allows us to separate the drag contributions associated with a cell into two parts that can be separately assigned to the control volumes for cell edge velocities. For example, for the x direction

$$K \cdot (M_r U_r + M_l U_l) / (2M_c) = \frac{KM_r}{2M_c} U_r + \frac{KM_l}{2M_c} U_l,$$

where K is the combined drag coefficient for all the particles in the cell. The first term on the right side only involves U_r and is assigned to the control volume for U_r . The second term on the right contains U_l and is assigned to the control volume for U_l . Using this approach the D'Arcy type drag effects associated with particles are included in *FLOW-3D*[®] by simply changing the cell value of DRG to include contributions to the drag coefficient from all the particles in the cell.

The addition of momentum to the fluid from particles is easy to include in the program using existing force arrays (fsigx, fsigy, fsigz) that are also used for thermocapillary forces or user defined force contributions.

Using this modification of the drag and force terms it is possible to show that the exchange of momentum between particles and fluid is a conservative process in which the momentum lost by the particles is gained by the fluid and vice versa, see the Appendix for more details.

Programming Comments

By making use of existing capabilities in the program we have minimized the code changes needed to implicitly couple particles with a surrounding fluid. Implementation of this model requires that we add a new subroutine (PARVEL) to the beginning of a calculation cycle that evaluates particle drag coefficients and accelerations and adds their contributions to the fluid drag function DRG and fluid forces FSIGX, FSIGY, and FSIGZ in the cell containing the particle.

After new fluid velocities have been computed subroutine PARMOV is called, which recomputes the particle velocities using the new fluid velocities and finally moves the particles to new positions using those velocities.

The implicit particle option has required that routine PARMOV be completely re-written so that particles are moved to new locations after their velocities are computed rather than before. It has also been necessary to relocate the time when PARMOV is called within a time cycle.

If the implicit particle option is not requested, routine PARVEL is not called and this reduces the program back to its original mode; and no extra computational work is needed when the particles do not influence the fluid.

The new implicit option has been incorporated into *FLOW-3D*[®] without adding any new memory requirements. The model does, however, require some duplicate computations for the particles in routine PARVEL, which could be eliminated by using additional storage arrays for the particles. Our recommendation at this time is to accept the added CPU cost instead of using extra memory.

ILLUSTRATIVE APPLICATIONS

In this section we illustrate the new model's capabilities using simple examples, but covering a range of applications that includes solid/liquid particles in air and gas bubbles in a liquid. We also show how this model may be used to approximate a rigid structure moving through a fluid with a prescribed velocity.

Liquid Jet in Air

The first example involves an axisymmetric jet of water droplets injected into air. This case could just as well represent a jet of solid particles since the particles are pure mass elements, i.e., their being liquid or solid makes no difference. The particles have the density of water and a diameter of 0.01 cm. The density of the air is 0.0012 gm/cc. A point source is used for the origin of the jet particles, with a source rate of 2,000 particles/s and an initial velocity of 100.0 cm/s. The Reynolds number-dependent drag coefficient built into *FLOW-3D*[®] was used with a coefficient of 1.0. Gravity was turned off.

The computational grid consisted of 12 radial cells ($\delta x=0.083\text{cm}$) and 100 axial cells ($\delta z=0.10\text{cm}$). A constant diffusion coefficient of $0.1\text{ cm}^2/\text{s}$ was applied to the particles to represent a low to moderate level of turbulent dispersion.

Figure 1 shows the air velocities (top) and particles (bottom) at $t=0.1\text{ s}$. Note that only one symmetric half of the particles has been plotted. A few particles have moved outside the main jet and are nearly stationary (or may even be moving counter to the jet because of air entrainment effects). The maximum air velocity in the jet is 89.1 cm/s, which is about the same as the velocity of the particles in the center of the jet near its tip.

