17th Australasian Fluid Mechanics Conference
Auckland, New Zealand
5-9 December 2010

Intersection Marker (ISM) method for tracking a deformable 2D surface in 3D Eulerian Space.

Mark K. Ho[1], Guan H. Yeoh[1,2], Victoria Timchenko[2] and John A. Reizes[2]

[1]Nuclear Operations Division,
Australian Nuclear Science and Technology Organisation (ANSTO), Menai, NSW 2234, Australia

[2]School of Mechanical Engineering,
University of New South Wales, Kensington, NSW 2052, Australia

Abstract

A new method for tracking a deformable interface in 3D Eulerian space has been developed. The method can model an arbitrary 3D shape immersed inside an array of uniform hexahedral control volumes by using a combination of planar polygons. Since each planar polygon intersects the edges of the control volume and the combination of cell-edge intersections uniquely identifies the type of polygon a control volume holds, this new explicit tracking method has been named the '(I)nter(S)ection (M)arker' (ISM) method for interface tracking. The intended application of the ISM method is in CFD simulations of gas-liquid multiphase flows.

Introduction

Numerical simulations of two-phase flows using the finite volume method are known to be difficult. In the case of bubbly flows with distinct interfaces, the challenge lies in the modelling and numerical solution of interface discontinuities in density, pressure, temperature, viscosity and surface tension. The requirement for simulating the detailed aspects of bubble surface evolution is somewhat daunting by virtue of the topological complexity involved. At present, few methods in interface tracking possess sufficient accuracy to directly track the evolution of an arbitrarily deforming bubble interface, especially when coupled with heat and mass transfer. The primary reason for this is a lack of detail in interface modelling which adversely affects calculations that rely on interface area data. Two key calculations: (1) the rate of mass transfer proportional to the interface area within the assessed control volume and (2) the surface tension force as calculated from interface curvature data, have a large influence in the evolution of the bubble topology under transient conditions.

For these reasons, there is an opportunity to make improvements in the detailed interface tracking of two phase flows in computational fluid dynamics (CFD) without the remeshing of the total flow domain for each time step. In a survey of current interface tracking methods, most commerical CFD codes provide the 'piecewise-linear-interface-reconstruction' (PLIC) method [9] or higher-order schemes for surface 'resharpening' which reconstruct the interface after fraction-of-volume (VOF) values have been transported by some general transport equation [4]. Both of these methods make a compromise between surface detail and computational requirement but neither have been demonstrably feasible for heat and mass transfer calculations across interfaces. The reason is partly due to the simplified manner of interface modelling which introduces curvature errors within the surface tension calculations and partly due to the way the surface term is discretised in the momentum equation. Usually the surface tension is modelled as a volumetric force using the continuum surface force (CSF) method of calculation but because of the introduction of a curvature error [2] this gives rise to a force imbalance at the interface which induces rotational structures known as 'parasitic currents' that oscillate the interface and break up the thermal boundary layer.

Tryggvason et al. [5] endeavoured to overcome this by using the Front Tracking method, essentially an immersed boundary method whereby a Lagrangian mesh tracking the evolution of the bubble topology is juxtaposed inside the non-moving Eulerian mesh which is responsible for the finite-volume method of solution. The results have been demonstrated to be better than VOF simulations for heat and mass transfer of bubbly flows and as an affirmation to this, Commissariat à l'énergie atomique (CEA) has adopted Front Tracking in its Trio-U CFD code to numerically investigate the heat and mass transfer of bubbly flows [8]. There are alternatives to tetrahedral-surface models for Front Tracking, most notably the 'Level Set (LS) Method' invented by Osher and Sethian [7] but more recently Aulisa et al. provides an intriguing alternative with its 'Surface marker algorithm coupled to an area preserving marker redistribution method for three-dimensional interface tracking.' [1].

