We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

A Coupling Algorithm of Computational Fluid and Particle Dynamics (CFPD)

Albert S. Kim and Hyeon-Ju Kim

Abstract

Computational fluid dynamics (CFD) and particle hydrodynamics (PHD) have been developed almost independently. CFD is classified into Eulerian and Lagrangian. The Eulerian approach observes fluid motion at specific locations in the space, and the Lagrangian approach looks at fluid motion where the observer follows an individual fluid parcel moving through space and time. In classical mechanics, particle dynamic simulations include molecular dynamics, Brownian dynamics, dissipated particle dynamics, Stokesian dynamics, and granular dynamics (often called discrete element method). Dissipative hydrodynamic method unifies these dynamic simulation algorithms and provides a general view of how to mimic particle motion in gas and liquid. Studies on an accurate and rigorous coupling of CFD and PHD are in literature still in a growing stage. This chapter shortly reviews the past development of CFD and PHD and proposes a general algorithm to couple the two dynamic simulations without losing theoretical rigor and numerical accuracy of the coupled simulation.

Keywords: computational fluid dynamics (CFD), computational fluid and particle dynamics (CFPD), dissipative hydrodynamics, tetrahedron mesh, mesh interpolation

1. Introduction

The first simulations of a liquid were conducted at the Los Alamos National Laboratory in the early 1950s, using the Los Alamos computer, Mathematical Analyzer, Numerical Integrator, and Computer (MANIAC). This computer was developed under the direction of Nicholas Metropolis, who is the pioneer of the modern (Metropolis) Monte Carlo simulation [1, 2]. The first MC simulation was conducted using the Lennard-Jones potential to investigate the material properties of liquid argon [3]. In the MC simulations, the phase space was searched to find more probable thermodynamic states and calculate macroscopic material properties (i.e., experimentally observable) using averages of the same physical quantity over the micro-thermodynamic states. In principle, the MC method assumes that the system of interest is in a static equilibrium state, and therefore, the time evolution is replaced by the phase-space averaging. The Boltzmann factor is used as a transition probability between two energy states.

The first CFD paper was published by Hess and Smith [4]. Their method is known as panel method as the surface was discretized with many panels. More accurate CFD work of advanced panel method can be found elsewhere [5, 6]. In general, CFD research is categorized by Eulerian and Lagrangian approaches, which are grid-dependent and meshfree, respectively. In the Eulerian category, the CFD performance regarding numerical accuracy and computational speed depends on how to discretize the computational domain. Popular methods include finite volume method (FVM) [7], finite element method (FEM), finite difference method (FDM), spectral element method (SEM), boundary element (BEM), and high-resolution discretization schemes.

The deterministic molecular dynamic (MD) method treats a particle as a point mass, which has a finite mass and other physical properties but does not have any volume and shape. Given external and interparticle forces, positions and velocities in linear motion are predicted using Newton's second law for N particles. When the motion of solute molecules in the solvent fluid is of concern, then the deterministic forces form solvent to solute molecules can be mathematically replaced by random fluctuating forces. These random forces have a zero mean over time, and its magnitude is determined by the dissipation fluctuation theorem [8].

Langevin's equation includes the hydrodynamic drag balanced by the random forces due to thermal fluctuation [9]. Solving N body Langevin equation is called Brownian dynamics (BD) [10]. Although BD can include effects of particle sizes in the dynamic simulations, it is fundamentally limited to the low concentration of solutes due to the Oseen tensor (see Appendix for details). Brownian dynamics (BD) was initially developed to reduce computational load by replacing deterministic interactions between a solute molecule and many solvent molecules by randomly fluctuating, probabilistic interaction as a net driving force. BD presumes a dilute solution, which means that a mean distance between solutes is much longer than the size of the molecule. A large number of solvent molecules exist around solutes, which is enough to exert random forces due to a tremendous number of collisions. When BD is applied to the dynamic motion of multiple particles, the lack of hydrodynamic interactions may provide erroneous results because the fluctuation-dissipation principle is not quantitatively well balanced. This limitation of BD to a single particle or a dilute solution is at the equivalent level of Stokes' drag coefficient, used to calculate the particle diffusivity.

Dissipative particle dynamics (DPD) is an updated version of BD, which specifically includes the interparticle hydrodynamic forces [11, 12]. Two functions, often denoted as w_R and w_D, are proposed to quantify the presumed relationship of pair-wise hydrodynamic forces/torques, determined by the dissipation fluctuation theorem. The proposed forms of the hydrodynamic interaction are vector wise as the real viscous forces are tensor-wise. Dissipative particle dynamics (DPD) updates BD by including an approximate form of tensor-wise hydrodynamic interactions as a pair-wise vector form. A force exerted on a particle has three types such as conservative, dissipative, and random forces, and DPD satisfies the fluctuation-dissipation theorem by balancing the dissipative and random forces.

