On the modeling of boundary conditions for embedded schemes

E.L. Mestreau[1], J.D. Baum[1], R. Löhner[2], C. Charman[3]
& D. Pelessone[4]
[1] Science Applications International Corporation, McLean, VA USA
[2] George Mason University, Fairfax, VA USA
[3] General Atomics, San Diego, CA USA
[4] ES3, San Diego, CA USA

Abstract

A simple embedded method for an unstructured grid is presented. An enhancement of the boundary treatments using ghost points is then described. Several examples are used to demonstrate the viability of the approach for inviscid flow simulations.

1 Introduction

With recent new threats to civilian infrastructures, the fast and accurate assessment of potential blast damage to targeted structures is requiring the use of more and more complex coupled CFD/CSD simulations. Two types of grids are commonly used: body conforming grids and embedded grids. For body-conforming grids, the external mesh faces match up with the surfaces (CSD faces, and outer boundaries) of the domain. The embedded method, which is also known as fictitious, immersed boundary or Cartesian method, takes a different approach where the surface of the structural model is placed inside the mesh of the fluid model. A special treatment of the fluid nodes close to the CSD surfaces is then handling the interaction between the fluid and the structure.

Considering the general case of moving and deforming surfaces with topology change, both approaches have strengths and weaknesses:

a) **Body conforming method:** the PDEs describing the flow need to be cast in an arbitrary Lagrangean-Eulerian frame of reference. The mesh moves in such a way as to follow the "wetted" CSD faces and to minimize the mesh distortion within the fluid model. If high distortion cannot be avoided or the structure topology has changed, the mesh is regenerated and the solution reinterpolated. All of these steps have been optimized in the course of the last decade and this approach has been extensively used [2][7]. Unfortunately, it also presents some serious shortcomings. The topology reconstruction of the water-tied fluid domain can fail due to singular surfaces. There is no easy way to get rid of those features by agglomerating surfaces in an intelligent way. The reconstruction also leads to very small elements when tiny gaps have to be filled reducing the timestep. Secondly, the performance of the parallel remeshing routines used to regenerate the mesh still poses problems beyond 8 processors. The more the structure deforms, the more likely remeshing has to be used hampering the advance of the solution. There is also an extra cost associated with the recalculation of the geometry parameters and a slight degradation of the solution induced by the interpolation.

b)**The embedded meshes:** The fluid mesh is not body conforming. Because none of the fluid boundaries move, the mesh does not change. Hence, the PDEs describing the flow can be left in the Eulerian reference frame. There is no expensive remeshing process and recalculation of the geometry parameters. At every timestep, the edges crossed by the CSD faces have to be identified and proper boundary conditions have to be applied in their vicinity. While this technique, more recently developed in our case, presents the advantage of bypassing the topology change reconstruction, it also presents some drawbacks: the boundary, which has the most profound influence on the ensuing physics, is also the place the worst elements are found; at the same time, near the boundary, the embedded boundary conditions have to be applied, reducing locally the order of approximation for the PDE. Mesh adaptation can be introduced to alleviate this approximation degradation, which diminishes with the mesh size, but at an extra cost of recalculating geometry parameters. This method has been extensively used for complex cases where structural folding and frequent change of topology are common [1][3][5].

The development of the present embedded, adaptive mesh capability was prompted by the inability of Computational Structural Dynamics (CSD) codes to ensure strict no-penetration during contact. Several blast-ship interaction simulations revealed that the amount of twisted and folded metal was such that it was virtually impossible to maintain a closed water-tied topology for the fluid. The embedded methodology offers an elegant solution to this problem.

The remainder of this paper provides more details related to the boundary conditions and the use of ghost points illustrated by a few examples.

Section 2 described the techniques used to mask the edges crossed by the CSD surface faces as well as the points close to it. The attention is then turned to the changes required in the flow code to treat embedded surfaces Sections 3-7.

Examples are given in Section 8 and finally Conclusion and Outlook are presented in Section 9.

2 Determination of crossed edges

Given the CSD triangulation and the CFD mesh, the first step is to find the CFD edges cut by the CSD faces. This is performed by building first an octree of the CSD faces. A parallel loop then processes each edge against the octree by using bounding boxes tests, building a list of possible intersecting CSD faces. This is followed by an in-depth test to determine which faces cross a given edge. The face closest to each of edge end-nodes is stored. In this way, cases thin gaps or cusps can be resolved.

Once the faces crossing edges are found, the closest face to the end-points of crossed edges is also stored. Boundary conditions for the points close to the embedded surface can be applied.

The building of the octree and its subsequent search is CPU expensive. For transient problems, the procedure described above can be improved considerably by using spatial coherence. Based on the assumption that the CSD faces triangulation does not move more than 1-2 elements from its current position in one timestep, it is therefore possible to reduce the test against the octree to a fairly limited number of edges surrounding the previously cut edges. If the topology of the CSD triangulation has changed, the crossed-edge information from the previous timestep is no longer valid. However the points close to the cut-edges in the previous timestep can be used to mark one or two layers of edges. As the change of topology only creates or deletes "wetted" faces in the proximity of the previous sets, only the marked edges have to be re-checked for crossing.