Aulisa's method tracks the interface as a Lagrangian but finds the intersection of the surface mesh with control volume faces and locally remeshes the surface contour whilst preserving the tracked volume. This Lagrangian method has the desirable attribute of being more tightly coupled to the Eulerian mesh whereby the surface intersection markers are anchored to the Eulerian mesh before remeshing. By doing so, Aulisa's method dispenses with global forms of remeshing that are computationally burdensome. On the other hand, Aulisa's Lagrangian method requires permanent markers which cannot be seeded or removed after the simulation starts. The result of this is that for a spherical bubble expansion problem, there will be a deficit in local surface mesh resolution. Thus inspired but also seeking to improve Aulisa's method, a new 3D interface tracking method, named the '(I)nter(S)ection (M)arker (ISM) method for interface tracking in 3D Eularian Space' has been developed. This method seeks to remove the need for permanent markers and address the local surface resolution issue in volume-inflationary type problems. For further reading on the history and development of explicit interface tracking types of two-phase flow simulations, Lakehal's [6] review is recommended.

The Method

The ISM method was created to model an arbitrary 2D surface in 3D space and track its evolution in a vector field with a minimum amount of positional and volumetric errors. To illustrate this method, imagine a curved surface juxtaposed within a cubic control volume (figure 1) and observe how the interface cuts through the cube to produce intersection points at the cube's edges. The interface within a cell can be represented as a flat polygonal whose vertices are coincident with the intersection points on the edges of the cube.

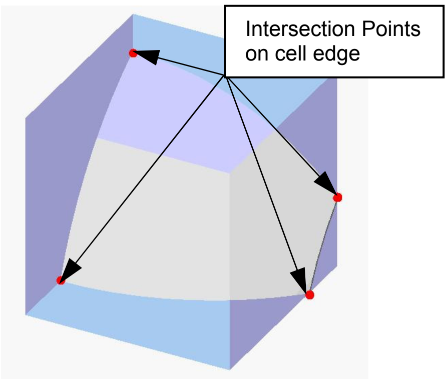

Figure 1. Spherical surface cutting through regular cube-shaped control volumes.

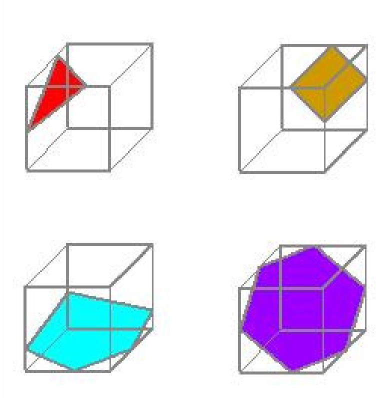

Figure 2. Different polygons generated depending on the orientation of the cutting plane.

As may be seen in figure 2, a plane can cut through a cube to form a polygon with the number of sides varying from three to six. It was also discovered that for every combination of edge point intersections, a polygon of unique orientation is produced and that there are only a finite number of possible combinations. Given that there were a finite number of uniquely orientated polygons, a 'look up' table could be programmed to identify the type of polygons a particular set of edge point intersections would make. From this finding, it was realized that any arbitrary 2D interface may be modeled with a variety of polygons. To demonstrate the feasibility of implementing this method, an algorithm was written in Fortran95 to generate a sphere composed of flat polygons whose vertices intersected with the cube cell edges.

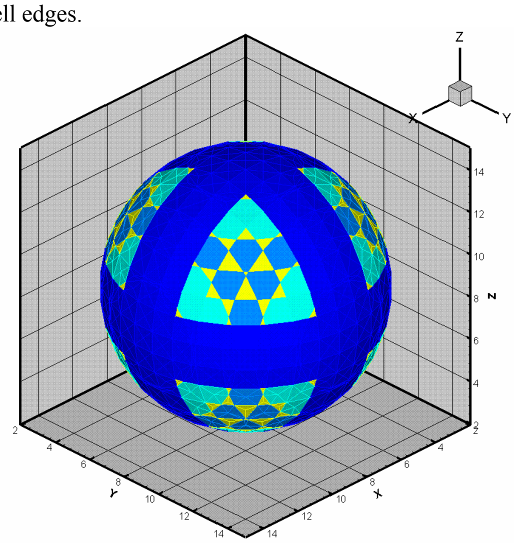

Figure 3. Spherical surface generated by an assortment of polygons. Control volumes are referenced on the x, y, z axes but are not shown..

