Journal of Computational Physics 228 (2009) 7357–7367

Contents lists available at ScienceDirect

Journal of Computational Physics

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

A set of particle locating algorithms not requiring *face belonging to cell* connectivity data

M. Sani, M.S. Saidi *

Center of Excellence in Energy Conversion, School of Mechanical Engineering, Sharif University of Technology, IR of Iran, P.O. Box 11155-9567, Tehran, Iran

ARTICLE INFO

Article history:
Received 28 January 2009
Received in revised form 11 May 2009
Accepted 24 June 2009
Available online 1 July 2009

Keywords:
Particle locating
Face belonging to cell connectivity
Host cell
Arbitrary polyhedral grid

ABSTRACT

Existing efficient directed particle locating (host determination) algorithms rely on the *face belonging to cell relationship* (F2C) to find the next cell on the search path and the cell in which the target is located. Recently, finite volume methods have been devised which do not need F2C. Therefore, existing search algorithms are not directly applicable (unless F2C is included). F2C is a major memory burden in grid description. If the memory benefit from these finite volume methods are desirable new search algorithms should be devised. In this work two new algorithms (line of sight and closest cell) are proposed which do not need F2C. They are based on the structure of the sparse coefficient matrix involved (stored for example in the compressed row storage, CRS, format) to determine the next cell. Since F2C is not available, testing a cell for the presence of the target is not possible. Therefore, the proposed methods may wrongly mark a nearby cell as the host in some rare cases. The issue of importance of finding the correct host cell (not wrongly hitting its neighbor) is addressed. Quantitative measures are introduced to assess the efficiency of the methods and comparison is made for typical grid types used in computational fluid dynamics. In comparison, the closest cell method, having a lower computational cost than the family of line of sight and the existing efficient maximum dot product methods, gives a very good performance with tolerable and harmless wrong hits. If more accuracy is needed, the method of approximate line of sight then closest cell (LS-A-CC) is recommended.

1. Introduction

Particle localization or host cell determination is a problem defined on a grid as: *given a set of coordinates for a point (target), determine the grid cell containing it (host cell).* The problem information may be accompanied by a guess for the potential host cell (potential cell). This problem should be solved in many cases of practical interest in computational fluid dynamics, including Lagrangian particle tracking procedures, over-set grid simulations, particle in cell methods, immersed boundary applications, thermal or material source inclusion procedures and free surface flow modeling.

To solve this problem, algorithms usually follow this procedure

1. Provide a cell as the potential cell.
2. Perform some kind of in-cell test on the potential cell and determine if the potential cell is the host cell.
3. If not, propose a new potential cell and follow the procedure from step 2 until the host cell is found.

* Corresponding author.
E-mail addresses: msani@mech.sharif.edu (M. Sani), mssaidi@sharif.edu (M.S. Saidi).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.06.031

Usually the algorithm receives a guess to start with or it begins from the cell number one or a random cell number to initialize the procedure (step 1).

As an example, the most primitive and well known algorithm called brute-force begins from the first cell of the grid, performs the in-cell test on it and if it is not the host cell, it assumes the cell whose number follows the current cell in the grid description as the next potential cell. As the name shows it performs the in-cell test cell by cell blindly, which of course, is the reason for its poor efficiency. Although inefficient, it always finds the correct cell. The amount of the computations becomes prohibitively large for large number of cells and large number of target locations. Therefore, this method is not usually the first choice but is used some times as the fall-back algorithm.

Because of the inefficiency associated with the primitive brute-force algorithm new algorithms emerged (which we call them directed search algorithms). They all try to find the next potential cell based on the target position and current cell topology. It should be noted that a modified version of the brute-force algorithm was introduced by Apte et al. [1] which should not be considered as a directed search method. In their method, instead of performing the in-cell test for all of the cells, the distance of the centroid of all domain cells to the target is computed and the closest cell is identified. Then that cell and its neighbors are considered for the in-cell test. If all of these cells fail in the in-cell test, the algorithm reverts back to the original brute-force algorithm.

Lohner and Ambrosiano [2] presented a method based on the linear shape functions (in FEM) which later Lohner and Ambrosiano [3] called it the known vicinity algorithm. Zhou and Leschziner [4] proposed a method called particle-to-the-left for convex cells. In their method a face based cross product technique was used to investigate if the particle is outside the cell with regards to each face. If any face fails in the test (which means the particle is not in the cell), the cell sharing the face is assumed to be the next potential cell and algorithm is repeated. The directed search method of Chen and Pereira [5] follows the same particle-to-the-left test for the faces of the potential cell but does not decide about the next potential cell upon hitting a face failing the test. Instead it composes a list of test-failed faces and searches for the exit face (the face which intersects the particle path). Then it marks the corresponding cell as the next potential cell. Chorda et al. [6] has compared both methods and modified the cost intensive intersection determination procedure in the directed search method by a trajectory-to-the-left test which is similar to the particle-to-the-left test. In the method introduced by Li and Modest [7] for triangular grids, the next potential cell was determined using the length of the line drawn from the old particle location normal to the face and comparing it with the projected length of the particle path on the line normal to the face. Kuang et al. [8] proposed a different methodology based on the sum of the partial volumes. Martin et al. [9] proposed a method based on the dot product of the face normal vector and face center to particle position vector to perform the in-cell test and to decide about the next potential cell. Two options were introduced. In the first one, FPDP, the first face of the cell who gives a positive dot product is used and the cell attached to it is introduced as the next potential cell. In the second one, MPDP, the face who has the maximum dot product is used to determine the next potential cell. Haselbacher et al. [10] proposed another method based on the trajectory intersection method using the parametric representation of the faces. In this work, the MPDP method of Martin et al. [9], which is low cost and effective, is used for comparison.