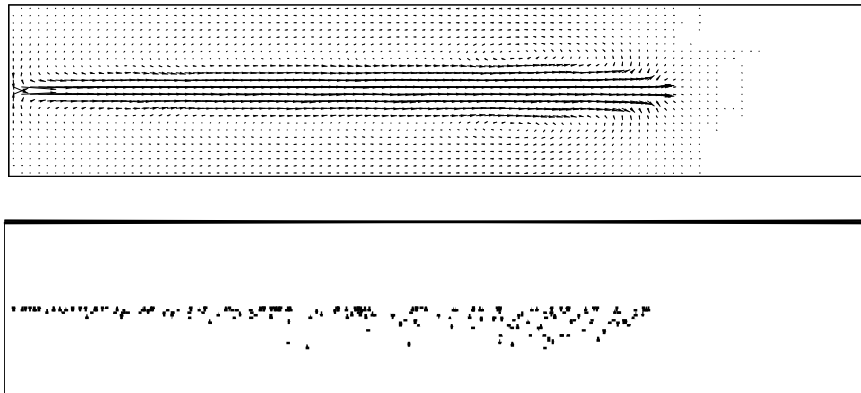


Figure 1. Axisymmetric liquid jet in air. Air velocities generated by particles (top) and one half of symmetric particles (bottom).

The results look realistic, and there is a realistic correlation between the fluid and particle velocities. Unfortunately, we do not have monodisperse spray data with which to compare.

Bubble Column in a Liquid

For comparison, and to check the model with an inverse density ratio, we next consider the case of air “particles” injected into water. For this example we located a free liquid surface 8.0 cm above the source of the gas bubbles and included gravity in the computation in order to have a buoyant force on the bubbles. Bubbles that pass through the surface are discarded.

Planer geometry was used so that the computation actually represents a cross section through a bubble curtain. A “large eddy simulation” model was used to represent turbulence generated in the water by the bubbles. The bubble particles were then assumed to diffuse with the turbulent viscosity that was computed by this model. The bubble particles of diameter 0.1cm were generated at the rate of 200/s with zero initial velocity.

A plot of the bubble curtain after 1.6s is shown in the left plot of Fig. 2. Velocities generated by the bubbles are shown in the right plot. By this time a number of bubbles have left the system by passing through the free surface. A small bulge can be seen in the free surface above the bubble column.

The bubbles in the center of the column move upward at about 22cm/s while the water velocities in this region only reach about 9.1cm/s.

Experimental observations on single bubbles of the size modeled indicate that they should rise at about 16cm/s. A simulation of a single bubble, which generates no significant fluid motion, does predict a rise speed of 16cm/s, so the computations are in good agreement with observation. The higher particle speed in the center of the bubble curtain is a result of the fluid motion generated by the bubbles.

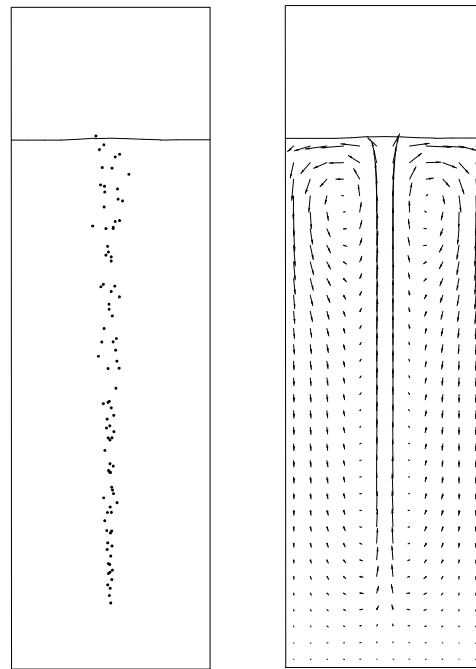


Figure 2. Bubble curtain rising to a free surface.

The amount of swelling of the free surface is dependent on the fluid velocity at the surface and should only be about 0.05cm in this example, which is about 1/5 of a grid cell dimension. The computational results are consistent with this estimate.

