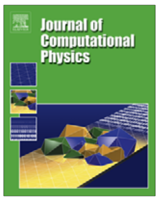
Contents lists available at ScienceDirect

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*

A R T I C L E  I N F O

*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

A B S T R A C T

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