All of the aforementioned methods need the *face belonging to cell relationship (F2C)* to decide about the next potential cell (usually the neighbor cell sharing a certain face with the current potential cell) and to perform the in-cell test. For example, the MPDM method needs to know the faces of the current potential cell, so F2C, to construct the vector from the face center to the target. The other vector used in MPDM is the face normal vector which knowing the current potential cell needs F2C data to be determined. This F2C data is expensive to store because it is a list which dimension matches the number of cells in the grid and each element in the list, corresponding to a cell, is a list by itself containing the indexes of the faces of the cell. If the flow solver is not dependent on F2C, the memory cost for grid data storage could be reduced substantially. Removing F2C, on the other hands, makes the application of current search algorithms impossible, as described above for MPDM. However, because the neighbor cells could be identified from the structure of the sparse coefficient matrix, new directed search methods could be designed which work without having F2C available.

To summarize, if the memory benefit of independence on F2C of the flow solver is sought and if a search algorithm (particle locating algorithm) is also needed for some purpose like particle tracking, new search algorithms are needed not because the existing ones are inefficient but because they could not work without F2C. In this work, new particle locating algorithms are developed which are not dependent on F2C. Moreover, some of them are more efficient than the current state of the art methods (a claim to be proved later in this work).

In what follows the directed search algorithms of line of sight and closest cell and their variations and combinations are illustrated. Then their performance in terms of number of cells visited before locating the target, the number and quality of wrong hits and the amount of computational time required are evaluated quantitatively on a variety of typical grids in a square cavity. The importance of accurately locating the host cell is also discussed.

2. Sparse matrix structure, its storage and neighbor finding

Transport equations for flow phenomena are usually numerically approximated by means of finite volume, finite difference or finite element methods. They usually have computational stencils extended only a few (usually just one) computational cells (or nodes) from the cell under consideration. This means that the row in the coefficient matrix related to each computational cell has just a few none-zero elements. For example, using cell-centered second order finite volume

discretization on a tetrahedral grid having N_{cells} cells results to a coefficient matrix which has just five none-zero elements in each row while the same row has N_{cells} elements (other elements are zero). Number of none-zeros for each row equals to the number of faces the corresponding cell has plus one. To exploit this feature, the coefficient matrix is usually stored in a compressed form. The matrix computations are also carried out having this zero pattern in mind. Because the none-zero elements are resulted from the linkage between cells (via their faces), this same structure could be exploited to find the neighbors of a cell. In this work compressed row storage (CRS) and cell-centered polyhedral grid finite volume method are used as the basis but the principles apply equally well to other compressed storage formats and discretization schemes.

For the finite volume method applied on polyhedral grids, the none-zero elements in each row (corresponding to a cell) are the effects of neighboring cells sharing a face with it or the cell itself. Therefore the number of none-zeros in each row equals to the number of neighbors of the cells plus one. In CRS the coefficient matrix is stored with three vectors. The first, a, contains only the none-zero elements of the coefficient matrix, so its length equals to twice the number of faces plus the number of cells $2N_{faces} + N_{cells}$. This is because the none-zero elements correspond to the linkage between different cells (off-diagonal elements) and the effect of each cell in its own equation (diagonal element). The total number of links is equal to the number of faces and each link appears twice. The self contribution terms are evidently equal to the number of cells. The second vector, col_{ind}, with the same length as a contains the column index for the corresponding none-zero elements in a. The third, row_{ptr}, is a pointer to the beginning element of each row in a which then should have a length equal to N_{cells} but is extended for computational ease with a single element so that the last element contains the total number of none-zeros.

For a given grid this CRS structure remains the same over the simulation unless the grid topology is changed (which could be accounted for very easily). Therefore, it is constructed once and stored to be available for later use. With the above discussion, this structure which was aimed to be used for matrix storage and computations has the potential to be used to identify the neighbors of the cells. For a given cell number p_0, the corresponding data chunk in col_{ind} is addressed directly in row_{ptr} as the indices beginning from $row_{ptr}[p_0]$ and extending to $row_{ptr}[p_0 + 1] - 1$. The corresponding elements in col_{ind} are the cell numbers for the neighbors of p_0 or p_0 itself. To remove the need for checking against retrieving the cell number itself (instead of its neighbors needed in this work), when constructing CRS storage vectors, the diagonal element was stored at the beginning of the data chunk (CRS routines for matrix computations do not care about the indices in col_{ind} being in ascending order). To sum up, the cell numbers of the neighbors of a given cell p_0 are listed in col_{ind} from $col_{ind}[row_{ptr}[p_0] + 1]$ to $col_{ind}[row_{ptr}[p_0 + 1] - 1]$.

Now that the neighbors of the potential cell are available with almost no cost, they should be examined to see which one is more eligible as the next potential cell. This issue is addressed next.

3. Line of sight method (LS) and its variations

This method is based on choosing the next potential cell as the cell whose center is nearest to the line of sight of the target for an observer standing at the center of the current potential cell. First a vector is constructed from the current potential cell center to the target (called line of sight, *LS*). Then the potential cell center is connected to the center of each of its neighbors (NL_{p_j} for jth neighbor). An index of angular closeness (*IAC*) to *LS* is computed for each neighbor based on the angle between *LS* and NL_{p_j} from the inner vector product as:

$$IAC_{p_j} = |(\vec{r}_{target} - \vec{r}_{p_0})| cos(\phi_{p_j}) = \frac{(\vec{r}_{target} - \vec{r}_{p_0}).(\vec{r}_{p_j} - \vec{r}_{p_0})}{|(\vec{r}_{p_j} - \vec{r}_{p_0})|} \quad (1)$$