Settling of Solid Particles in Water

In another two-dimensional example the particles were increased in density to 1.5 times that of water. A two-dimensional container, 3.0cm wide and 10.0cm high, was filled with water to a height of 8.0cm. An initial block of solid particles was placed in the center of the container at a height of 7.4cm and allowed to settle to the bottom. To make things more interesting a circular obstacle was inserted in the water so the particles would have to flow around it. The particles had a diameter of 0.1cm and the initial particle block was a square of 0.2cm on a side containing 20x20 particles in a staggered configuration. A small, constant diffusion coefficient of 0.01 cm²/s was assigned to the particles.

Figure 3 shows the initial particle configuration (left) and what it evolves to by 2.0s (center). A few particles appear to penetrate the obstacle, but this is only a result of the limited grid resolution, which is 12x40 cells. There is only a 4x4 grid of cells defining the obstacle.

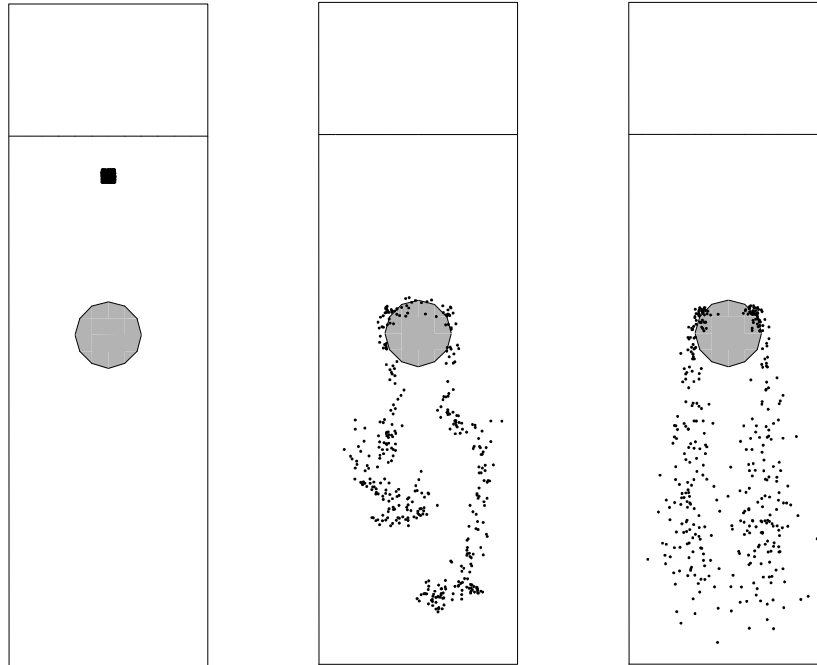


Figure 3. Settling of a block of heavy particles in water. Initial configuration (left) and after 2.0s (middle). The right plot shows a corresponding result at 12.0s when particles do not influence the water.

Flow eddies in the wake of the obstacle cause a snaking pattern in the sinking particles. For comparison, the right most frame is a repeat of the simulation in which the particles were not allowed to interact with the fluid (the fluid remains stationary in this case). The consequences of coupling the particles with the fluid is clearly evident.

Using Particles to Model Rigid Moving Bodies

The final example illustrates how the implicit coupling technique can be used to model the motion of a rigid body. In this case the body is simply a block 1.0cm square. The density of the block particles is 1000 times that of the surrounding water so that the water does not significantly affect them. Also, the drag coefficient is set to a constant 1000/s, which is a large enough value to force the fluid to move with the particles.

With these settings the block simply free falls through the water under the action of gravity. The goal of the simulation is to show that the particles move as a rigid mass and that the water outside the block is forced to flow around the block, while the water inside the block must move with the particles.