Stokesian dynamics (SD) uses the grand mobility matrix to include the far-field hydrodynamic forces/torques [13]. In-depth comparative analyses of present particle dynamic methods can be found elsewhere [14]. This mobility matrix is inverted and updated by adding differences between near-field lubrication forces and far-field two-body interactions. As the lubrication force is proportional to the logarithm of the surface-to-surface distance between two particles, it diverges during events of particle collisions. Besides, if two particles are close to each other, i.e., the surface-to-surface distance is much smaller than the particle diameter, the SD

formalism of near-field hydrodynamic contribution to the global hydrodynamic resistance (i.e., mobility inverted) often overestimates the real hydrodynamic repulsion forces. One of the primary reasons is that the lubrication theory assumes that particle surfaces are perfectly smooth, while in reality particle surfaces are rough and therefore surface friction also plays an essential role during collision events. Stokesian dynamics includes the most accurate estimation of the hydrodynamic tensors using the grand mobility matrix and pair-wise lubrication interactions. The fluctuation-dissipation theorem is better satisfied in SD by using many-body hydrodynamic interactions. In SD simulations, the fluid flow is given at any point in the computational domain as a combination of the unidirectional velocity, vortex velocity, and the rate of strain that is traceless and symmetric. The feedback interaction between a particle and fluid is already included in the expansion of Faxen rule. Due to the logarithmic characteristics of the lubrication forces, if two particles touch each other, they experience infinite repulsive forces. This characteristic of the lubrication force indicates the hard-sphere behavior of the colliding particles conceptually, implicitly assuming a perfectly elastic interparticle collision i.e., $\varepsilon = 1$ where ε is the coefficient of restitution.

Dissipative hydrodynamics (DHD) overcomes significant limitations of the particle dynamic method discussed above. DHD is a generalized method of which special cases converge to MD, BD, DPD, and SD by turning on or off specific force mechanisms. Details of DHD can be found elsewhere [14–16]. CFD and PHD were developed and applied without strong mutual influences. Particle tracking method can be viewed as a reasonable way to investigate the hydrodynamic motion of particles under the influence of ambient fluid flow. But, it has several fundamental limitations by neglecting particle density, particle shapes and sizes, and particle-fluid interactions. More importantly, the basic two-body interactions due to collisions, viscous flow, and electromagnetic properties are not included, and dispersion forces were dropped. In this light, the particle tracking method does not track particles but fluid elements moving along streamlines. While theories of particle hydrodynamics are not rigorously applied to engineering processes, this chapter includes a possible method to couple CFD and DHD in a seamless, robust way.

2. Dissipative hydrodynamics as unified particle dynamics

In this section, a unified view of preexisting particle dynamic method is discussed.

2.1 Overview

In the deterministic simulations, particle dynamics can be classified based on sizes of objects of interests. The purposes of particle dynamics are to provide the exact solution of a complex problem, bridging the theory and experiments. Note that the fundamental principles often provide governing equations, which were proven to be valid. It is difficult to solve a governing equation of a problem, if it has complex geometry and coupled boundary conditions. Experimental observations show natural processes using quantified information. Dissipative hydrodynamics is a generalized algorithm that unifies most of the preexisting particle dynamic simulation algorithms [16, 17].

At microscale of an order of nanometers, molecular dynamics deals with the motion of many molecules in various phases such as gas, liquid, and solid. Suppose

that a system contains N molecules in volume V at temperature T. Newton's second laws of motion for this system is

$$m_j a_j = m_j \frac{\mathrm{d}^2 r_j}{\mathrm{d}t^2} = \sum_{i=1,\, i \neq j}^{N} F_{ij} \qquad (1)$$

where F_{ij} is a force exerted on particle j of mass m_j from particle i of mass m_i at time t. Position r_i and velocity v_i of molecule i are updated from time t to $t + \delta t$, i.e., from its initial values of $r_i(t = 0) = r_i^0$ and $v_i(t = 0) = v_i^0$, respectively, using the acceleration a_i determined using Eq. (1). Numerical evolution of Eq. (1) requires a specific algorithm for double integration [18–21].

A macroscale of order of millimeters, granular dynamics often called the discrete element method (DEM) includes specifically collision rules using the restitution and friction coefficients. During inelastic collisions of nonrotating particles, the particle kinetic energy is continuously lost, and their motion is decelerated. For a collision of rotating spheres, the surface friction provides an effective torque (as action and reaction) of the same magnitude and opposite directions to two colliding spheres. Rotational motion of a non-touching particle in a fluid medium generates angular dissipation of kinetic energy. Considering granules, i.e., non-Brownian particles, in a gas phase often neglect solute molecules and approximate the system as multiple particles undergoing the gravitational force field in a vacuum phase. As granular particle mass is much higher than that of colloids or nanoparticles in an aqueous solution, in a stationary phase the gravitational force is often balanced by normal forces developed at interfaces of particles to touching neighbors or a rigid wall. Implementation of the hydrodynamic lubrication interactions to granular particles is a difficult task, which requires an in-depth understanding of microscopic surface-deformation phenomena, linked to macroscopic particle motion.

A universal simulation method that can seamlessly include forces/torques exerted on arbitrary particles is therefore of great necessity. The method, first of all, should be able to:

1. Investigate the accelerating/decelerating motion of particles.

2. Satisfy the fluctuation-dissipation theorem for Brownian particles.

3. Include many-body hydrodynamic interactions.

4. Mimic inelastic collisions between spherical particles.

5. Apply constraint forces to form a nonspherical rigid body consisting of unequal spherical particles.

Dissipative hydrodynamics (DHD) has all the features required to be a universal simulation method for particle dynamics by taking specific advantages from MD, BD, SD, DPD, and DEM. A detailed review can be found elsewhere [14].