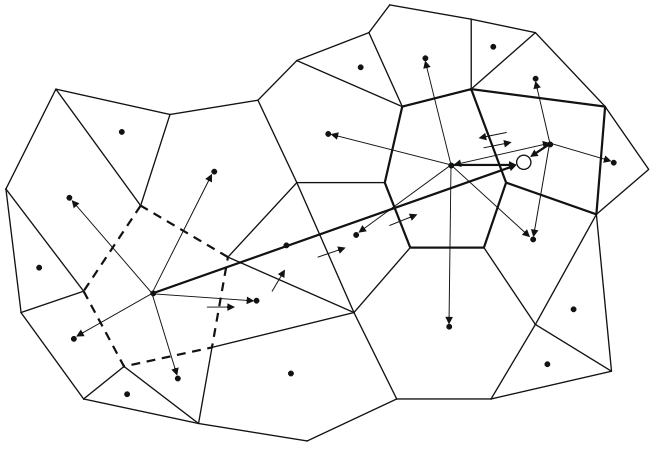


Fig. 1. A polyhedral grid and the sequence of cells chosen by the method of line of sight. The starting cell (guess) is shown with dashed faces and two terminating cells are shown with highlighted faces.

where p_0 is the current potential cell and p_j is its jth neighbor (Fig. 1). Obviously the higher the *IAC*, the closer the corresponding neighbor to the line of sight. The search is terminated when the next potential cell becomes the old potential cell (*go and return*). Between the last two cells the cell being closer to the target is chosen as the host cell if there is no F2C available. If F2C is available or it is allowed to be invoked temporarily, instead of checking the distance, the last cell is subjected to the in-cell test and if it fails the other cell is chosen as the host cell.

In order to decrease the computational cost related to the square root evaluation in the denominator of Eq. (1), assuming that the centers of the neighbors are nearly equidistant from the potential cell center, an approximate index of angular closeness is defined as:

$$IAC_{p_j}^{app} = |(\vec{r}_{p_j} - \vec{r}_{p_0})| IAC_{p_j} = (\vec{r}_{target} - \vec{r}_{p_0}).(\vec{r}_{p_j} - \vec{r}_{p_0}) \tag{2}$$

This value could be used instead of IAC_{p_j}.

At the last step, when choosing between the two terminating cells, while not having F2C data, situations may happen where the cells have very different sizes and/or aspect ratios (Figs. 2 and 3). If the target is located near the sharing face, choosing the closest cell may result to a wrong hit. To modify the behavior of the method in this situation, two variations of the method are proposed here. Both methods weight the distance and are suitable in different situations.

In the first variation, which is suitable for different cell volumes but not high aspect ratio grids (Fig. 2), instead of choosing the host cell based on the center-to-target distance, the distance is divided by the cell characteristic length and this weighted distance is used to decide about the host cell. The cell characteristic length is defined for each cell as:

$$\begin{cases} \ell_p = A_p^{1/2} & (2D) \\ \ell_p = V_p^{1/3} & (3D) \end{cases} \tag{3}$$

and the length weighted index of linear closeness, ILC^{LW}, is defined as:

$$ILC_p^{LW} = \frac{(\vec{r}_{target} - \vec{r}_p).(\vec{r}_{target} - \vec{r}_p)}{\ell_p^2} \tag{4}$$

The cell having the smaller ILC^{LW} is chosen as the host cell.

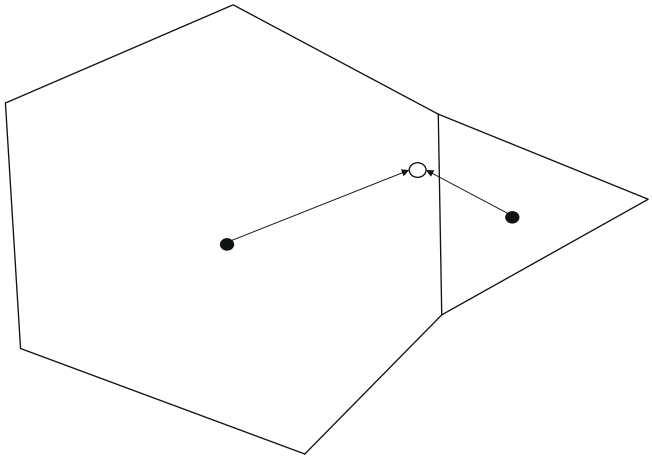


Fig. 2. Choosing between the two terminating cells which have very different sizes in the method of line of sight.

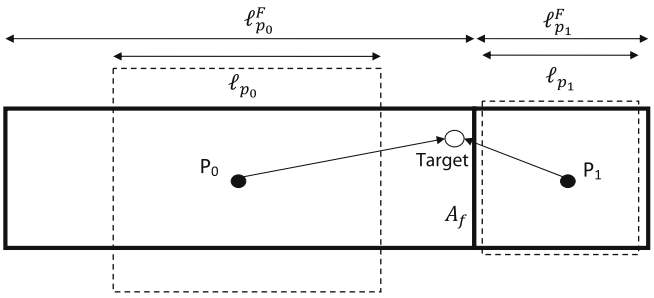


Fig. 3. Choosing between the two terminating cells which have very different sizes and aspect ratios in the method of line of sight.

In the second variation, which is suitable for high aspect ratio grids, the face normal characteristic length is used as the weight. It is obtained by estimating the face area form:

$$A_f = \frac{V_{p_0} + V_{p_1}}{2|\vec{r}_{p_0} - \vec{r}_{p_1}|} \tag{5}$$

Then the face normal characteristic length of each cell is estimated as:

$$\ell_p^F = \frac{V_p}{A_f} \tag{6}$$

and the face normal length weighted index of linear closeness, ILC^{FLW}, is defined as:

$$ILC_p^{FLW} = \frac{(\vec{r}_{target} - \vec{r}_p).(\vec{r}_{target} - \vec{r}_p)}{\ell_p^{F2}} \tag{7}$$

The cell having the smaller ILC^{FLW} is chosen as the host cell.