Figure 3 displays the composite surfaces in colours which correspond to their number of sides. (yellow =3 sided, blue =4 sided, light-blue =5 sided, sky-blue = 6 sided.) Although the spherical interface may be represented by simple polygons, the polygon vertices themselves could not be directly advected by randomly pointing streamline vectors because the translated vertices would not necessarily form a flat plane. The answer to this is to further subdivide the polygon into triangles before the interface is advected because a triangle will always remain a plane, regardless of the orientation of the velocity vectors. There were a number of open options for subdividing a polygon. The simplest method was to connect the 2 points of each side of the polygon to the polygon centroid as shown in figure 4.

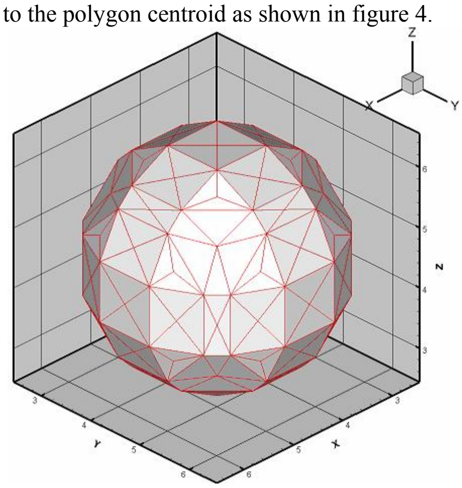

Figure 4. Simple polygon subdivision
Sphere radius = 2h
where h = width of cube shaped control volumes.

However, there is no reason to subdivide the polygon so simply because the need to subdivide presents an opportunity to include more detail in the tracked interface. A logical addition to detail was to project the centroid to the real curved spherical surface. The result of plotting a raised centroid for each polygon is demonstrated in figure 5.

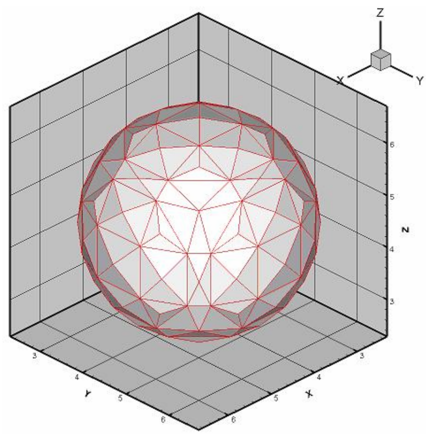

Figure 5. Polygon subdivision with projected polygon centroids for a sphere of radius 2h.

It is readily apparent when comparing figures 4 and 5 that the projection of the centroid of the polygon onto the sphere's surface in figure 5 gives a better result than the simple subdivision method in figure 4. However, this can be improved further. More detail may be introduced if the curved lines joining two edge points are approximated by three connected lines as may be seen in figure 6.

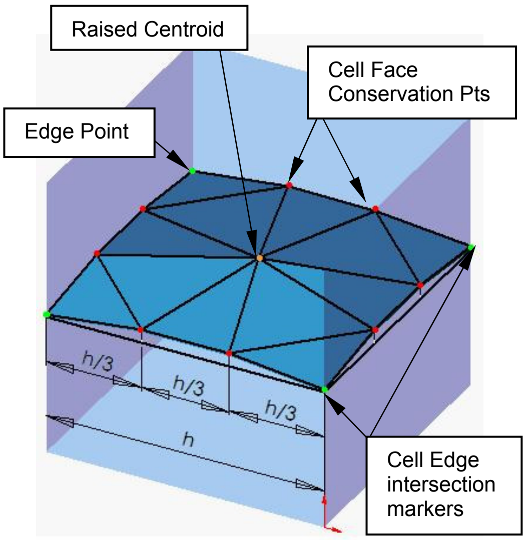

Fig. 6. Complex polygon subdivision with projected polygon centroid and cell face conservation points.