2.2 DHD formalism

Particle relaxation time. For a single particle motion in a viscous fluid, the governing equation can be in 1D space for simplicity:

$$m\ddot{x} = -\beta v + f'(t) \qquad (2)$$

where m is the particle mass at position x, β is the drag coefficient, and f' is a random fluctuating force of zero mean:

$$\langle f'(t) \rangle = 0 \tag{3}$$

$$\langle f'(0)f'(t) \rangle = 2k_B T \beta \delta(t) \tag{4}$$

where T is the absolute temperature, k_B is Boltzmann's constant, and $\delta(t)$ is Dirac's delta function. Discarding the random force, one divides both sides of Eq. (2) by m to have

$$\frac{dv}{dt} = -\frac{v}{m/\beta} \tag{5}$$

which indicates that m/β has a dimension of time. This time scale is called the particle relaxation time, define as

$$\tau_p = \frac{m}{\beta} \tag{6}$$

which is a time scale that after a particle noticeably slows down after it starts moving with an initial velocity under the drag force. Stokes derived the drag coefficient

$$\beta = 6\pi \mu a_p \tag{7}$$

where μ is the absolute fluid viscosity and a_p is the radius of the primary particle [22].

Governing equation. A governing equation of DHD simulation is as follows:

$$\boldsymbol{M} \cdot d\boldsymbol{v} = [\boldsymbol{Q}^p - \boldsymbol{R} \cdot (\boldsymbol{v} - \boldsymbol{U})]dt + \boldsymbol{B} \cdot d\boldsymbol{W} \tag{8}$$

where $\boldsymbol{M}$ is a diagonal matrix of mass and moment of inertia; $\boldsymbol{v}$ and $\boldsymbol{U}$ are translational/rotational velocities of the particle and the fluid, respectively; $\boldsymbol{Q}^p$ is the generalized interparticle and conservative force/torque vector; $\boldsymbol{R}$ is the grand resistance matrix; and $\boldsymbol{B}$ is the Brownian matrix of zero mean and finite variance:

$$\langle \boldsymbol{B} \rangle = 0 \quad \text{and} \quad \langle \boldsymbol{B}^{\text{tr}} \cdot \boldsymbol{B} \rangle = 2k_B T \boldsymbol{R} \tag{9}$$

where $\delta(t)$ is the Dirac-Delta function. And, $d\boldsymbol{W}$ is the Ito-Wiener process [23, 24] having the following mathematical properties: $W_k = 0$ at $t = 0$, $W_k(t)$ is continuous, $dW_k (\equiv W_k(t + \delta t) - W_k(t))$ follows the normal distribution, and finally $d\boldsymbol{W} \cdot d\boldsymbol{W} = dt$. The Brownian matrix $\boldsymbol{B}$ can be calculated by decomposing the grand resistance matrix such as

$$\boldsymbol{R} = \boldsymbol{A}^{\text{tr}} \cdot \boldsymbol{I} \cdot \boldsymbol{A} \tag{10}$$

where $\boldsymbol{A}$ is the decomposed matrix to be obtained and $\boldsymbol{I}$ is the identity matrix. Statistically, the the identity matrix can be expressed as

$$\boldsymbol{I} = \langle \boldsymbol{C}^{\text{tr}} \cdot \boldsymbol{C} \rangle \tag{11}$$

where $\boldsymbol{C}$ is a vector with zero mean and unit variance, i.e., $\langle \boldsymbol{C} \rangle = 0$ and $\langle \boldsymbol{C}^2 \rangle = 1$, respectively. The Brownian matrix is defined as

$$B = \sqrt{2k_B T} C \cdot A \tag{12}$$

and substituted into (10) to provide

$$R = (A^{\mathrm{tr}} \cdot C^{\mathrm{tr}}) \cdot (C \cdot A) = (C \cdot A)^{\mathrm{tr}} \cdot (C \cdot A) = \frac{B^{\mathrm{tr}} \cdot B}{2k_B T} \tag{13}$$

Therefore, B is obtained by calculating a square root of R matrix, which is equal to A matrix of Eq. (10). The identity relationship of Eq. (11) is not satisfied at specific time t but statistically by taking an average of $C^{\mathrm{tr}} \cdot C$ for a much longer period than the particle relaxation time τ_p. The effective force acting on a particle of a swarm of many particles in a viscous fluid is then represented as

$$Q^p - R \cdot (v - U) + \sqrt{2k_B T} C \cdot A \cdot W' \tag{14}$$

where $W' = \mathrm{d}W/\mathrm{d}t$. Although A is deterministically calculated to satisfy Eq. (10), $C^{\mathrm{tr}} \cdot C = \mathrm{I}$ is not valid at an instance but statistically. In the same sense, Eq. (11) is satisfied statistically.

Hydrodynamic tensors In Eq. (14), the generalized force requires a calculation of the grand resistance matrix R, which will allow to generate A. Consider particle i among N_p particles in a given volume V, translating with a linear velocity v_i and rotating with an angular velocity ω_i at an instantaneous position $r_i(t)$. In the absence of particles, the fluid flow at the center of particle i can be represented as $U^\infty(r_i)$. At a point $r = (x, y, z) \in S_i$ on surface S_i of particle i from the particle center r_i, the flow field is described as

$$V(r) = U^\infty(r_i) + \Omega^\infty \times (r - r_i) + E^\infty : (r - r_i) \tag{15}$$

where U^∞ is the unidirectional velocity, Ω^∞ is the vorticity,

$$\Omega^\infty = \frac{1}{2} \nabla \times V \tag{16}$$

and E^∞ is the rate of strain

$$E_{ij}^\infty = \frac{1}{2} \left(\frac{\partial V_i}{\partial x_j} + \frac{\partial V_j}{\partial x_i} \right) \tag{17}$$

which are evaluated at r_i. Because the rate-of-strain matrix is symmetric and traceless, the original nine components are reduced to

$$E_1^\infty = E_{xx}^\infty - E_{zz}^\infty = \frac{\partial V_x}{\partial x} - \frac{\partial V_z}{\partial z} \tag{18}$$

$$E_2^\infty = 2E_{xy}^\infty = \frac{\partial V_x}{\partial y} + \frac{\partial V_y}{\partial x} \tag{19}$$

$$E_3^\infty = 2E_{xz}^\infty = \frac{\partial V_x}{\partial z} + \frac{\partial V_z}{\partial x} \tag{20}$$

$$E_4^\infty = 2E_{yz}^\infty = \frac{\partial V_y}{\partial z} + \frac{\partial V_z}{\partial y} \tag{21}$$

$$E_5^\infty = E_{yy}^\infty - E_{zz}^\infty = \frac{\partial V_y}{\partial y} - \frac{\partial V_z}{\partial z} \tag{22}$$