To make things automatic, the aspect ratio should be determined automatically. This is done by computing the aspect ratio as:

$$AR_p = \left(\frac{\ell_p^F}{\ell_p}\right)^2 \tag{8}$$

A value near one for a quadrilateral cell shows closeness of the cell to the square shape. A large value shows the cell is stretched in the direction normal to the face in between and vice versa.

4. Closest cell method (CC) and its variation

In this method, among the current potential cell and its neighbors, the cell which is closest to the target is chosen as the next potential cell (Fig. 4). The index of linear closeness, *ILC*, for these cells is defined as:

$$ILC_p = (\vec{r}_{target} - \vec{r}_p).(\vec{r}_{target} - \vec{r}_p) \tag{9}$$

The cell having the lowest *ILC* is then chosen as the next potential cell. If the index of linear closeness of the current potential cell, ILC_{p_0}, is lower than that of all of its neighbors, ILC_{p_j}, the search is terminated and p_0 is assumed to be the host cell.

Although this method does not have to decide between the two terminating cells, the same comment about the different cell volumes and high aspect ratio cells applies during the course of next cell determination. Therefore two logical extensions introduced for the method of line of sight could be applied here (cell characteristic length and face normal characteristic length weighting). This means, instead of using *ILC* to find the next potential cell, ILC^{LW} or ILC^{FLW} could be used. Unfortunately, using ILC^{FLW} makes the computational cost unacceptable. This is because the number of A_f values which should be assigned to a cell equals the number of faces it possesses. This requires specific computation and evaluation of a different ℓ_p^F for different faces of a single cell. Therefore, this is not used in this work.

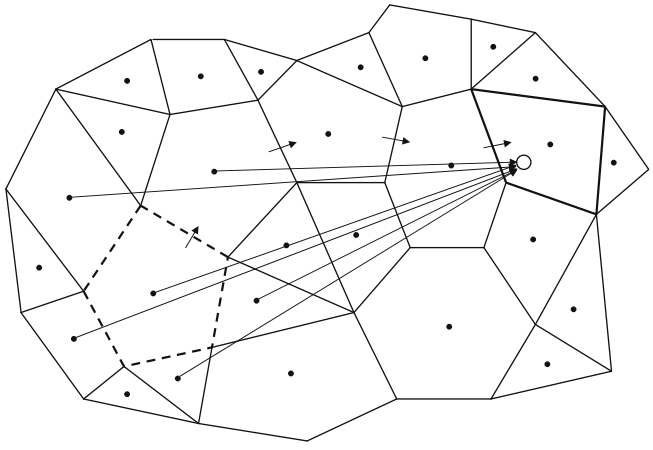


Fig. 4. A polyhedral grid and the sequence of cells chosen by the method of closet cell. The starting cell (guess) is shown with dashed faces and the host cell chosen by the method with highlighted faces.

5. Methods comparison strategy

The methods described above were implemented into a code to be tested and compared. Four meshes were generated on the unit square ($0 < x < 1$ and $0 < y < 1$) which had different cell topologies typical to fluid simulations (Fig. 5). A population of 1000 particles was generated on each grid. For each case, correct host cells were identified using MPDP algorithm [9]. Table 1 lists the abbreviations used to represent the results which are detailed in Tables 2–5.

To compare the methods, some quantitative features were extracted. The first is the average number of cells visited (Avg CV) before locating the host. The second is the number of wrong hits (WH) or unsuccessful localizations. To understand how wrong the result is, two quantities are introduced. The average normalized distance of the target location to the correctly identified host cell center is one of them. The normalization is based on the characteristic length of the host cell and averaging is carried out based on the L2 vector norm as:

$$\langle d_n^{L2}\rangle_{correct} = \sqrt{\sum^{N_{correct}} \left(\frac{|\vec{r}_{target} - \vec{r}_p|}{\ell_p}\right)^2 \Big/ N_{correct}} \tag{10}$$

where, as is implied by $N_{correct}$, the summation is carried over the correct hits.

The next parameter is the average normalized distance of the target location to the *correct* host cell center for wrong hit cases. Again normalization is based on the characteristic length of the *correct* host cell and averaging is carried out based on the L2 vector norm as:

$$\langle d_n^{L2}\rangle_{wrong} = \sqrt{\sum^{N_{wrong}} \left(\frac{|\vec{r}_{target} - \vec{r}_{correct}|}{\ell_{correct}}\right)^2 \Big/ N_{wrong}} \tag{11}$$

where, as is implied by N_{wrong}, the summation is carried over the wrong hits.

To make these measures comparative, the value of $\langle d_n^{L2}\rangle_{correct}$ for MPDP method which always locates the correct host cell, $\langle d_n^{L2}\rangle_{MPDP}$, is considered as the base and the value for other methods relative to this base is presented as percentage in the corresponding columns, $\langle d_n^{L2}\rangle_{c\%}$ and $\langle d_n^{L2}\rangle_{w\%}$. The value corresponding to MPDP method shows that on average how far the targets are distributed in the cells relative to the cell dimensions. For correct hits the higher the $\langle d_n^{L2}\rangle_{c\%}$, the more the method is successful to find more distant targets. For wrong hits, $\langle d_n^{L2}\rangle_{w\%}$ shows the average relative distance required for the method to fail, hence, the quality of its error. Higher values of this percentage for a method show that the method is able to capture the correct locations unless for very distant targets relative to their correct host cells.