A line connecting two edge points is broken into 3 equal lengths and the points at 1/3 distance on either side of the cube edges are projected to meet the real surface. The interface triangles of the subdivided quadrilateral are then generated from the 8 cell face conservation points and a single projected centroid as shown in figure 6. The resulting, much smoother surface of the complex subdivision shown in figure 7 is a significant improvement over the initial simple subdivision method shown in figure 4 and the improvement generated by the projected polygon centroid which led to figure 5.

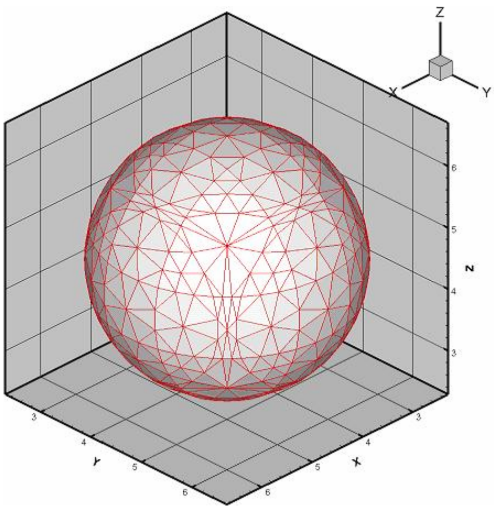

Fig. 7. Complex polygon subdivision for a sphere of radius 2h.

When a model of a sphere is initially generated, the strategy to subdivide polygons of 3 to 6 sides remains the same. Lines connecting cell edges are divided into thirds and the points at 1/3 distance from either side of the cube edges are projected upwards along the cell face to meet the interface surface. The resulting patterns of this complex form of subdivision are displayed in figure 8. With a complex representation of the polygon interface within each cube, surface curvature information within one cube can be calculated to a higher degree of accuracy. Also, the volume underneath the interface can be easily calculated by summing the volumes of all triangle-base columns within the cell. The contours of fraction-of-volume f values in the range $0 \le f \le 1$ within interface-cells are displayed in figure 9, with dark patches signifying an f value approaching zero and light areas signifying an f value approaching 1.

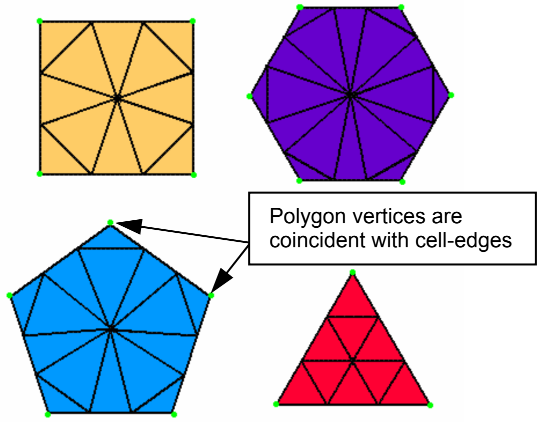

Fig. 8. Polygon subdivision strategies for complex interface representation.

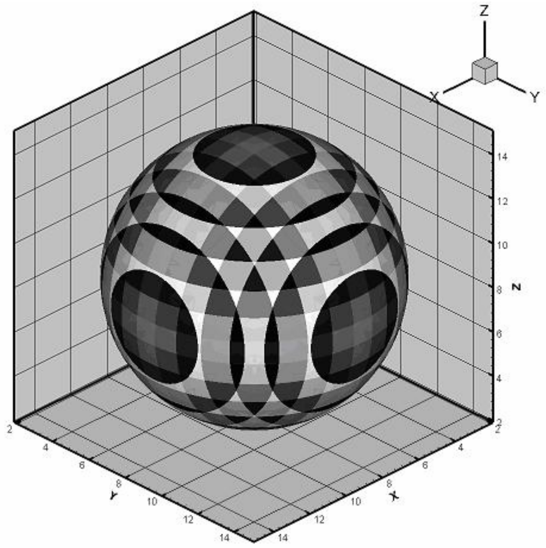

Fig. 9. Complex polygon model of sphere of radius 6h. Contours represent each cell's VOF where: dark ≈ VOF of 0, light ≈ VOF of 1.

