International Journal of Multiphase Flow 34 (2008) 1023–1030

Contents lists available at ScienceDirect

International Journal of Multiphase Flow

journal homepage: www.elsevier.com/locate/ijmulflow

A new point-locating algorithm under three-dimensional hybrid meshes

S.B. Kuang [a], A.B. Yu [a,*], Z.S. Zou [b]

[a] Centre for Simulation and Modelling of Particulate Systems, School of Materials Science and Engineering, The University of New South Wales, Sydney, NSW 2052, Australia
[b] School of Materials and Metallurgy, Northeastern University, Shenyang, Liaoning 11004, PR China

ARTICLE INFO

Article history:
Received 11 December 2007
Received in revised form 6 May 2008
Accepted 27 June 2008
Available online 13 July 2008

Keywords:
Point-locating algorithm
Hybrid meshes
Non-planarity
Discrete particle simulation

ABSTRACT

When particles are traced individually in a numerical simulation of particle-fluid flow, a point-locating scheme is necessary to relate the coordinates of a given point to the grid cell containing it. A new point-locating scheme combining the advantages of the existing ones is presented in this paper. The new method is first explained in detail. Then its performance under three-dimensional hybrid meshes is compared with the existing approaches in the literature. The results show that the proposed method is efficient, multifunctional and easy to implement in a structured/unstructured fluid flow solver.

1. Introduction

When particles in a particle-fluid flow are individually traced while fluid flow is solved over grid cells, the coordinates of a particle are required to relate to the cell containing it by a point-locating scheme at each time step to calculate particle trajectory, its interaction with fluid, and other properties of interest. In the case of uniform Cartesian grid, point location is simple and straightforward. But this dose not hold irregular grids (Sadarjoen et al., 1997), which are widely used for complex geometry.

Generally speaking, there are four approaches proposed to locate particles in irregular grids, i.e. Cartesian background grid method (Seldner and Westermann, 1988; Muradoglu and Kayaalp, 2006), mapping technique (Oliveira et al., 1997; Patankar and Karki, 1999; Schafer and Breuer, 2002; Haegland et al., 2007), tetrahedral walk method (Garrity, 1990; Kenwright and Lane, 1995; Lohner, 1995; Deguzman et al., 1995; Sadarjoen et al., 1998), and face-by-face search method (Frank and Schulze, 1994; Chen, 1997; Chen and Pereira, 1999; Zhou and Leschziner, 1999; Subramaniam and Haworth, 2000; Li and Modest, 2001; Chordá et al., 2002; Apte et al., 2003; Vaidya et al., 2006; Haselbacher et al., 2007). By locating particles in a regular grid instead of the original irregular (physical) grid, the first two schemes can be simplified. However, there are various deficiencies associated with them. For example, inaccuracy and inefficiency may rise in the calculation of particle trajectory on the basis of the Cartesian background grid method due to large variations in grid size, and the mapping method is limited to structured grids (Lohner and Ambrosiano, 1990). It has been recognized that such deficiencies do not exist in the last two methods which perform point location directly in a physical grid. In practice, some cell faces of physical grids are inevitably slightly non-planar due to the deformation of three-dimensional (3D) grids in order to account for complex geometries. The tetrahedral walk method, which locates a particle based on a tetrahedron enclosed by triangular plane surfaces, needs not to consider the non-planarity of grid. But the method involves a sophisticated tetrahedral decomposition in structured curvilinear grids containing hexahedral cells or unstructured hybrid grids containing mixed element types (Albertelli and Crawfis, 1997; Sadarjoen et al., 1998; Dompierre et al., 1999). On the contrary, the face-by-face search method approximately treats non-planar faces as plane surfaces. With such approximation, the point-locating scheme is simple and easy to implement, but it may "force" particles to be trapped near non-planar faces.

In order to obtain a good CPU-time performance with enough robustness, as summarized by Lohner (1995) and Vaidya et al. (2006), various approaches have been proposed to optimize the search path of point location. Among the most prominent methods is the one proposed by Chen and Pereira (1999). The method, also referred as to directed search, only examines the cells passed through by a considered particle, and is thus very beneficial to the determination of some crucial information for modeling multiphase flow, such as particle–wall interaction and particle residence time (Chen, 1997; Chen and Pereira, 1999; Vaidya et al., 2006). However, it suffers from some important disadvantages, such as large CPU times in a general particle-locating context, and not-too-clear extension to 3D grids due to the non-planar faces, as pointed out by Chordá et al. (2002).

* Corresponding author. Tel: +61 2 93854429; fax: +61 2 93855856.
E-mail address: a.yu@unsw.edu.au (A.B. Yu).

0301-9322/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijmultiphaseflow.2008.06.007

In this paper, the above-mentioned non-planar problem is first addressed. An improved directed point-locating scheme applicable to 3D structured/unstructured grids is then proposed in the framework of the face-by-face search method. Finally, the performance of the proposed method is analysed with reference to various applications in particle-fluid flow.

2. Non-planar problem

In the face-by-face search method, a particle is determined to be inside a grid cell when the particle is located inside a set of the surfaces of the cell. In 3D grids, it is mathematically expressed as (Chordá et al., 2002):

$$l_j = \mathbf{n}_j \cdot \mathbf{r}_p \geqslant 0 \quad \text{for all} \quad j, 1 \leqslant j \leqslant N_c \tag{1}$$

where $\mathbf{n}_j = (\mathbf{p}_i - \mathbf{p}_{i+1}) \times (\mathbf{p}_{i-1} - \mathbf{p}_i)$, $\mathbf{r}_p = \mathbf{p}_i - \mathbf{p}_{new}$, $\mathbf{p}_i$ represents the vertex i anticlockwisely ordered on cell faces (see Fig. 1), $\mathbf{p}_{new}$ represents a considered particle, and N_c is the number of the faces of a cell. It is found that Eq. (1) sometimes fails to detect particles which reside around non-planar faces. To highlight this problem, a cell with one non-planar face (see Fig. 1(a)) is first considered. Fig. 1 shows that the application of Eq. (1) to the cell results in a small part of the cell being missed in the point location. Note that the missed volume may vary with the choice of vertices in Eq. (1). If there is only one non-planar face at the same space of grid, the missed volume can be recounted in the neighbours of the cell, thus no trouble happens to point location. However, a high-density distribution of non-planar faces may lead to a virtual gap in grid. Once a particle enters such a virtual gap, it "disappears" in the domain of interest.