A figure of merit (f_m) is defined which is aimed to measure the accuracy and the amount of computations required by each method. The effect of accuracy could be encoded by the number of correct hits (relative to the total number targets) and the mean distance of them (relative to that of MPDP). The computational costs could be encoded by RCT, relative computational time (relative to MPDM). The f_m becomes:

$$f_m = \frac{N_{correct}}{N_{targets}} \times \frac{\langle d_n^{L2}\rangle_{correct}}{\langle d_n^{L2}\rangle_{MPDP}} \times \left(\frac{1}{RCT}\right)^{\gamma} \tag{12}$$

where γ is used to weight the effect of computational time. For MPDP, f_m becomes unity and values greater than this indicate a better method on this scale. In this work $\gamma = 1$ is used. The method requires less computational effort than MPDP as RCT becomes smaller than one, therefore RCT values less than one are more desirable. It should be noted that RCT does not include the memory cost. Therefore, although f_m could be used to evaluate the relative performance of the methods introduced herein, it should be modified to reflect the memory costs when the methods are to be compared with other methods that use F2C.

6. Discussion

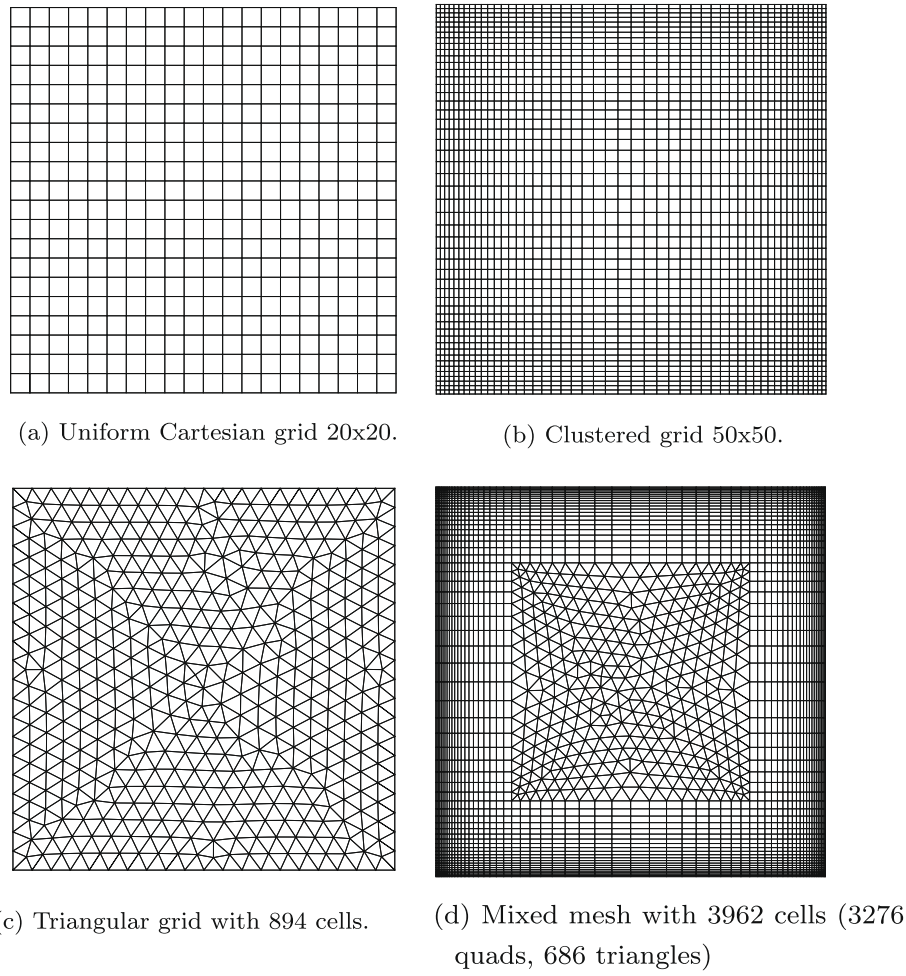
Since all of the methods introduced herein fail to identify the correct host cell to some degree, their mode of failure is discussed first.

Uniform Cartesian grid (Fig. 5(a)): Referring to Table 2, none of the methods failed on the equidistance orthogonal grid. They differ in the amount of computations required. CC was the best having a figure of merit (f_m) of 1.2. LS-A schemes followed it with $f_m = 1$. The high costs of the evaluation of the vector length in the denominator of Eq. (1) made the computational cost of LS schemes high, hence, reducing their f_m to 0.6.

Clustered grid (Fig. 5(b)): On the clustered grid, Table 3, the family of line of sight methods failed more frequently. The approximate version of the method failed even more drastically. Referring to the columns $\langle d_n^{L2}\rangle_{c\%}$ and $\langle d_n^{L2}\rangle_{w\%}$ using the F2C or weighting the distances did not alter the situations. Therefore, the mode of failure was not related to different dimensions of the terminating cells. It should be somehow related to the aspect ratio of the cells which is the major difference between the clustered grid and the uniform one. Referring to Fig. 6, the sequence of cells chosen by the method as the next potential cell is shown. The method fails to find its path towards the correct host cell when it has reached cells numbered 5 and 6. It chooses cell number 5 between these terminating cells. The higher the aspect ratio, the more the probability of this failure mode. Approximate versions of the family are more prone to fail in higher aspect ratio grids because they assume

Table 1
Abbreviations used in the representation of the search methods.