The disturbance velocity field at the particle surface S_i is

$$v_i^D(r) = u_i - U^\infty + (\omega_i - \Omega^\infty) \times (r - r_i) - E^\infty : (r - r_i), \quad r \in S_i \qquad (23)$$

where u_i and ω_i are the translational and angular velocities of particle i, respectively. The translational/angular velocities and the rate of strain of particle i relative to the ambient flow field have then 11 degrees of freedom such as

$$
\begin{aligned}
(u_i - U^\infty, \omega_i - \Omega^\infty, -E^\infty) &= \left(v_{ix} - V_x^\infty, v_{iy} - V_y^\infty, v_{iy} - V_y^\infty\right) \\
&+ \left(\omega_{ix} - \Omega_x^\infty, \omega_{iy} - \omega_y^\infty, \omega_{iz} - \Omega_z^\infty\right) \\
&+ \left(-E_1^\infty, -E_2^\infty, -E_3^\infty, -E_4^\infty, -E_5^\infty\right)
\end{aligned}
\qquad (24)
$$

For non-Brownian particles, the governing Eq. (8) is reduced back to that of Stokesian dynamics, which is Langevin's equation with the constant drag coefficient β, replaced by the grand resistant matrix R.

When particle j is moving with linear and angular velocities of u_j and ω_j under the influences of the ambient flow field characterized using U^∞, Ω^∞, and E^∞, it experiences the hydrodynamic force F^H and torque T^H. The stresslet S^H can be obtained but does not directly contribute to the particle acceleration. The generalized velocity and force are related through the grand mobility matrix μ^∞. Here, we use q, $\dot{q}$, and $\tilde{F}$ for generalized coordinates, velocities, and forces, respectively:

$$q = (r, \theta), \quad v = (u, \omega), \quad \tilde{F} = (F, T)$$

The generalized relative velocity is

$$\Delta v = (u - U^\infty, \omega - \Omega^\infty)$$

for both translational and angular motion. Then, the hydrodynamic interactions, i.e., forces and torques exerted on N bodies, can be expressed as

$$
\begin{bmatrix} \Delta v_j \\ -E_j^\infty \end{bmatrix} = - \begin{bmatrix} \mu_{vF}^\infty & \mu_{vS}^\infty \\ \mu_{EF}^\infty & \mu_{ES}^\infty \end{bmatrix} \begin{bmatrix} \tilde{F}_i^H \\ S_i^H \end{bmatrix}
\qquad (25)
$$

where S^H is the hydrodynamic stresslet. The matrix μ^∞ (multiplied to $\left[\tilde{F}_i^H, S_i^H\right]^{tr}$) is called far-field grand mobility matrix. An inverse relationship of Eq. (25) is

$$
\begin{bmatrix} \tilde{F}_i^H \\ S_i^H \end{bmatrix} = - \begin{bmatrix} R_{Fv}^\infty & R_{FE}^\infty \\ R_{Sv}^\infty & R_{SE}^\infty \end{bmatrix} \begin{bmatrix} \Delta v_j \\ -E_j^\infty \end{bmatrix}
\qquad (26)
$$

where the matrix R^∞ (multiplied to $\left[\Delta v_j, -E_i^\infty\right]^{tr}$) is the far-field grand resistance matrix as an inverse of μ^∞, having the mathematical identity as

$$
\begin{bmatrix} \mu_{vF}^\infty & \mu_{vS}^\infty \\ \mu_{EF}^\infty & \mu_{ES}^\infty \end{bmatrix} \cdot \begin{bmatrix} R_{Fv}^\infty & R_{FE}^\infty \\ R_{Sv}^\infty & R_{SE}^\infty \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}
\qquad (27)
$$

where I is the identity matrix. Note that $(\mu^\infty)^{-1} = R^\infty$ and $(R^\infty)^{-1} = \mu^\infty$ for grand matrices but $\left(R_{Fv}^\infty\right)^{-1} \neq \mu_{vF}^\infty$ for sub-matrices. The grand resistance matrix R in

Eq. (8) refers to R_{Fv}^{∞} of Eq. (26), and $R_{FE}^{\infty} \cdot E_j^{\infty}$ is an extra forcing term due to the rate of strain.

Note that at time t, the right-hand side of Eq. (25) is known, and one can calculate the generalized hydrodynamic force $\tilde{F}_H$ by using an appropriate solver in numerical linear algebra [25, 26]. As noted in Eq. (15), the required information to evolve the motion of particle j, located at r_j with linear and angular velocities of v_j and ω_i, respectively, is the ambient flow field consisting of $(U^{\infty}, ;\Omega^{\infty}, -E^{\infty})$ at the particle location. Unlike MD and BD, DHD explicitly includes the sizes of individual spherical particles. In this case, r_j indicates the position of the particle centroid or center of mass, and the fluid field is calculated at r_j without considering the presence of the particles. Therefore, the calculation of the ambient flow field is highly dependent on a mesh structure used for CFD simulations.