How a virtual gap is formed can be schematically illustrated in Fig. 2. It is assumed that the four hexahedra of the grid in the figure each have a non-planar face and these non-planar faces share a common line. The application of Eq. (1) to these cells (referring to Fig. 1) gives four "new hexahedra" and one "pyramid". This pyramid is the so-called virtual gap and not considered in locating particles. Depending on the degree of planarity, the number of non-planar faces, and the grid vertices involved, the shape of a virtual gap varies. However, it is inevitable for most actual applications involving complex geometries as their grids usually have non-planar faces to some extent.

As an example to support the above arguments, the motion of a particle in a pneumatic pipeline is studied. The computational domain, consisting of horizontal and vertical straight sections and a bend (see Fig. 3(a)), is meshed with hexahedral cells. The particle trajectory shown in Fig. 3(b) is calculated by a commonly used discrete element method (DEM) code (Zhu et al., 2007). At each simulation time step, Eq. (1) is applied to each cell to detect the particle. Note that in this case the point locations do not affect the DEM calculation because fluid flow is not considered. The results with different time steps and initial particle positions show that the point location sometimes fails near some cell face in the bend. Fig. 3(c) is a typical result of such a case. Note that the figure

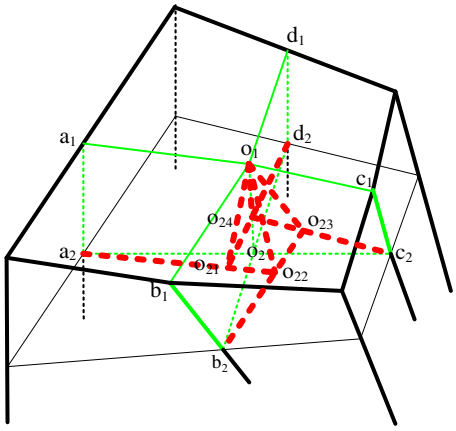

Fig. 2. A virtual gap (pyramid $o_1o_{21}o_{22}o_{23}o_{24}$) in mesh due to the approximate treatment of non-planar faces ($o_1o_2a_1a_2$, $o_1o_2b_1b_2$, $o_1o_2c_1c_2$, and $o_1o_2d_1d_2$ are non-planar faces, $o_1o_{22}a_1a_2$, $o_1o_{23}b_1b_2$, $o_1o_{24}c_1c_2$, and $o_1o_{21}d_1d_2$ are plane faces).

only shows the particle trajectory before an unsuccessful point location happens. Moreover, it can be seen from Fig. 4 that the non-planarity of cell faces in the vertical and horizontal sections is quite small and negligible, whereas a relatively large non-planarity for most cell faces in the bend is found. The non-planarity can lead to virtual gaps accounting for the failure of the point location in the bend. In Fig. 4 and other results in this work, the degree of non-planarity θ is introduced and defined as the angle between two triangles of a quadrilateral, as shown in Fig. 1(a). According to this definition, θ is equal to zero for triangular faces and is greater than zero for non-planar faces.

In light of the consideration above, Eq. (1) is modified:

$$l_j = \mathbf{n}_j \cdot \mathbf{r}_p \geqslant \delta_j \quad \text{for all} \quad j, 1 \leqslant j \leqslant N_c \tag{2}$$

where δ_j is a constant depending on the degree of non-planarity of a face and may be different for the same face owned by various cells. In the same case above, successful point locations can always be obtained according to Eq. (2) instead of Eq. (1). Here, the value of δ for all faces has been uniformly set to -10^{-6} after several trials. This constant largely corresponds to the biggest non-planarity in the bend in this work, and could be very different for different applications.

It is of interest that Haselbacher et al. (2007) have also proposed Eq. (2) to overcome the failure of Eq. (1), and attributed this problem to the round-off errors related to infinite-precision computation. If this consideration is correct, a particle may "disappear" everywhere in a considered domain where Eq. (1) applies to all cells having planar or non-planar faces. However, our numerical experiments show that a particle gets trapped only in the bend with an unignorable non-planarity (see Figs. 3(b) and 4), and this trapping does not occur to the vertical and horizontal sections. In fact, Eq. (1) only involves the calculation of the dot and cross products of vectors, the resultant round-off errors are trivial for the case

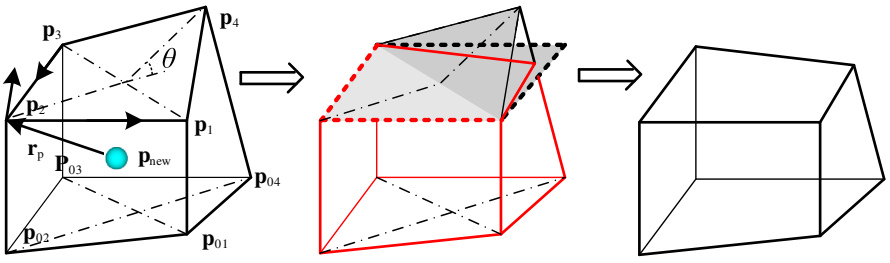

Fig. 1. Approximate treatment of a non-planar face.

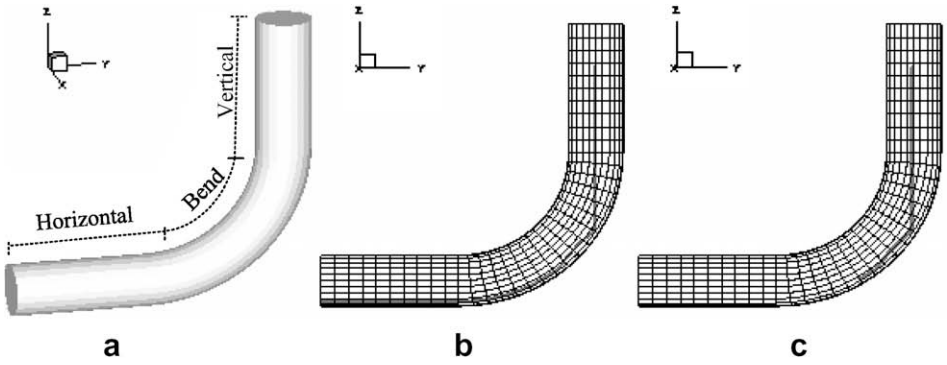

Fig. 3. Particle motion in a pneumatic conveying pipeline: (a) geometries; (b) particle trajectory; and (c) unsuccessful point location according to Eq. (1).