Interface Advection

The method for interface advection is to use the calculated velocity field and individually advect each of the points of the small interface triangles at time 't0' to a new position at time 't1'. Velocity vectors for interface points inside the control volume are calculated using the tri-linear interpolation method as outlined by Aulisa [1]. The type of polygon residing in the 'target' cell at time 't1' is determined by the combination of new cell-edge intersection points the advected interface triangles make with the target-cell's edges. (figure 10) New cell-face points are formed by the intersections of advected triangle edges and the target-cell's faces. What is now left are the new cell edge intersection points and many new cell-face points and triangular vertices inside the target cell.

These excess points need to be removed during the remeshing process. To do this, the volume underneath the composite surface is first evaluated by summing the volumes of the triangular and other polygonal columns underneath the interface. Finding the total volume is necessary for the calculation of the final position of the 'raised centroid' as shown in figure 6.

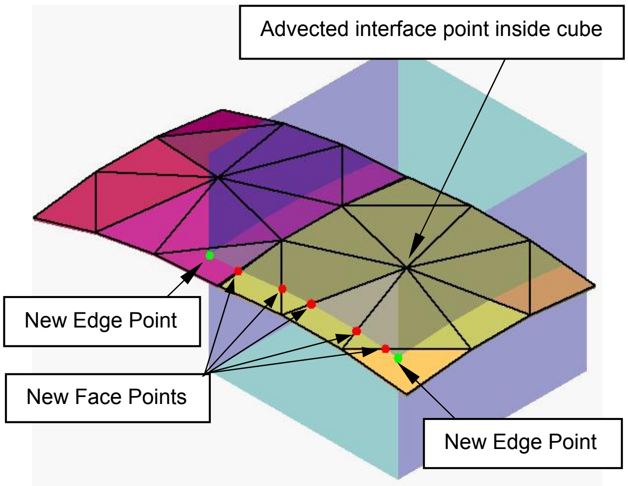

Figure 10. New edge points and face points in the 'target cell' at time t1.

Next the excess cell face points must be removed and re-seeded. This is done by calculating the blue 2D area (figure 11) underneath the intersection lines formed on the cell faces and redrawing an identical cell face area with two new cell-face conservation points. The postion of these points are uniquely calculated at 1/3 distance from either side of the cell edge points and projected perpendicular to the line joining the two cell edge points and along the cell face to form a new area which is identical in size to the old area in blue (figure 11). The height of this perpendicular projection is identical for both cell face conservation markers.

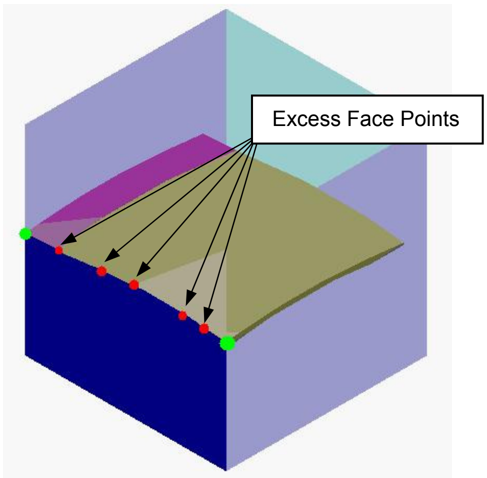

Figure 11. Excess face points on the cell face are removed and reseeded with 2 cell-face area 'conservation points'.

Once all new cell-face points have been established, the point which remains to be calculated is the position of the new 'raised centroid' (figure 6.) Its position is calculated uniquely by first determining the centroid of the new polygon and then adjusting the height of the raised centroid so that the volume underneath the smoothed interface is equal to the volume underneath the initially advected interface. In this manner the volume underneath the advected interface is conserved during the interface resmoothing process. The local remeshing process is repeated for all interface cells during advection.