We see that this is the case in Fig. 4, which shows the computed flow at $t=0.01s$ when the block has speed 9.82cm/s and at $t=0.11s$ when the block's speed is 98.2cm/s. This example demonstrates is that it is possible to represent a rigid object as an assemblage of particles. To have a solid body it is necessary to insure the particles are placed close enough together that

there is always at least one particle per grid cell no matter where the object is located within the grid. In the general case these particles would then be given a prescribed motion instead of having it be computed as it was in this example. If a very large drag coefficient is used, the fluid must flow with or around the object as it moves.

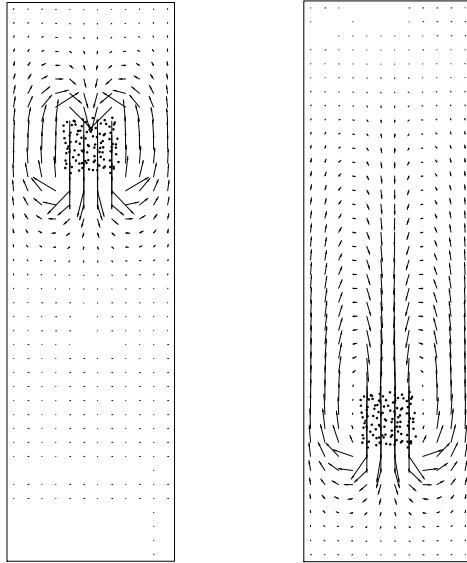


Figure 4. Example of a block of particles acting like a rigid body falling through water.

In general, when particles are used to model rigid objects that move with a specified velocity it is necessary to define the drag coefficient, say $PDRG1$, to be much larger than the reciprocal of the time-step size so that the drag will appear effectively infinite. Also, the mass of the particle should be much larger than the mass of fluid in one computational cell so that the inertia of the fluid is unimportant compared to that of the particle (i.e., the specified velocity of the particle will not be affected by the fluid).

In more general cases, for example where the rigid object is to respond dynamically to fluid forces, the mass of the particles must be chosen to represent the actual mass of the body. Furthermore, all the forces acting on the set of particles representing the object will have to be summed into net forces and moments acting on the object. The particles would then be moved in a manner defined by the translation and rotation of the object.

In this latter case it is likely that some further thought will have to be given to such details as what happens when an object passes through a free surface, and how to properly account for the mass of fluid frozen about the particles. This development will be considered at a later time.

SUMMARY

A model extension to the *FLOW-3D*[®] program has been described that implicitly couples the momentum of discrete mass particles with a continuous fluid. The extension was made with a

minimum of surgery on the program because it used two features existing in the program for other purposes.

All of the original mass particle capabilities in the program have been retained. This means that it is possible to have particles move in both gas and liquid regions and particles may have a variable density or size distribution. There are also a variety of options for how particles reflect, stick, or disappear when they encounter obstacles and free surfaces.

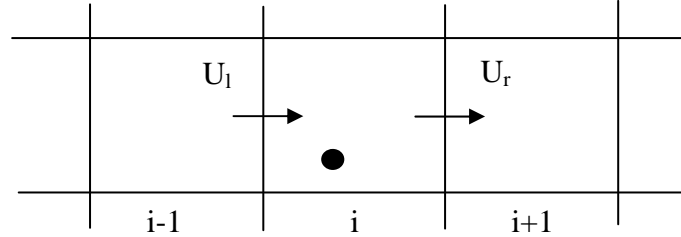
Illustrative examples have demonstrated that the new method is capable of dealing with arbitrary particle and fluid density ratios. It was also shown that particles could be used to model rigid structures that move through a fluid.

To exercise this model in *FLOW-3D*[®] after Version 7.5 it is only necessary to request the implicit particle option by setting the input variable *impprt=1*.

APPENDIX

MOMENTUM CONSERVATION IN PARTICLE-FLUID COUPLING

To demonstrate how momentum conservation is achieved in the particle-fluid coupling model, it is sufficient to consider only one particle located in cell *ijk* and only one component of momentum, say the *x* direction.