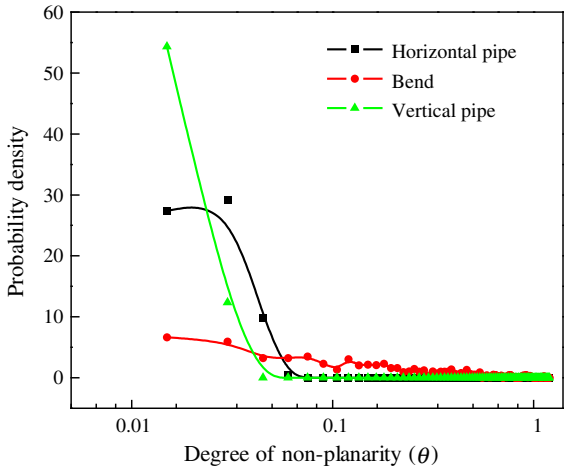

Fig. 4. Probability density distribution of the degrees of non-planarity in the grid shown in Fig. 3.

shown in Fig. 3. So the failure of Eq. (1) can stem from the virtual gap associated with the non-planarity, but not the numerical round-off errors.

Our arguments can be further supported by the underlying physics of δ. For clarity, Eq. (2) applies to the cell with one non-planar face shown in Fig. 1, and the result is plotted in Fig. 5. Here, we assume that the value of δ for a plane face is equal to zero. It can be seen that the introduction of δ for the non-planar face actually form a new conical face, which makes up a new cell together with other plane faces during point location. Here, the angle between

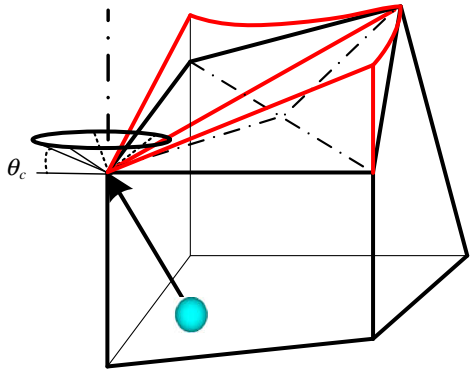

Fig. 5. Schematic illustration of the considered volume of a cell in point location based on Eq. (2), corresponding to Fig. 1(a) (the original cell is enclosed by black lines).

the conical face and the face $\mathbf{n}_j$ is θ_c (=arccos($\delta_j/|\mathbf{n}_j||\mathbf{r}_p|$)). When the magnitude of δ is big enough to make the new cell cover the whole space occupied by the original one (this is possible for Eq. (1)), the missed volumes and resultant virtual gaps according to Eq. (1) can be avoided and successful point locations (based on Eq. (2)) are hence obtained.

Eq. (2) can therefore be used to overcome the non-planar problem. In fact, it has been used in many previous studies although it is not explicitly mentioned and physical meaning is not understood properly. Its use is simply attributed to round-off errors or the so-called "fuzziness" in the topological position of a particle (Haselbacher et al., 2007). In theory, Eq. (2) can be used as a general method to solve the non-planar problem. However, as manifested by Fig. 5, the determination of δ according to non-planarity is practically difficult. The value of δ is not universal, and has to be estimated case by case. It also results in overlapping regions both around planar or non-planar faces. In these overlapping regions, the result of point location depends on search direction. More importantly, non-planar faces bring troubles not only to the location of particle (based on Eq. (1)), but also to the determination of search path in point location (Garrity, 1990; Chordá et al., 2002). Therefore, a more generalized and straightforward solution to the non-planar problem should be developed. This is the major contribution of this work, as explained below.

3. A new point locating scheme

A point-locating scheme is typically composed of two key procedures, i.e. identifying host cells which contain considered particles, and determining search paths. The section is hence organized according to the two components to explain the present point-locating scheme.

3.1. Identification of host cell

In order to eliminate non-planar faces, a 3D non-tetrahedron cell in the tetrahedral work method (Sadarjoen et al., 1998) needs to be decomposed into several tetrahedra in a sophisticated way. For simplicity, this work decomposes the non-triangle surfaces of a 3D cell into non-overlapping triangles to overcome the non-planar problem, as commonly used in the calculation of the volume of a deformed cell in computational fluid dynamics (Ferziger and Peric, 2002). The method, termed as triangular decomposition in this work, has no limit to cell type and is much easier to implement compared to the tetrahedral decomposition. But at the same time it yields concave polyhedra due to non-planar faces. So far there is no generalized point-locating scheme available to a concave grid. However, bearing in mind that fluid grids are necessarily smooth so as to acquire correct flow fields for most existing CFD codes

(Liu et al., 2006), a cell face could only be limitedly curved to a non-planar face. Below we propose a host-cell-identification method applicable to triangulated grids where some cells may be concave to some degree.

For convenience, the concept of positive face is first introduced. A face is supposed to be positive when a particle locates inside one of its triangles, and a host cell should be the one only having positive faces. This gives the following expression for a host cell:

$$l_{j,k} = \mathbf{n}_{j,k} \cdot \mathbf{r}_p \geqslant 0 \quad \text{for all} \quad j \quad \text{and for one of} \quad k, 1 \leqslant j \leqslant N_c, \quad 1 \leqslant k \leqslant N_t \tag{3}$$

where $\mathbf{n}_{j,k}$ ($=\mathbf{a}_1 \times \mathbf{a}_2$) is the area vector of a triangle and points outside the considered cell, as shown in Fig. 6, $\mathbf{p}_{ref}$ is a reference point randomly chosen from the vertices on the triangle of interest, $\mathbf{r}_p$ ($=\mathbf{p}_{ref} - \mathbf{p}_{new}$) directs from the current position of a particle to the reference point, $\mathbf{a}_1$ and $\mathbf{a}_2$ are two edges of the considered triangle, and N_t is the number of triangles decomposed from a quadrilateral face.