Conclusion

A new method for tracking a deformable interface in 3D Eulerian space called the (I)nter(S)ection (M)arker (ISM) Method has been presented. The method can model an arbitrary 3D shape immersed inside an array of uniform hexahedral control volumes by using a combination of 2D polygons. These polygons are further subdivided into sub-element triangular interfaces to better model the surface curvature within each cell. The ISM method was designed as an alternative Front-Tracking method that is characterised by a tightly coupled exchange of information between the Lagrangian and Eulerian Mesh. At this stage of development, the ISM method cannot handle coalescence and breakage implicitly as can the FronTieR tracking code [3]. However, the ISM method does model the interface as a continuous surface to a high degree of detail. The degree of detail requires less Eulerian mesh to achieve comparable levels of modelled surface curvature as compared to competing Lagrangian methods (LS not included). The other qualities of this method is that its remeshing is conducted locally instead of globally; it is not as CFL-restrictive as with the LS method and that it is unnecessary to use the distance function for the evaluation of interface curvature. The downside to modelling the interface to such detail is that it requires a large computational memory to keep track of the many interface points and extra subroutines would be necessary to handle coalescence and breakage actions. The ISM method also requires a comparatively large programming effort to implement. An animation of the interface tracking method will be shown during the conference oral presentation to demonstrate the evolution of the interface, polygon patterns and VOF contours as the tracked volume is transported along a periodic streamline function.

Acknowledgements

Thanks go to Dr. Aulisa et al. who inspired this work and to Dr. S. Cummins who pointed it out back at IHTC-13 2006. This work would not have been possible without the approval of Dr. Greg Storr, general manager of Nuclear Operations, ANSTO and Dr. George Braoudakis, leader of Nuclear Analysis Section. Thank you for your support and encouragement.

References

[1] Aulisa E., Manservisi, S. and Scardovelli R., A Surface Marker Algorithm Couple to Area-Preserving Marker Redistribution Method for Three-Dimensional Interface Tracking, *J. Computational Physics* **197** (2004), pp. 555-584.

[2] Cummins S., Francois M. and Kothe D. Estimating Curvature From Volume Fractions, *Computers and Structures* **83** n 6-7 (2005), pp. 425-434, *Frontier of Multi-Phase Flow Analysis and Fluid-Structure*.

[3] Du J., Fix B., Glimm J., Jia X., Li X., Li, Y. and Wu L. A Simple Package for Front Tracking *J. Computational Physics* **213** (2006), pp. 613-628.

[4] Hirt C.W. and Nichols. B.D., Volume of Fluid VOF Method for the Dynamics of Free Boundaries, *J. Computational Physics* **39** (1981), pp. 201-225.

[5] Juric D., Tryggvason G. Computations of Boiling Flows, *Int. J. Multiphase Flow* **24** n 3 (1998), pp. 387-410.

[6] Lakehal D., Meier M., Fulgosi M., Interface Tracking Towards the Direct Simulation of Heat and Mass Transfer in Multiphase Flows, *Int. J. Heat and Fluid Flow* **23** (2002), pp. 242-257.

[7] Sethian J., Evolution, Implementation and Application of Level Set and Fast Marching Methods for Advancing Fronts, *J. Computational Physics*. **169** (2001), pp. 463–502.

[8] Toutant A., Chandesris M., Jamet D., Lebaigue O., Jump Conditions for Filtered Quantities at an Under-Resolved Discontinuous Interface Part 1: Theoretical Development, *Int. J. Multiphase Flow* **35** n 12 (2009), pp. 1100-1118.

[9] Youngs D. Time-Dependent Multi-Material Flow with Large Fluid Distortion. *Numerical Methods for Fluid Dynamics*, Morton KW, Baines MJ (eds). Academic Press: New York, (1982); pp. 273–285.