Abbreviation	Meaning
Avg CV	Average number of cells visited
WH	Wrong hits
$\langle d_n^{L2} \rangle_{c\%}$	The ratio of $\langle d_n^{L2} \rangle_{correct}$ to $\langle d_n^{L2} \rangle_{MPDP}$ (Eq. (10))
$\langle d_n^{L2} \rangle_{w\%}$	The ratio of $\langle d_n^{L2} \rangle_{wrong}$ to $\langle d_n^{L2} \rangle_{MPDP}$ (Eq. (11))
RCT	Computational time relative to that of MPDP
BF	Brute force
MPDP	Maximum positive dot product method [9]
LS-F2C	Line of sight with *face to cell relationship* used in last step
LS	Line of sight without *face to cell relationship*
CC	Closest cell
LSCC	Line of sight then closest cell
CCLS	Closest cell then line of sight
LS-A-CC	Approximate line of sight then closest cell
LW	Cell characteristic length weighted
FLW	Face normal characteristic length weighted
A	Approximate



(a) Uniform Cartesian grid 20x20.

(b) Clustered grid 50x50.

(c) Triangular grid with 894 cells.

(d) Mixed mesh with 3962 cells (3276 quads, 686 triangles)

Fig. 5. Meshes used for the evaluation of the methods.

that neighbors are equidistant (Eq. 2). As a result of high aspect ratio, this assumption becomes unacceptable for example for cells 4 and 6 neighboring cell 5. Unfortunately this mode of failure is very dangerous because the high aspect ratio cells are

Table 2
Search results with different algorithms on a uniform 20×20 mesh (Fig. 5(a)) for 1000 random target position and random potential cell guess (for abbreviations refer to Table 1; values of Avg CV, WH(%), $\langle d_n^{L2} \rangle_{c\%}$ and $\langle d_n^{L2} \rangle_{w\%}$ are within $\pm 3\%$ for different samples each having 1000 particles and initial host cell guess pairs, RCT is within $\pm 2.5\%$ for 1000 runs, f_m is correct to one decimal place).

Method	Avg CV	WH(%)	$\langle d_n^{L2} \rangle_{c\%}$	$\langle d_n^{L2} \rangle_{w\%}$	RCT (%)	f_m
BF	192	0	100	–	710	0.14
MPDM	14	0	100	–	100	1
LS-F2C	15	0	100	–	170	0.6
LS	15	0	100	–	170	0.6
LS-LW	15	0	100	–	170	0.6
LS-FLW	15	0	100	–	170	0.6
LS-A	15	0	100	–	95	1.0
LS-A-LW	15	0	100	–	100	1.0
LS-A-FLW	15	0	100	–	100	1.0
CC	14	0	100	–	85	1.2
CC-LW	14	0	100	–	105	1.0
LSCC	16	0	100	–	170	0.6
CCLS	16	0	100	–	110	0.9
LS-A-CC	16	0	100	–	100	1.0

Table 3
Search results on a Cartesian clustered 50×50 mesh (Fig. 5(b); conditions and precisions are the same as Table 2, for abbreviations refer to Table 1).

Method	Avg CV	WH(%)	$\langle d_n^{L2} \rangle_{c\%}$	$\langle d_n^{L2} \rangle_{w\%}$	RCT (%)	f_m
BF	1231	0		–	1975	0.05
MPDM	32	0	100	–	100	1
LS-F2C	33	6.2	95.08	156.6	165	0.5
LS	33	6.9	94.94	152.7	165	0.5
LS-LW	33	6.6	95.06	154.5	165	0.5
LS-FLW	33	6.5	95.06	154.5	165	0.5
LS-A	33	21	87.74	136.6	90	0.8
LS-A-LW	33	20.6	87.93	136.9	90	0.8
LS-A-FLW	33	20.6	87.93	136.9	90	0.8
CC	32	1.5	99.41	133.2	80	1.2
CC-LW	32	0.5	99.86	124.5	100	1
LSCC	35	1.5	99.41	133.2	159.7	0.6
CCLS	34	1.5	99.41	133.2	95	1.1
LS-A-CC	35	1.5	99.41	133.2	90	1.1

Table 4
Search results on a triangular mesh with 894 cells (Fig. 5(c); conditions and precisions are the same as Table 2, for abbreviations refer to Table 1).

Method	Avg CV	WH(%)	$\langle d_n^{L2} \rangle_{c\%}$	$\langle d_n^{L2} \rangle_{w\%}$	RCT (%)	f_m
BF	461	0	100	–	1320	0.08
MPDM	22	0	100	–	100	1
LS-F2C	23	0.3	99.58	194.8	165	0.6
LS	23	3.7	97.43	152.3	170	0.5
LS-LW	23	2.9	97.83	156.1	170	0.6
LS-FLW	23	2.9	97.83	156.1	175	0.5
LS-A	23	3.9	97.28	152.3	100	0.9
LS-A-LW	23	3.1	97.69	155.8	100	0.9
LS-A-FLW	23	3.1	97.69	155.8	105	0.9
CC	22	3.8	97.34	152.5	85	1.1
CC-LW	10	57.5	96.26	102.7	47	0.9
LSCC	25	3.7	97.43	152.3	170	0.6
CCLS	24	3.7	97.43	152.3	110	0.9
LS-A-CC	25	3.7	97.43	152.3	105	0.9

used where the gradients are high. Therefore, choosing a wrong host cell in the direction of alignment of the gradient vector results to a very poor interpolation accuracy. To conclude, method of line of sight is not a good choice where the aspect ratio becomes high.

To deal with the problems discussed above, the line of sight method was followed by the closest cell method which performed well on the same grid. The combination, LSCC, resulted to some more steps to be taken by the search algorithm and it captured the correct host cells to 98.5%. In terms of f_m, the efficiency was improved because of the more correct hits. The same strategy applied to the approximate version, LS-A-CC, resulted to a very efficient scheme with a figure of merit of 1.1 which was equally obtained by CCLS.

Table 5
Search results on a mixed mesh with 3962 cells (Fig. 5(d); conditions and precisions are the same as Table 2, for abbreviations refer to Table 1).