In order to demonstrate how Eq. (3) works in point locations and compare it with Eq. (2), we apply the two equations to the cell shown in Fig. 1(a). The cell volumes considered by Eqs. (2) and (3) are shown in Fig. 7. Here we have considered all the triangular decompositions in Eq. (3) and the methods for the calculation of $\mathbf{n}_j$ in Eq. (2) for the same non-planar face, and assumed that the δ_j of each face in Eq. (2) can be accurately calculated according to the non-planarity of face. It can be seen that both equations may lead to overlapping regions around non-planar faces. However, the overlapping regions do not get particle to be trapped. Fig. 7 also indicates that for the same non-planar face, the corresponding δ in Eq. (2) could be different with the choice of vertices for the calculations of the vectors $\mathbf{n}$ and $\mathbf{r}_p$ in the equation (see Fig. 7(a) and (b)). The situation may complicate the implementation of Eq. (2). However, Eq. (3) does not have this trouble, and its implementation does not vary with different triangular decompositions and is as easy as that of Eq. (1) but with much better robustness.

3.2. Determination of search path

The simple search path used is the brute-force algorithm which examines all grid cells to detect host cell. The method has already been used in Section 2. But it is extremely slow for a large number of particles. In the past, different search paths have been proposed in the literature (Lohner, 1995; Vaidya et al., 2006). In this work, we extend the potential directed search method of Chen and Pereira (1999) from 2D to 3D with the help of triangular decomposition, and some new techniques are also used to improve its efficiency and robustness.

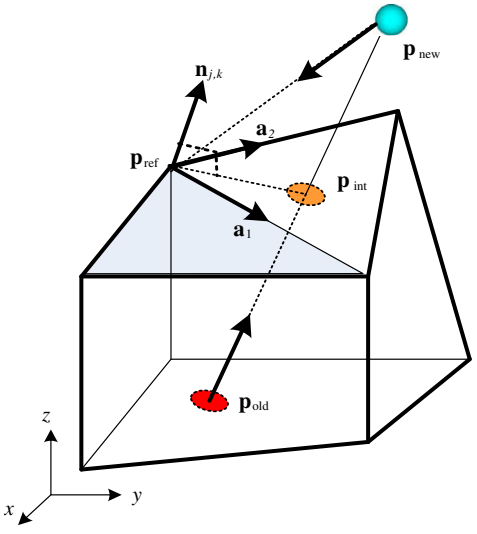

Fig. 6. Intersection point between a particle trajectory and a non-planar face.

The directed search path is developed to coincide with the particle trajectory in order to avoid examining redundant cells where a particle never resides at a current time step, as shown in Fig. 8. This is done as follows. Suppose that a considered particle has moved from one location to another. The search starts from the particle position at the last time step. The previous host cell is first examined. Then the search proceeds to an adjacent cell from an *exit face*, this repeats until the particle is detected. Evidently, the key operation in the procedure is to determine *exit faces*. Here, an *exit face* is defined as the cell surface which meets the following three conditions: (1) it is negative at the current time step; (2) it is not the *exit face* at the last time step; and (3) it has an internal intersection point with the particle trajectory. When a face meets the first two conditions, it is termed as a potential *exit face*, as illustrated in Fig. 9. Chen and Pereira (1999) have proposed to use intersection point to find an *exit face* in 2D grid. However, the method becomes uncertain in implementation due to non-planar faces in 3D grid. Actually, in order to avoid calculating intersection point, other methods have also been proposed to acquire an *exit face* (Garrity, 1990; Chordá et al., 2002; Haselbacher et al., 2007). The method of Milgram (1989) can also be a choice for such purpose although it originally aimed at the point location in a polygon. In this work, an approach similar to Chen and Pereira's method is proposed to determine an *exit face*, as explained in the following.

In a 3D triangulated grid, the intersecting point pint between a face and a particle trajectory should be considered between one of triangles decomposed from the face and the particle path no matter whether the face is non-planar or not, as shown in Fig. 6. Since $\mathbf{p}_{int}$ lies in the particle trajectory, it is subject to the equation:

$$\mathbf{p}_{int} = \mathbf{p}_{new} - \mathbf{v}\Delta t_s \tag{4}$$

where $\mathbf{v}$ is particle velocity vector, Δt_s is the time interval in which the considered particle moves from the intersection point to the new position.

At the same time, $\mathbf{p}_{int}$ belongs to the triangular face $\mathbf{n}_{j,k}$, so it can be expressed as

$$\mathbf{n}_{j,k} \cdot (\mathbf{p}_{int} - \mathbf{p}_{ref}) = 0 \tag{5}$$

Substituting Eq. (4) into Eq. (5) and combining the resultant equation with Eq. (3) give

$$\Delta t_s = -l_j / \mathbf{n}_{j,k} \cdot \mathbf{v} \tag{6}$$

For computational efficiency, the intersection point $\mathbf{p}_{int}$ is calculated according to Eqs. (4) and (6) only when a face is a potential *exit face*, and the following method is proposed to determine whether $\mathbf{p}_{int}$ is an internal intersection point.

Enlightened by the design principle of the tetrahedral walk method in determining a host cell (Lohner, 1995), we introduce a local coordinate system (η_1, η_2) in addition to one node of a triangle to describe the triangular surface owing $\mathbf{p}_{int}$, as shown in Fig. 6. The intersection point can hence be written as

$$\mathbf{p}_{int} = \mathbf{p}_{ref} + \eta_1 \mathbf{a}_1 + \eta_2 \mathbf{a}_2 \tag{7}$$

If $\eta_1 \geqslant 0$, $\eta_2 \geqslant 0$ and $\eta_1 + \eta_2 \leqslant 1$, $\mathbf{p}_{int}$ is located inside the triangle enclosed by lines $\eta_1 = 0$, $\eta_2 = 0$ and $\eta_1 + \eta_2 = 1$, thus the face owning the triangle is an *exit face*. Note that there are only two unknown variables (η_1, η_2) but three equations in Eq. (7) because the triangular face of interest is in a 3D space and $\mathbf{p}_{int}$ is located on the plane. Consequently, two equations are chosen to calculate the unknown variables η_1 and η_2 according to Cramer's rule. Once the *exit face* is determined, the particle residence time Δt in the cell can be readily calculated by