3. Computational fluid dynamics coupled with dissipative hydrodynamics

A general governing equation for fluid dynamics is Navier–Stokes equation, of which most general form for incompressible fluid is

$$\frac{\partial U}{\partial t} + U \cdot \nabla U = -\frac{1}{\rho}\nabla P + \frac{\eta}{\rho}\nabla^2 U \tag{28}$$

where U is the flow velocity, P is pressure, η and ρ are the viscosity and density of the fluid, respectively. In the adjacent space of particle j at r_j, U can be expanded as expressed in Eq. (15). Most fluid dynamic problems have at least three boundaries, which are the inlet, outlet, and side walls. For inlet and outlet surfaces, Neuman and Dirichlet or mixed boundary conditions are often used to set values or conditions of U and P. On the wall, zero velocity and zero-gradient pressure are usually assigned. The former condition assumes that there are strong adhesion forces between solvent molecules and wall surfaces. The solvent molecules are fixed on the wall. The velocities of the wall-adsorbed solvent molecules are equal to those of the solid walls, which is zero for non-moving walls. Values of (U, P) are calculated at grid points in internal spaces surrounded by the boundary surfaces. The simulation accuracy and numerical convergence highly depend on the mesh structures. When CFD is coupled with particle dynamics, which is in this study, DHD, there are additional requirements that should be satisfied to evolve the motion of multi-particles moving in a fluid flow:

1. To identify a cell of the constructed mesh grid, which contains the kth particle's position, r_k, for $k = 1 - N_p$

2. To calculate the distance between r_k and wall surfaces, if the particle is close to wall boundaries

3. To interpolate the flow field $(U^{\infty}, ;\Omega^{\infty}, -E^{\infty})$ at r_k within a cell that contains kth particle

Possible methods to satisfy the three requirements are dependent on available CFD solvers and flexibility of applying customized modifications. Here we suggest fundamental approaches to meet the requirement numerically.

3.1 Geometric calculations

3.1.1 How to determine a cell containing the centroid position of my particle

Computational grid cells have various structures such as hexahedron, wedge, prism, pyramid, tetrahedron, and tetrahedron wedge. Among them, hexahedron followed by tetrahedron structures is widely used to generate mesh structure of bulk (internal) spaces. Cubic and rectangular shapes are representative structures of hexahedrons, consisting of eight vertices (points) and six rectangular (or square) surfaces. On the other hand, a tetrahedron cell has only four vertices (as compared to eight in hexahedron) and three triangular surfaces. Each of these two cell structures has its advantages and disadvantages in CFD simulations. Formation of hexahedron meshes is straightforward, and numerical solutions are well converged to provide accurate results within a tolerable error, especially if edge lines are well aligned to the flow directions. However, if a computational domain includes complex and curved surface structures such as human faces or globes, the hexahedron meshes often provide unrealistic small exuberances instead of well-curved surfaces. As three triangular surfaces surround the tetrahedron volume, it can form well-fitted boundary layers of arbitrary shapes. As edges of tetrahedrons cannot be fully aligned on a straight line, the numerical convergence of tetrahedron meshes is often more sensitive to the fineness of generated mesh structures than that of hexahedron meshes. To overcome this limitation, one can make tetrahedron meshes often finer than that of hexahedron meshes to solve the same problem with similar accuracy. Nevertheless, there are unique mathematical advantages of using tetrahedron meshes for coupled simulations of CFD and particle hydrodynamics.

Location test using volume calculation. A tetrahedron have four points, p_i, located at position r_i for $i = 1 - 4$, where $r_i = (x_i, y_i, z_i)$. The volume of the tetrahedron can be calculated as

$$V = \frac{1}{6} \det(J) \tag{29}$$

where J is the Jacobian matrix given as

$$J(1, 2, 3, 4) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{bmatrix} \tag{30}$$

Using this mathematical relationship, one can easily test whether a particle position $r_p = (x_p, y_p, z_p)$ is inside or outside the tetrahedron. If r_p is inside of the tetrahedron, then the total volume of the tetrahedron can be divided by four small pieces and their sum is equal to V:

$$V = V_p \tag{31}$$

using

$$V_p = \frac{1}{6}[J(p, 2, 3, 4) + J(1, p, 3, 4) + J(1, 2, p, 4) + J(1, 2, 3, p)] \tag{32}$$

where, for example, $J(p, 2, 3, 4)$ means the Jacobian matrix of Eq. (30) with the first column $(1, x_1, y_1, z_1)^{tr}$ replaced by $(1, x_p, y_p, z_p)^{tr}$. Conversely, if r_p is outside of the tetrahedron, we have an inequality of

$$V < V_p \qquad (33)$$

If a particle is very close to one of the triangular surfaces of the tetrahedron, then the inequality check of Eq. (33) may not be done accurately. In this case, this location check can be extended to the nearest neighbor cells, especially one that shares the triangular surface that the particle is closely located.

For r_p within the tetrahedron, dividing both sides of Eq. (32) by V gives

$$1 = \xi_1 + \xi_2 + \xi_3 + \xi_4 \qquad (34)$$

where, for $i = 1 - 4$,

$$\xi_i = \frac{J(\text{without } i)}{6V} \qquad (35)$$

which is solely determined by the internal position r_p. A relationship between r_p and ξ is

$$
\begin{bmatrix} 1 \\ x_p \\ y_p \\ z_p \end{bmatrix} =
\begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{bmatrix}
\begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \end{bmatrix} \qquad (36)
$$

where the inverse of the 4×4 matrix can be analytically available. Eq. (36) assumes that r_p is located within the tetrahedron, but it can also use the particle location check as a better alternative to Eq. (33) because Eq. (36) requires one time solving of a 4×4 linear system, but Eq. (33) requires four times calculation of 4×4 Jacobian matrices.