Method	Avg CV	WH(%)	$\langle d_n^{L2} \rangle_{c\%}$	$\langle d_n^{L2} \rangle_{w\%}$	RCT (%)	f_m
BF	2511	0	100	–	2530	0.04
MPDM	41	0	100	–	100	1
LS-F2C	41	20.5	79.71	155.4	160	0.4
LS	41	23.3	78.44	150.5	155	0.4
LS-LW	41	21.4	79.17	14.0	155	0.4
LS-FLW	41	21.4	79.17	154.0	155	0.4
LS-A	40	42.6	75.32	125.8	85	0.5
LS-A-LW	40	41.4	75.70	126.7	85	0.5
LS-A-FLW	40	41.4	75.70	126.7	90	0.5
CC	40	4.9	98.05	132.2	80	1.2
CC-LW	24	55.9	96.29	102.8	65	0.7
LSCC	43	4.8	98.14	131.5	155	0.6
CCLS	42	4.9	98.05	132.2	90	1.0
LS-A-CC	43	4.8	98.14	131.5	90	1.0

On the same grid, CC method performed very well because on a Cartesian grid, wherever cells have equal dimensions, the criterion of *ILC* is perfect. The small amount of failure could be assigned to unequal volume adjacent cells. On these rare cases, weighting the method, modified the probability of wrong hits with a factor of three which proves the assumption about the mode of failure. The very good figure of merit of 1.2 for CC method was decreased to 1.0 by weighting because of the added computational cost. To further improve f_m, the weighting could be applied just at the last step, therefore, cutting the search path evaluation costs related to weighting. The mode of failure associated with CC on high aspect ratio grids is not as dangerous as LS, because the direction of incorrect identification of the host cell is normal to the clustering direction of the grid and low variation of field variables are expected in this direction.

Triangular grid (Fig. 5(c)): CC method failed a bit more than LS on the triangular grid (Table 4). Weighting made situation very worse for CC. This is because the search path was wrongly determined from the initial steps. Weighting improved the situations for LS, because it used distances just for the terminating cells. The remaining wrong hits for LS and all of them for CC could be related to the grid having high skewness in some places (Fig. 7). To prove this claim the LS method was used with F2C. It found host cells in 99.7% of the cases. With F2C the mechanism for deciding between terminating cells changes from distance based to the in-cell test. Therefore, the mode of failure for both LS and CC is related to high skewness of the host cell. Fortunately, this is not too bad (c.f., Section 7).

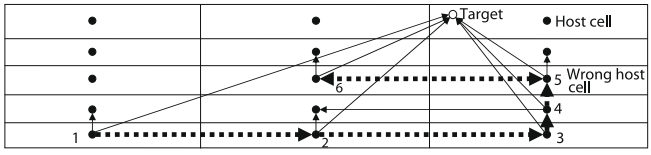


Fig. 6. The failure mode of the LS family of methods.

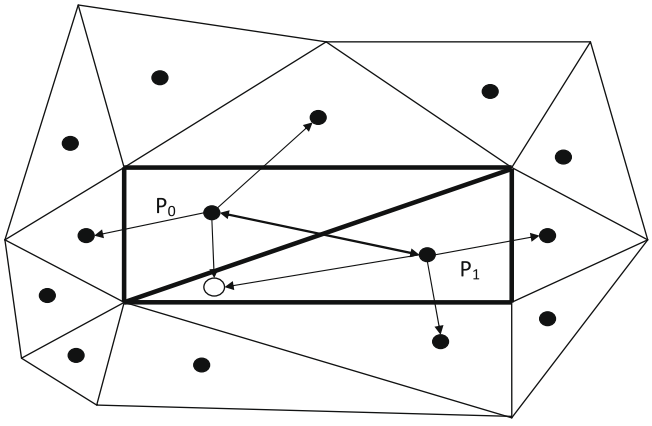


Fig. 7. The CC failure mode and also the mode related to LS when it has to decide between terminating cells based on distance.

Mixed grid (Fig. 5(d)): As a more realistic grid, typical to the fluid simulations, a grid with quadrilateral cells clustered normal to the boundaries and triangular cells in the central portion was constructed. Results for search on this grid are summarized in Table 5. Because there were many high aspect ratio cells, the methods of LS family had a high level of wrong hits (especially the approximate version). CC performed almost perfectly and the error remaining could be attributed to high skewness triangular cells. The LS methods followed by CC improved the behavior of LS and made them as good as CC in capturing the correct host cells. In case of LS-A-CC, it captured more host cells than CC. On the basis of figure of merit rating (assuming $\gamma = 1$), CC without weighting was the best method with f_m equal to 1.2 followed by LS-A-CC and CCLS with f_m equal to 1.

A note on lower computational costs. As is seen in the Tables 2–5, in most cases, CC, LS-A-CC and CCLS methods had less computational costs than MPDP. All these methods require the same number of computations per neighbor in dot product account. MPDP also needs to construct the vector from the face center to the target which needs vector subtraction. This task is face dependent and therefore needs to be done once per face. In CC, LS-A-CC and CCLS methods the vector from the potential cell to the target is constructed once and is used for all of the neighbors. The amount of saving dependens on the number of faces of the potential cell. For a triangular (quadrilateral) cell there are two (three) less vector subtractions. For a polyhedral cell having N_f faces there are $N_f - 1$ saved vector subtractions. For methods of LS-A-CC and CC-LS, in some cases like pure triangular grids, the benefit is out weighted by the more number of cells visited. For CC method, because the number of cell visits were always less or equal to MPDP, the cost was always less.

A note on parallel application: It is worth noting that the methods proposed herein have the same parallel features as MPDM. In finding the path to the host cell, methods proposed herein and MPDP need to evaluate some kind of metric to find the direction to look for the next potential cell. This could be evaluated in parallel for the neighbors (faces in MPDP) of the potential cell. The comparison to decide about the next potential cell needs the same amount of data transfer and serial work. Methods proposed herein and MPDP could be used in parallel equivalently for different targets (particles), because finding the host cells of different targets is a parallel task in nature.