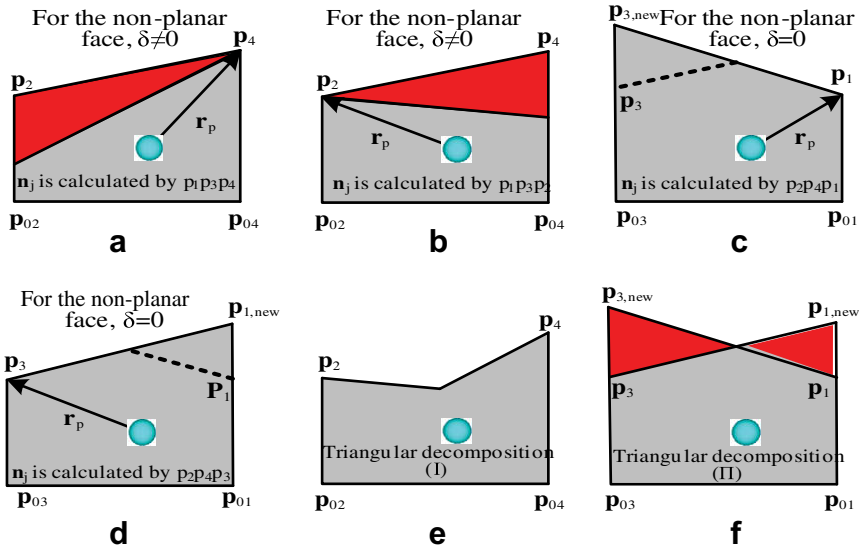

Fig. 7. Schematic illustration of the considered volume of a cell (on central cross-section) in point location based on Eqs. (2) and (3), corresponding to Fig. 1(a): (a)–(d) are obtained according to Eq. (2), and (e) and (f) are according to Eq. (3) (grey: original volume; red: enlarged volume). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

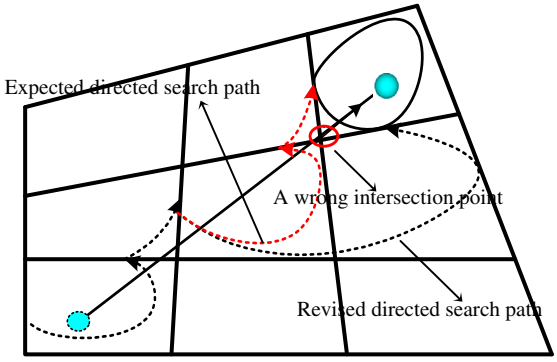

Fig. 8. Directed point-locating method.

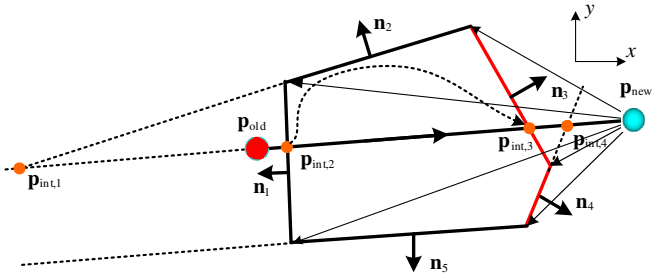

Fig. 9. Determination of *exit face* in the directed point-locating scheme (red line: potential *exit face*). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

$$\Delta t = \Delta t_{s-1} - \Delta t_s \tag{8}$$

Compared to other directed search methods, the new approach for determining search path has the following advantages: (i) the treatment of non-planar faces by the triangular decomposition is more straightforward and simpler than by the tetrahedral decomposition (Garrity, 1990) or the method based on a bilinear patch (Haselbacher et al., 2007), and applicable to a polygon; (ii) it can determine an *exit face* only by examining a small part of the triangles of a considered cell, whereas the methods (Garrity, 1990; Haselbacher et al., 2007), which determine the *exit face* according to the minimum intersection distance, have to examine all faces of the cell and further handle some degenerate cases, e.g. a cell has more than one face located in the same plane; (iii) it can share the information which has been calculated in the host-cell findings and the major calculations are based on two simple linear equations, which cannot be achieved in the methods proposed by Chordá et al. (2002) and Haselbacher et al. (2007); and (iv) it dose not require the ordered arrangement of vertices in a considered face in the method proposed by Chordá et al. (2002), and thus easier for implementation.

It is noted that the accurate determination of *exit face* is of great importance to all direct search methods no matter which methods have been used to obtain *exit face*. However, a wrong *exit face* may be obtained when a particle is residing around the corners of cells or the like due to computer truncation error. In such a case a particle could be trapped in a non-host cell without *exit face*. To overcome this problem, the method of Zhou and Leschziner (1999) is introduced to revise the search path, thus when a particle cannot find its *exit face*, the particle proceeds to an adjacent cell from the last negative of the considered cell and this repeats until the host cell is detected, as shown in Fig. 9.

3.3. Summary of the present scheme

For convenience, the outline of the proposed scheme is given below:

Compute and store vectors $\mathbf{n}_{j,k}$, $\mathbf{a}_1$ and $\mathbf{a}_2$ of each triangular face, and determine the initial host cell of each particle by using the brute-force method. These are all done only once. The search of a particle starts from its previous host cell and is carried out according to the following steps:

1. Determine whether a cell face is negative according to Eq. (3). If yes, go to step 2. When all faces of a cell are found to be positive, the new host cell is found and the search stops.
2. Examine whether a negative face is an *exit face* at the last time step according to the recorded information. If yes, go to step 1 to examine next face. If not, calculate the intersection point $\mathbf{p}_{\text{int}}$ according to Eq. (4) and go to step 3.

3. Determine whether $\mathbf{p}_{int}$ is an internal intersection point according to Eq. (7). If yes, the face is recorded and the search proceeds to the adjacent cell from the considered face and go to step 2. If not, go to step 2 to examine the next face.
4. Negative faces have been found, but there is no *exit face*. The search proceeds to the adjacent cell from the last negative face, and go to step 2.

4. Application examples

The applicability of the proposed method is examined with reference to a few typical particle-fluid flow cases. A dilute-phase gas–solid flow is first considered to evaluate the performance of the present point-locating scheme (KYZ). The comparison of the KYZ method with the methods presented by Zhou and Leschziner (1999) (ZL) and Chordá et al. (2002) (CBF) is carried out. Note that the ZL and CBF methods are recognized as the fastest point-locating schemes at this stage of development.