3.1.2 How to calculate a distance between wall surfaces to the position of my particle

Using the four vertices of (r_1, r_2, r_3, r_4) of a tetrahedron cell, one can identify four triangular surfaces of $S_1(r_2, r_3, r_4)$, $S_2(r_1, r_3, r_4)$, $S_3(r_1, r_2, r_4)$, and $S_4(r_1, r_2, r_3)$. For example, let us consider S_4 having vertices of (r_1, r_2, r_3). If we calculate relative position of vertices 2 and 3 with respect to vertex 1, denoted as

$$r_{2/1} = r_2 - r_1 \qquad (37)$$
$$r_{3/1} = r_3 - r_1 \qquad (38)$$

their cross product allows us to calculate a unit vector normal to surface S_4:

$$n_4 = \frac{r_{2/1} \times r_{3/1}}{|r_{2/1} \times r_{3/1}|} \qquad (39)$$

The particle position relative to r_1 is $r_{p/1} = r_p - r_1$. Then, the distance between surface S_4 and position r_p is simply

$$d = |n_4 \cdot r_p| \qquad (40)$$

where the absolute value is necessary because n can direct inside or outside the tetrahedron volume, depending on choice of the reference position, r_1. This wall-particle distance calculation is necessary when the surface S_4 is known as a wall

boundary, so the distance d cannot be smaller than the particle radius a_p. After a single-step time evolution, if a particle is found overlapped with a wall surface, then collision rule will be applied to return the particle to the tetrahedron interior.

3.1.3 How to interpolate the flow field at the position of my particle

If a particle k is found inside a tetrahedron (of index l), the flow field consisting of $(\boldsymbol{U}^\infty, ; \boldsymbol{\Omega}^\infty, -\boldsymbol{E}^\infty)$ needs to be interpolated using values calculated adjacent locations. A proper choice of a set of these locations are the vertex points of the containing cell. Based on the definition of $\boldsymbol{\Omega}^\infty$ and $\boldsymbol{E}^\infty$, the basic quantities needed are U_i^∞ and $\partial_j U_i \equiv \partial U_i / \partial x_j$ for i and $j = 1 - 3$. (Note that pressure value is not required to calculate the generalized hydrodynamic forces $\tilde{\boldsymbol{F}}_H$.)

Four vertices of a cell can be represented as $\boldsymbol{r}_l = (x_l, y_l, z_l)$ for $l = 1 - 4$. A scalar quantity of interest S at each vertex point can be denoted as S_l. Then, the value of S at an arbitrary internal position $\boldsymbol{r}_p = (x_p, y_p, z_p)$ can be calculated as a linear superposition of S_l and ξ_l:

$$S_p = \boldsymbol{S} \cdot \boldsymbol{\xi} = \sum_{l=1}^{4} S_l \xi_l \tag{41}$$

For DHD simulations, S_p represents each element of the unidirectional velocity $\boldsymbol{U}^\infty$, vortex $\boldsymbol{\Omega}^\infty$, and the rate of strain $\boldsymbol{E}^\infty$, which are calculated using U_i^∞ or its gradient $\partial_j U_i$.

3.2 Collision rules

Suppose there are two particles colliding each other, which are particle i and j located at $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$, translating with $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ and rotating with $\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_j$, respectively. Relative position of particle i with respect to particle j is defined as $\boldsymbol{r}_{ij} = \boldsymbol{r}_i - \boldsymbol{r}_j$, and similarly the relative velocity of i to j is $\boldsymbol{v}_{ij} = \boldsymbol{v}_i - \boldsymbol{v}_j$. A normal vector from j to i is denoted as

$$\boldsymbol{n}_{ij} = \frac{\boldsymbol{r}_{ij}}{|\boldsymbol{r}_{ij}|} \tag{42}$$

After an instantaneous collision, the two particle have the following velocities [14]:

$$\boldsymbol{v}_i' = \boldsymbol{v}_i - \left(\frac{\mu_{ij}}{m_i}\right)\left[(1+\varepsilon^n)\boldsymbol{g}_{ij}^n + \frac{1-\varepsilon^t}{1+1/\tilde{J}\boldsymbol{g}_{ij}^t}\right] \tag{43}$$

$$\boldsymbol{v}_j' = \boldsymbol{v}_j + \left(\frac{\mu_{ij}}{m_j}\right)\left[(1+\varepsilon^n)\boldsymbol{g}_{ij}^n + \frac{1-\varepsilon^t}{1+1/\tilde{J}\boldsymbol{g}_{ij}^t}\right] \tag{44}$$

$$\boldsymbol{\omega}_i' = \boldsymbol{\omega}_i + \left(\frac{\mu_{ij}}{m_i a_i}\right)\left(\frac{1-\varepsilon^t}{1+\tilde{J}}\right)\boldsymbol{n}_{ij} \times \boldsymbol{g}_{ij}^t \tag{45}$$

$$\boldsymbol{\omega}_j' = \boldsymbol{\omega}_j + \left(\frac{\mu_{ij}}{m_j a_j}\right)\left(\frac{1-\varepsilon^t}{1+\tilde{J}}\right)\boldsymbol{n}_{ij} \times \boldsymbol{g}_{ij}^t \tag{46}$$

where $0 \leq \varepsilon^n \leq 1$ and $-1 \leq \varepsilon^t \leq 1$ are restitution coefficients in the normal and tangential directions, respectively, μ_{ij} is a reduced mass defined as