The above figure shows a sketch of the particle containing grid cell with neighbors in the *x* direction. Observe that the fluid velocity components are located at the left and right edges of the grid cell. Our goal is to show that particles can be associated with cell centered quantities and then used to adjust the edge velocities such that the momentum exchange between the particle and the fluid is conservative. We shall show here that for this to happen we must use the proper definition of an average fluid velocity at the position of the particle.

Only the force that couples the fluid particle will be explicitly expressed in the following equations. All other accelerations are assumed to be contained in velocities with a tilde written over them.

The particle momentum equation can be expressed as,

$$m_p u_p^{n+1} = m_p \tilde{u}_p + m_p \delta t k (U_f^{n+1} - u_p^{n+1}), \quad (A.1)$$

where U_f is some kind of local average of the fluid velocities located at the left and right edges of the cell. A suitable value of U_f will be determined below. The last term in this equation is the momentum change of the particle (i.e., force times the time step) caused by the fluid through the drag coefficient k . We can algebraically solve Eq.A.1 for the new particle velocity and then re-

$$f_p = m_p \delta t k (U_f^{n+1} - u_p^{n+1}) = m_p \omega (U_f^{n+1} - \tilde{u}_p) \quad (\text{A.2})$$

express the momentum exchange with the fluid, f_p as,

where,

$$\omega = \frac{\delta t k}{1.0 + \delta t k}.$$

This x direction momentum change must be shared by changes in the fluid velocity components U_l and U_r at the left and right sides of the cell containing the particle, respectively. To do this we make the following observations. The fluid masses associated with each velocity component are M_l and M_r . It is desirable to de-couple the U_l and U_r equations to simplify their solution. It is also desirable to make use of the existing formulation of the velocity update equations, which contain contributions from applied forces and flow losses proportional to the velocities. In other words, we want the particle contributions to the fluid equations to be expressed in the form,

$$\begin{aligned} M_l U_l^{n+1} &= M_l \tilde{U}_l + \frac{1}{2} F - \frac{1}{2} D M_l U_l^{n+1} \\ M_r U_r^{n+1} &= M_r \tilde{U}_r + \frac{1}{2} F - \frac{1}{2} D M_r U_r^{n+1} \end{aligned} \quad (\text{A.3})$$

where F is a cell centered x directed force in cell i and D is a cell centered drag coefficient in cell i . The $\frac{1}{2}$ factors come from averaging F and D values across cell boundaries where the fluid velocity components are located. In this case, however, there are no other F and D values because there is only one particle in cell i . Summing these expressions and comparing the result with the momentum change of the particle, we see that the following relations must hold in order

$$F = m_p \omega \tilde{u}_p, \quad \frac{1}{2} D (M_l U_l^{n+1} + M_r U_r^{n+1}) = m_p \omega U_f^{n+1} \quad (\text{A.4})$$

to conserve momentum,

Finally, we note that U_f^{n+1} must be a cell centered quantity and it must be independent of particle location within the cell. We can meet these requirements by introducing the cell fluid mass, M_{ijk} , which leads to the relations

$$U_f^{n+1} = \frac{1}{2} \frac{M_l U_l^{n+1} + M_r U_r^{n+1}}{M_{ijk}}, \quad \text{and} \quad D = \frac{m_p \omega}{M_{ijk}} \quad (\text{A.5})$$

In other words, the fluid velocity, U_f at the location of the particle that is needed in the momentum exchange term for momentum conservation is a mass-weighted velocity.

Finally, we observe that these relations show how coupling a particle to the fluid produces a pseudo fluid whose mass is equal to the fluid mass plus some portion of the particle mass, and whose momentum is the fluid momentum plus some portion of the particle momentum. The amount or portion of particle mass and momentum added to the fluid depends on the fluid-particle drag coefficient k and is given by the function ω , see Eq.A.2.

Momentum conservation in the y and z directions occurs in the same way. Also, the addition of more particles does not change the argument presented here because each particle is separately added to the cell centered values for the F and D terms.