A 3D unstructured mesh is set up for a domain with a stagnant flow. The grid is shown in Fig. 10 and consists of 3864 hexahedra, 440 pyramids and 28,869 tetrahedra. The fluid flow is solved by an in-house unstructured solver. Particles are fed into the system from the inlet and traced by using a traditional Lagrangian method (Crowe et al., 1977) after the field of velocity vectors is available. Particle–particle interaction is neglected and one-way coupling, in which the influence of solid particles on fluid phase is neglected, is adopted. The calculation stops when all particles leave the domain of interest from the outlet. In the numerical experiments, three Lagrangian time steps 0.2, 0.002 and 0.00002 s are chosen to consider various situations in the field of multiphase flow. Some typical results of particle trajectories are also presented in Fig. 10.

In order to investigate the number of crossed cells which is inversely proportional to the efficiency of a point-locating scheme, one particle moving from the inlet to the outlet is first studied. The average number of crossed cells in one time step is given in Table 1. It can be seen that the KYZ and CBF methods traverse exactly the same cells. This fact is expected since both methods employ the directed search. On the other hand, the number of crossed cells with the ZL method is slightly higher (around 1%) when the Lagragian time step is large. But this difference disappears when the time step is small. It is because that the particle displacement is very small compared to grid cell in a small time step and point locations need not frequently proceed to adjacent cells as particles may stay in the same cell in a few time steps.

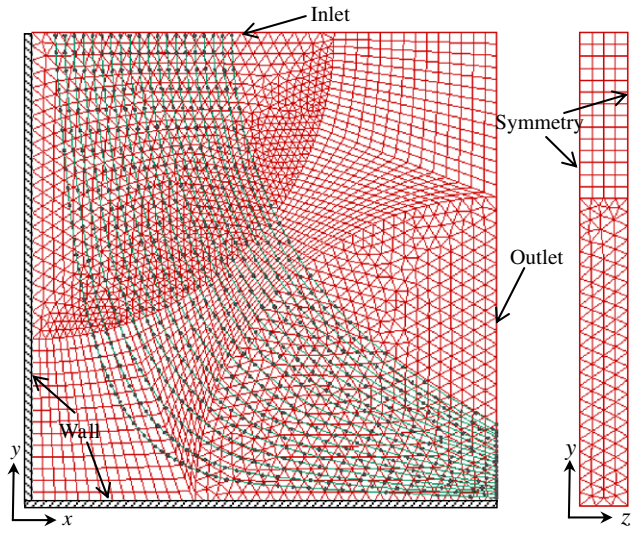

Fig. 10. Lagrangian particle tracing in 3D hybrid mesh (red line: fluid grid; green line: particle trajectory; black circle: particle). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

Table 1
Average crossed cell number of various point-locating schemes

Time step (s)	Methods		
	ZL	CBF	KYZ
	Crossed cell number		
0.2	14.85714	12.78571	12.78571
0.002	1.193237	1.191304	1.191304
0.00002	1.001909	1.001909	1.001909

The point-locating method of Chen and Pereira (1999) (CP) is the most favorite point-locating scheme in multiphase models because it can provide diverse additional information in addition to point location. But it could be 3 or 4 times slower than the ZL and CBF methods (Chordá et al., 2002). The KYZ method is actually a 3D version of CP method with a new approach in finding an *exit face*, which considerably improves the computational efficiency. To prove this point, the same setup for the simulation of fluid flow and particle motion are preserved except that 20 particles are repeatedly traced for 100 times to ensure properly counting CPU time of point location. The results in Table 2 indicate that the ZL, CBF and KYZ methods almost have the same computational efficiency. The KYZ method is faster than the other two methods, but the improvement is small, below 1%. Note here Eq. (3) are used to overcome the non-planar problem in the three methods.

The KYZ method has achieved an evident improvement in computer efficiency compared to the CP method, and is as efficient as the fastest ZL and CBF methods. More importantly, it has obtained great robustness by considering non-planar faces of grid and combining itself with the ZL method. At the same time, the new method provides additional information which is not available in the ZL and CBF methods, i.e. particle residence time and intersection point. The information is very useful in modeling multiphase flow, e.g. residence time can be used to improve the accuracy and efficiency of multiphase modes (Chen and Pereira, 1999; Vaidya et al., 2006), and intersection point can facilitate handling particle–wall interaction (Chen, 1997) and representing the position of a particle when it leaves the computational domain as illustrated in Fig. 10. In addition, in the KYZ method, the triangulation for handling non-planar faces does not introduce new elements and change the existing data structure of fluid solver, and the determination of *exit face* does not require the ordered arrangement of vertices on cell faces in the CBF method and applicable to a polygonal surface. The features make the KYZ method easier in implementation compared to other methods such as the tetrahedral walk method and the CBF method.

Finally, the KYZ method is applied to one of our coupled CFD–DEM (computational fluid dynamics and discrete element model) codes to further demonstrate its applicability to complex geometries. The related modeling techniques have been explained elsewhere (Chu and Yu, 2008; Kuang et al., 2008; Feng et al., 2004;

Table 2
CUP time of various point-locating schemes

Time step (s)	Methods		
	ZL	CBF	KYZ
	CUP time (s)		
0.2	65.55426	65.25383	64.85326
0.002	91.23047	91.59171	91.21123
0.00002	552.9150	552.0839	550.5917

Xu and Yu, 1997), thus not presented in this paper for brevity. Two interesting dense-phase multiphase systems, i.e. fluidization and pneumatic transportation, are considered. As the surfaces of a hexahedron are more prone to being curved compared to those of other elements in hybrid meshes, hexahedral meshes are adopted here. This helps examine the robustness of the KYZ method in handling non-planar faces. Fig. 11 shows the movement of bubbles in a fluidized bed with two immersed tubes. In this case the grid is severely twisted near the tubes as shown in Fig. 11(a). It yields a big deal of curved cell surfaces. Fig. 12 shows that the degree of non-planarity θ in the grid is very large and some faces with θ ranging from 10 up to 50 are found. Fig. 13 shows the snapshot of solid flow pattern in a dense-phase pneumatic conveying system with roping and cluster phenomena. Due to its complexity, high grid quality has to be used. It can be seen from Fig. 14 that θ in the grid is small, and only up to 1.2. The two successful applications indicate that the KYZ method is robust enough to handle complex geometry and overcome the non-planar problem encountered in the existing face-by-face search method.