7. How important is hitting the correct host cell?

The answer depends on the application. For example, in Lagrangian particle tracking, the value of fluid velocity should be interpolated to the position of the particle. Two methods are available; velocity profile reconstruction within the cell and weighted interpolation between the nearby cell centers.

If the *velocity reconstruction* is used with *triangular grids*, extrapolating to a position outside a cell would not result to a bad approximation for high skewness cases shown in (Fig. 7). It may be even a better approximation than the value obtained by interpolation using velocity reconstruction inside the correct host cell whose center is far away from the target. Because the failure mode for all of the methods introduced on triangular grid was related to the high skewness cells, it could be concluded that all of them are acceptable for interpolation except the CC method with weighting. Considering the costs, the best method is CC followed by LS-A-LW, LS-A-FLW, LS-A-CC, CCLS and LS-A.

For *velocity reconstruction* on a *highly clustered grid* where there are usually high gradients in the velocity field in the clustered region, even one cell departure from the correct host cell in the gradient direction would result to unacceptable results and extrapolation is destructive. On this account, none of the methods of the LS family could be used unless they are followed by CC. The analysis of the mode of failure of CC shows that in rare cases of its failure, it fails in the safe direction (normal to the gradient) and extrapolation is not so destructive. Weighting at the last step should improve its wrong hit counts because the failure is related to unequal cell volumes mode. The cost analysis shows that CC is the best, followed by LS-A-CC and CCLS.

To sum up, for *velocity reconstruction* on any grid, the best choice is between CC, LS-A-CC and CCLS. The figure of merit of CC, based on $\gamma = 1$ is 10–20% higher. If the correct hitting is more desired, $\gamma < 1$, then LS-A-CC with length weighting may become more acceptable.

For the second option of *weighted interpolation between nearby cell centers*, host cell boundaries become unimportant. Instead, it is important for the interpolation stencil constructed from the identified host cell center and its neighboring cell centers to surround the particle. In this situation, CC is more reliable than the state of the art method MPDP [9] because it always locates the cell center which is closest to the target location and therefore is more suitable to be used in weighed interpolation. CC also needs less computational time. LS-A-CC having the same computational time as MPDP follows CC. Both methods do not need F2C which is an added memory benefit.

It is worth reminding that even if CC was not superior on computational time and memory accounts, it and all of the methods proposed herein do not need F2C which makes them the only available methods which could be used without the penalty of storing F2C.

8. Conclusion

Two new search algorithms of closest cell (CC) and line of sight (LS) for particle localization (host determination) were introduced. A number of modifications and their combinations were discussed. The major feature of the newly introduced methods was their independence to the *face belonging to cell relationship* (F2C). The proposed methods were blind to the

dimensionality of the grid and could be used in three dimensions without any modification. These methods were tested against one of the best methods available in the literature, namely, maximum positive dot product (MPDP). The issue of accuracy (finding the host cell correctly) and reliability (quality of wrongly identified host cells) was addressed in a comparative study. As a result, CC and LS-A-CC were found to be less computational effort and memory demanding than MPDP and more reliable for interpolation on highly skewed grids with weighed interpolation. All of the methods proposed share the feature that they do not need F2C which makes them less memory demanding than the existing methods (the benefit which increases with the size of the grid).

Acknowledgment

This work was carried out in Center of Excellence in Energy Conversion, Sharif University of Technology.

References

[1] S.V. Apte, K. Mahesh, P. Moin, J.C. Oefelein, Large-eddy simulation of swirling particle-laden flows in a coaxial-jet combustor, International Journal of Multiphase Flow 29 (8) (2003) 1311–1331.
[2] R. Lohner, J. Ambrosiano, A vectorized particle tracer for unstructured grids, Journal of Computational Physics 91 (1990) 22–31.
[3] R. Lohner, J. Ambrosiano, Robust, vectorized search algorithms for interpolation on unstructured grids, Journal of Computational Physics 118 (1995) 380–387.
[4] L.M. Zhou, Q. Leschziner, An improved particle-locating algorithm for Eulerian–Lagrangian computations of two-phase flows in general coordinates, International Journal of Multiphase Flow 25 (August) (1999) 813–825, doi:10.1016/S0301-9322(98)00045-7 (13).
[5] X. Chen, J.C.F. Pereira, New particle-locating method accounting for source distribution and particle-field interpolation for hybrid modeling of strongly coupled two-phase flows in arbitrary coordinates, Numerical Heat Transfer, Part B: Fundamentals 35 (1) (1999) 41–63.
[6] R. Chorda, J.A. Blasco, N. Fueyo, An efficient particle-locating algorithm for application in arbitrary 2D and 3D grids, International Journal of Multiphase Flow 28 (9) (2002) 1565–1580.
[7] G. Li, M. Modest, An effective particle tracing scheme for structured/unstructured grids in hybrid finite volume/PDF Monte Carlo methods, Journal of Computational Physics 173 (2001) 187–207.
[8] S.B. Kuang, A.B. Yu, Z.S. Zou, A new point-locating algorithm under three-dimensional hybrid meshes, International Journal of Multiphase Flow 34 (11) (2008) 1023–1030.
[9] G.D. Martin, E. Loth, D. Lankford, Particle host cell determination in unstructured grids, Computers and Fluids 38 (1) (2009) 101–110.
[10] A. Haselbacher, F.M. Najjar, J.P. Ferry, An efficient and robust particle-localization algorithm for unstructured grids, Journal of Computational Physics 225 (2) (2007) 2198–2213.