$$\mu_{ij} = \frac{m_i m_j}{m_i + m_j}, \tag{47}$$

and g_{ij} is the relative velocity at the point of contact defined as

$$g_{ij} = v_{ij} - \left(a_i \omega_i + a_j \omega_j\right) \times n_{ij} \tag{48}$$

whose normal and tangential components are

$$g_{ij}^n = \left(g_{ij} \cdot n_{ij}\right) n_{ij} \tag{49}$$

$$g_{ij}^t = -n_{ij} \times \left(n_{ij} \times g_{ij}\right) \tag{50}$$

If a collision between a wall surface and particle i occurs, then the wall can be represented as a stationary spherical particle j having infinite mass and radius, i.e., $m_j \to \infty$, $v_j \to 0$, and $a_j \to \infty$:

$$v_i' = v_i - \left[(1 + \varepsilon^n) g_{ij}^n + \left(\frac{1 - \varepsilon^t}{1 + \tilde{J}^{-1}}\right) g_{ij}^t\right] \tag{51}$$

$$\omega_i' = \omega_i + \left(\frac{1 - \varepsilon^t}{1 + \tilde{J}}\right) \frac{n_{ij} \times g_{ij}^t}{a_i} \tag{52}$$

$$v_j' = v_j = 0 \tag{53}$$

$$\omega_j' = \omega_j = 0 \tag{54}$$

where

$$g_{ij} \to v_i - a_i \left(\omega_i \times n_{ij}\right) \tag{55}$$

$$n_{ij} \to n \text{ to particle } i \tag{56}$$

and n is the normal vector of the colliding wall surface inward to the liquid volume. During the collision of particle i with the wall, the wall is not moving so that $\omega_j' = \omega_j = 0$ and $v_j' = v_j = 0$.

4. Concluding remarks

Each of computational fluid dynamics and particle hydrodynamics is a challenging research topic, as applied to real engineering problems. Movements of particles (viewed as small solid pieces) in a moving fluid require rigorous interfaces to couple the two strongly correlated dynamic events. When the ambient flow pushes suspended particles in a liquid, dynamic responses of particles to exerting fluid change the fluid motion at the next time step, which returns to the particles with modified magnitude and direction. This fluid-particle (or fluid–solid) interaction is under the regime of Newton's third law, i.e., action and reaction. Multi-body simulations including the fluid-particle interaction are generally a difficult task. In this chapter, we briefly reviewed the CFD and PHD literature and discussed a feasible

method to simulate the coupled fluid-particle motion within a reasonable time duration. A tetrahedron-based mesh is proposed to take the mathematical advantages of tetrahedron structure, consisting of four vertices and four triangular surfaces. Advantages of tetrahedron meshes include the following features: first, to test a location of particle within or outside a tetrahedron mesh-cell; second, to calculate a distance between a particle surface and a wall surface; third, to interpolate a fluid velocity and its gradient using those of values given at vertex locations; and finally to track each particle from one cell to the other by using a pre-built list of nearest neighbor cells. As DHD uses the SD algorithm for hydrodynamics of non-Brownian particles and Ito-Weiner process for random fluctuating forces, it can be used as a general particle hydrodynamic simulation method when it is coupled with CFD using specific mesh structures. Hexahedron-based meshes can be used for the same purpose with the intrinsic advantages of aligning grid edges to estimated streamline directions. When particles move in a channel of complex geometry, boundary surfaces can be better constructed using tetrahedron meshes. Open-sourced meshes include gmsh, tetgen, and netgen, which can import structure files, generate meshes, and export them to a CFD solver package. The current coupling algorithm of CFD and DHD is limited to cases that particle Reynolds number does not exceed 1.0, but this restriction can be avoided by considering dominant forces/torques exerted on particles and simulation time intervals as compared to the particle relaxation time. The new coupling method covered in this chapter may provide a new foundation in a coupled simulation of CFD and DHD including DEM.

Acknowledgements

A. Appendix

A.1 The Oseen tensor and Faxen law

The Navier–Stokes equation for a laminar flow is from Eq. (28):

$$-\nabla p + \eta \nabla^2 \boldsymbol{U} = -\boldsymbol{F}\delta(\boldsymbol{x}) \tag{57}$$

$$\nabla \cdot \boldsymbol{U} = 0 \tag{58}$$

where $\delta(\boldsymbol{r})$ is the Dirac-Delta function, which indicates

$$\delta(\boldsymbol{x} \neq 0) = 0 \tag{59}$$

$$\int_V -\boldsymbol{F}\delta(\boldsymbol{r})\mathrm{d}V' = -\boldsymbol{F} \tag{60}$$

where V' is a volume enclosing the origin $\boldsymbol{r} = 0$. The fundamental solution for $\boldsymbol{v}$ and p are

$$\boldsymbol{U}(\boldsymbol{x}) = \boldsymbol{F} \cdot \frac{\mathcal{G}(\boldsymbol{x})}{8\pi\eta} \tag{61}$$

$$p(\boldsymbol{x}) = \boldsymbol{F} \cdot \frac{\mathcal{P}(\boldsymbol{x})}{8\pi\eta} \tag{62}$$