5. Conclusions

We have introduced and evaluated a new directed point-locating scheme valid for 3D hybrid meshes. The method is developed by combining the advantages of the existing point-locating schemes. Its performance is demonstrated from numerical experiments involving the use of traditional Lagragian particle tracing method and CFD–DEM model. The numerical results show that the proposed method is not only robust enough to handle point location under complex geometries but also has a smaller CPU-time requirement than the fastest point-locating methods such as the ZL and CBF methods. The proposed particle-locating algorithm can be readily integrated into a 3D structured/unstructured fluid flow solver because of its unsophisticated design principles.

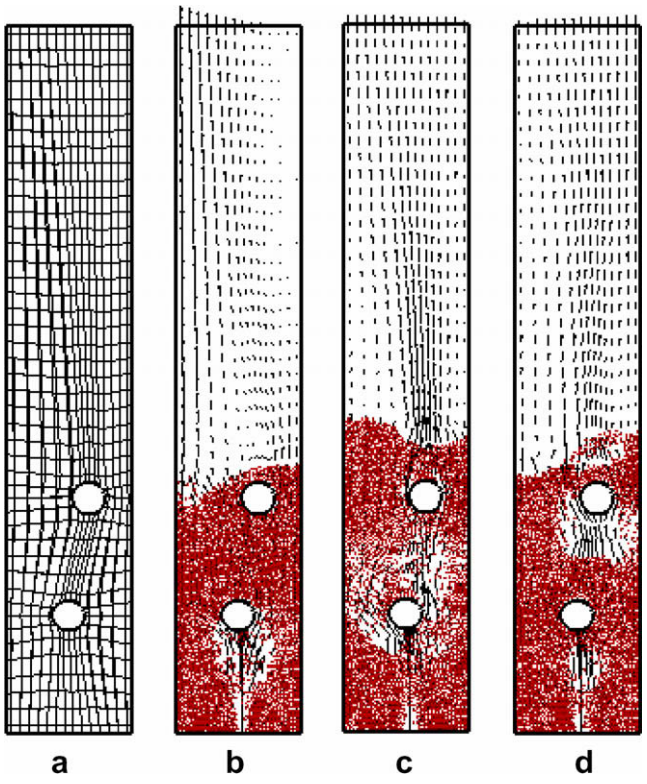

Fig. 11. Bubble movement around immersed tubes in a fluidized bed: (a) meshes and gas–solid flow patterns at (b) time = 1.275 s, (c) time = 1.44 s, and (d) time = 1.665 s.

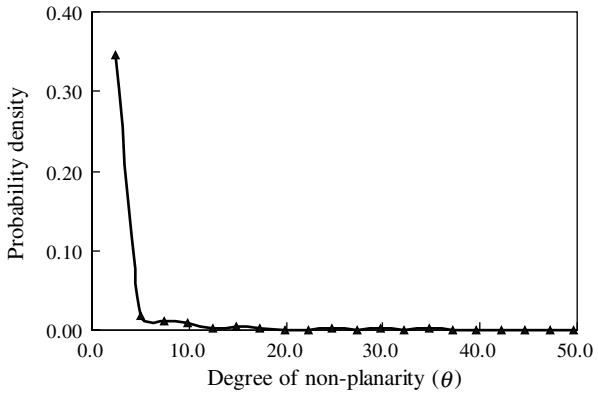

Fig. 12. Probability density distribution of the degrees of non-planarity in the grid shown in Fig. 11.

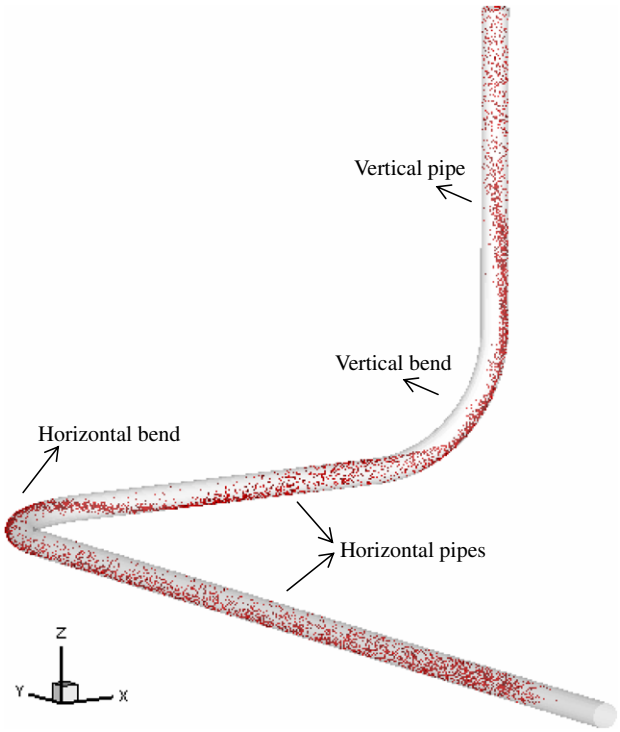

Fig. 13. High-velocity gas–solid flow in a typical pneumatic conveying system.

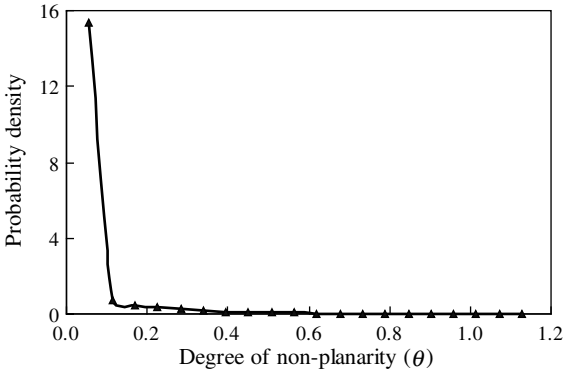

Fig. 14. Probability density distribution of the degrees of non-planarity in the grid shown in Fig. 13.

Acknowledgements

The authors are grateful to the Australia Research Council (ARC) for the financial support of this work.

References

Albertelli, G., Crawfis, R.A., 1997. Efficient subdivision of finite-element datasets into consistent tetrahedra. In: Proceedings of the 8th Conference on Visualization'97, Phoenix, Arizona, United States, pp. 213–219.

Apte, S.V., Mahesh, K., Moin, P., Oefelein, J.C., 2003. Large-eddy simulation of swirling particle-laden flows in a coaxial-jet combustor. Int. J. Multiphase Flow 29, 1311–1331.

Chen, X.Q., 1997. Efficient particle tracking algorithm for two-phase flows in geometries using curvilinear coordinates. Numer. Heat Transfer A 32, 387–405.

Chen, X.Q., Pereira, J.C.F., 1999. A new particle-locating method accounting for source distribution and particle-field interpolation for hybrid modeling of strongly coupled two-phase flows in arbitrary coordinates. Numer. Heat Transfer B 35, 41–63.

Chordá, R., Blasco, J.A., Fueyo, N., 2002. An efficient particle-locating algorithm for application in arbitrary 2D and 3D grids. Int. J. Multiphase Flow 28, 1565–1580.

Chu, K.W., Yu, A.B., 2008. Numerical simulation of complex particle-fluid flows. Powder Technol. 179, 104–114.

Crowe, C.T., Sharma, M.P., Stock, D.E., 1977. Particle-source in cell (psi-cell) model for gas-droplet flows. J. Fluid Eng. Trans. ASME 99, 325–332.

Deguzman, M.M., Fletcher, C.A.J., Behnia, M., 1995. Gas-particle flows about a cobra probe with purging. Comput. Fluids 24, 121–134.

Dompierre, J., Labbé, P., Vallet, M.G., Camarero, R., 1999. How to subdivide pyramids, prisms and hexahedra into tetrahedra. In: Proceedings of the 8th International Meshing Roundtable, Sandia National Laboratories, Albuquerque, NM, pp. 195–204.

Feng, Y.Q., Xu, B.H., Zhang, S.J., Yu, A.B., Zulli, P., 2004. Discrete particle simulation of gas fluidization of particle mixtures. AIChE J. 50, 1713–1728.

Ferziger, J., Peric, M., 2002. Computational Methods for Fluid Dynamics. Springer, New York.

Frank, T., Schulze, I., 1994. Numerical simulation of gas-droplet flow around a nozzle in a cylindrical chamber using Lagrangian model based on a multigrid Navier–Stokes solver. In: International Symposium on Numerical Methods for Multiphase Flows, Lake Tahoe, NV, United States.

Garrity, M.P., 1990. Raytracing irregular volume data. In: Proceedings of the 1990 workshop on Volume visualization, San Diego, California, United States, pp. 35–40.

Haegland, H., Dahle, H.K., Eigestad, G.T., Lie, K.A., Aavatsmark, I., 2007. Improved streamlines and time-of-flight for streamline simulation on irregular grids. Adv. Water Resour. 30, 1027–1045.

Haselbacher, A., Najjar, F.M., Ferry, J.P., 2007. An efficient and robust particle-localization algorithm for unstructured grids. J. Comput. Phys. 225, 2198–2213.

Kenwright, D.N., Lane, D.A., 1995. Optimization of time-dependent particle tracing using tetrahedral decomposition. In: Proceedings of the 6th Conference on Visualization'95, IEEE Computer Society, pp. 321–328.

Kuang, S.B., Chu, K.W., Yu, A.B., Zou, Z.S., Feng, Y.Q., 2008. Computational investigation of horizontal slug flow in pneumatic conveying. Ind. Eng. Chem. Res. 47, 470–480.

Li, G., Modest, M.F., 2001. An efficient particle tracing scheme for structured/unstructured grids in hybrid finite volume/PDF Monte Carlo methods. J. Comput. Phys. 173, 187–207.

Liu, Y., Vinokur, M., Wang, Z.J., 2006. Spectral (finite) volume method for conservation laws on unstructured grids v: extension to three-dimensional systems. J. Comput. Phys. 212, 454–472.

Lohner, R., 1995. Robust, vectorized search algorithms for interpolation on unstructured grids. J. Comput. Phys. 118, 380–387.

Lohner, R., Ambrosiano, J., 1990. A vectorized particle tracer for unstructured grids. J. Comput. Phys. 91, 22–31.

Milgram, M.S., 1989. Does a point lie inside a polygon?. J. Comput. Phys. 84, 134–144.

Muradoglu, M., Kayaalp, A.D., 2006. An auxiliary grid method for computations of multiphase flows in complex geometries. J. Comput. Phys. 214, 858–877.

Oliveira, P.J., Gosman, A.D., Issa, R.I., 1997. A method for particle location and field interpolation on complex, three-dimensional computational meshes. Adv. Eng. Software 28, 607–614.

Patankar, S.V., Karki, K.C., 1999. Calculation of particle trajectories in complex meshes. Numer. Heat Transfer B 35, 431–437.

Sadarjoen, I.A., Walsum, T.V., Him, A.J.S., Post, F.H., 1997. Particle tracing algorithms for 3D curvilinear grids. In: Scientific Visualization, Overviews, Methodologies, and Techniques, IEEE Computer Society, pp. 311–335.

Sadarjoen, I.A., Boer, A.J.D., Post, F.H., Mynett, A.E., 1998. Particle tracing in σ-transformed grids using tetrahedral 6-decomposition. In: Proceedings of the Eurographics Workshop, Blaubeuren, Germany, pp. 71–80.

Schafer, F., Breuer, M., 2002. Comparison of *c*-space and *p*-space particle tracing schemes on high-performance computers: accuracy and performance. Int. J. Numer. Meth. Fluids 39, 277–299.

Seldner, D., Westermann, T., 1988. Algorithms for interpolation and localization in irregular 2D-meshes. J. Comput. Phys. 79, 1–11.

Subramaniam, S., Haworth, D.C., 2000. A probability density function method for turbulent mixing and combustion on three-dimensional unstructured deforming meshes. Int. J. Eng. Res. 1, 171–190.

Vaidya, A.M., Subbarao, P.M.V., Gaur, R.R., 2006. A novel and efficient method for particle locating and advancing over deforming, nonorthogonal mesh. Numer. Heat Transfer B 49, 67–88.

Xu, B.H., Yu, A.B., 1997. Numerical simulation of the gas–solid flow in a fluidized bed by combining discrete particle method with computational fluid dynamics. Chem. Eng. Sci. 52, 2785–2809.

Zhou, Q., Leschziner, M.A., 1999. An improved particle-locating algorithm for Eulerian–Lagrangian computations of two-phase flows in general coordinates. Int. J. Multiphase Flow 25, 813–825.

Zhu, H.P., Zhou, Z.Y., Yang, R.Y., Yu, A.B., 2007. Discrete particle simulation of particulate systems: theoretical developments. Chem. Eng. Sci. 62, 3378–3396.