The Oseen tensor $\mathcal{G}(\boldsymbol{x})$ for the fluid velocity is given by

$$\mathcal{G}_{ij} = \frac{1}{r}\delta_{ij} + \frac{1}{r^3}x_i x_j \tag{63}$$

which is independent of fluid properties. The Oseen tensor for the pressure p is

$$\mathcal{P}_j(\boldsymbol{x}) = 2\eta\frac{x_j}{r^3} + \mathcal{P}_j^\infty \tag{64}$$

where $\mathcal{P}_j^\infty$ is a constant at the ambient condition. The Faxen laws determine the hydrodynamic force and torque, especially, exerted on a sphere of radius a, moving with the linear and angular velocities of $\boldsymbol{u}$ and $\boldsymbol{\omega}$:

$$\boldsymbol{F} = 6\pi\eta a\left[\left(1 + \frac{a^2}{6}\nabla^2\right)\boldsymbol{U}(\boldsymbol{x})\right]_{\boldsymbol{x}=0} - 6\pi\eta a\boldsymbol{u} \tag{65}$$

$$\boldsymbol{T} = 8\pi\eta a^3[\Omega(\boldsymbol{x}) - \boldsymbol{\omega}]_{\boldsymbol{x}=0} \tag{66}$$

which indicates that the Stokes flow requires a quadrupole $a^2\boldsymbol{F}\nabla^2\delta(\boldsymbol{x})$ in addition to a monopole of $-6\pi\eta a\boldsymbol{u}$, which is a drag on a sphere undergoing steady translation.

Author details

Albert S. Kim[1*] and Hyeon-Ju Kim[2]

1 Civil and Environmental Engineering, University of Hawai'i at Manoa, USA

2 Offshore Plant and Marine Energy Research Division, Korea Research Institute of Ships and Ocean Engineering, Korea

*Address all correspondence to: albertsk@hawaii.edu

IntechOpen

References

[1] Metropolis N, Ulam S. The Monte Carlo method. Journal of the American Statistical Association. 1949;**44**(247): 335-341

[2] Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. Equation of state calculations by fast computing machines. The Journal of Chemical Physics. 1953;**21**(6):1087-1092

[3] Wood WW, Parker FR. Monte carlo equation of state of molecules interacting with the lennard-jones potential. i. a supercritical isotherm at about twice the critical temperature. The Journal of Chemical Physics. 1957; **27**(3):720-733

[4] Hess J, Smith A. Calculation of potential flow about arbitrary bodies. Progress in Aerospace Sciences. 1967;**8**: 1-138

[5] Rubbert P, Saaris G. Review and evaluation of a three-dimensional lifting potential flow computational method for arbitrary configurations. In: 10th Aerospace Sciences Meeting. Reston, VA: American Institute of Aeronautics and Astronautics; 1972

[6] Maskew B. Prediction of subsonic aerodynamic characteristics: A case for low-order panel methods. Journal of Aircraft. 1982;**19**(2):157-163

[7] Hyman JM, Knapp RJ, Scovel JC. High order finite volume approximations of differential operators on nonuniform grids. Physica D: Nonlinear Phenomena. 1992;**60**(1–4):112-138

[8] Kubo R. The fluctuation-dissipation theorem. Reports on Progress in Physics. 1966;**29**:255-284

[9] Langevin P. Sur la the'orie du mouvement brownien. Comptes Rendus de l'Académie des Sciences (Paris). 1908;**146**:530-533

[10] Ermak DL, McCammon JA. Brownian dynamics with hydrodynamic interactions. The Journal of Chemical Physics. 1978;**69**(4):1352-1360

[11] Hoogerbrugge P, Koelman J. Simulating microscopic hydrodynamic phenomena with dissipative particle dynamics. Europhysics Letters. 1992;**19**: 155

[12] Koelman J, Hoogerbrugge P. Dynamic simulations of hard-sphere suspensions under steady shear. Europhysics Letters. 1993;**21**:363

[13] Brady JF, Bossis G. Stokesian dynamics. Annual Review of Fluid Mechanics. 1988;**20**:111-157

[14] Kim AS, Kim H-J. dissipative dynamics of granular materials. In: Granular Materials. Rijeka: InTechOpen; 2017. pp. 9-42

[15] Kim AS. Constraint dissipative hydrodynamics (HydroRattle) algorithm for aggregate dynamics. Chemistry Letters. 2012;**41**(10): 1285-1287

[16] Kim AS. Dissipative hydrodynamics of rigid spherical particles. Chemistry Letters. 2012;**41**(10):1128-1130

[17] Kim AS, Kang S-T. Microhydrodynamics simulation of single-collector granular filtration. Chemistry Letters. 2012;**41**(10): 1288-1290

[18] Verlet L. Computer 'experiments' on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. Physics Review. 1967;**159**(1):98-103

[19] Martys NS, Mountain RD. Velocity Verlet algorithm for dissipative-particle-dynamics-based models of suspensions. Physical Review E. 1999; **59**(3):3733 LP-3733736

[20] Hockney RW, Eastwood JW. Computer Simulation Using Particles. New York: Adam Hilger; 1988

[21] Hockney RW. The potential calculation and some applications. In: Methods in Computational Physics. Vol. 9. New York: Academic Press; 1970

[22] Stokes GG. On the effect of internal friction of fluids on the motion of pendulums. Transactions of the Cambridge Philosophical Society. 1851;**9**:1-106

[23] Wiener N. Differential space. Journal of Mathematical Physics. 1923;**58**:31-174

[24] Ito M. An extension of nonlinear evolution equations of the K-dV (mK-dV) type to higher orders. Journal of the Physical Society of Japan. 1980;**49**(2): 771-778

[25] Anderson E. LAPACK Users' Guide. Philadelphia, PA: Society for Industrial and Applied Mathematics; 1987

[26] Blackford LS, Choi J, Cleary A, D'Azevedo E, Demmel J, Chillon I, et al. ScaLAPACK User's Guide. Philadelphia, PA: Society for Industrial and Applied Mathematics; 1997