3 Treatment of crossed edges

The embedded surface method used is as simple as possible. It consists in nullifying the cut edges contribution, while correcting the mass-matrix to take into account the removed edges and finally by adding the newly created boundary conditions at the nodes used by the cut-edges. This is done as follows: If we consider an arbitrary field point I, the time-advancement of the unknowns for an explicit time integration scheme is given by:

$$M^{i}\Delta u^{i} = \Delta t \sum_{\Omega} C^{ij}\left(F_{i} + F_{j}\right) \qquad (1)$$

For any edges ij crossed by the CSD face, the coefficients C^{ij} are nullified. This implies that for a uniform state u=const, the balance of the fluxes for interior points with cut edges will not vanish. This is taken care of by the addition of a new boundary contribution, resulting in:

$$M^i \Delta u^i = \Delta t \left[\sum_{\Omega} C^{ij} \left(F_i + F_j \right) + C^i_\Gamma F_i \right] \tag{2}$$

The point-coefficients C^{ij} are obtained from the condition that Δu=0 for u=const. The mass-matrix M^i of points surrounded by cut edges must be modified to reflect the locally reduced volume due to the cut elements. This is done by finding the smallest cut edge fraction E of all cut-edges connected to each point. The mass-matrix modification is then given by

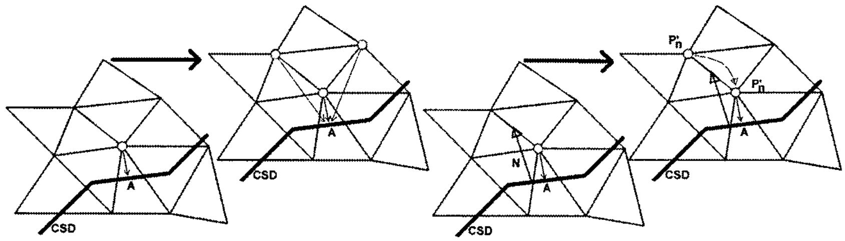
$$M^i_* = \frac{1 + \xi_{\min}}{2} M^i \tag{3}$$

Note that the value of the modified mass-matrix can never fall below half of its original value, implying that timestep sizes will always be acceptable.

4 Treatment of boundary conditions: low-order

Finding proper PDE boundary conditions is required for the nodes linked to the cut-edges. In the case of flow solvers, these are either an imposed velocity or an imposed normal velocity. For limiting and higher order schemes, one may also have to compose boundary conditions on the gradients. The required surface normal and boundary velocity are obtained directly from the closest CSD face to each of the new boundary points.

These lower-order boundary conditions may be improved by extrapolating the velocity from the surface with field information. The location where the flow velocity is equal to the surface velocity is the surface itself, and not the closest boundary point. As shown in Figure 1, for each boundary point, the closest point on the CSD face is found. Then, two (three) neighboring field (i.e., non-boundary) points are found and a triangular (tetrahedral) element that contains the boundary point is formed. The velocity imposed at the field point is then found by interpolation. In this way, the boundary velocity lags the field velocities by one timestep.
The normal gradients at the boundary points can be imposed by considering the "most aligned" field (i.e. inner) point to the line formed by the boundary point and the closest point on the CSD (Figure 2).



Figures 1 and 2: Boundary Condition Treatment

5 Treatment of boundary conditions: higher-order

Another way to approach the boundary condition problem is to use ghost points or mirrored points to compute the contribution of the crossed edges to the overall solution. This approach presents the advantage of not requiring the modification of the mass matrix as all edges (even the crossed ones) are taken into consideration. It also does not require an extensive modification of the various solvers. Figure 3 depicts the contribution due to the edges surrounding Point *Ptw*. A CSD boundary crosses the CFD domain. In this particular situation, P_i, located on the opposite side of the CSD face, will have to use the flow value of its mirror image pm1 based on the crossed CSD face.

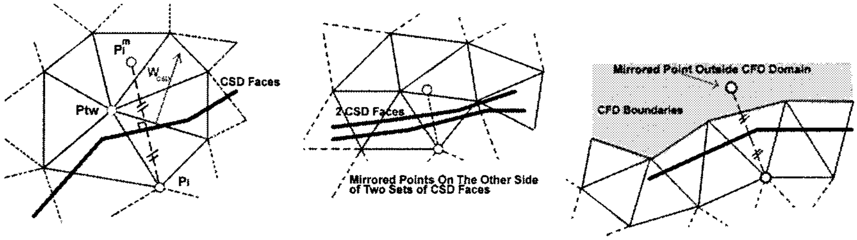

Pi^m
W_csd
CSD Faces
Ptw
Pi
2 CSD Faces
Mirrored Points On The Other Side of Two Sets of CSD Faces
Mirrored Point Outside CFD Domain
CFD Boundaries

Figure 3: Higher Order Wall Boundary Conditions

The flow values of the mirrored points are then interpolated from the element the point resides in using the following formulation:

$$\rho_m = \rho_i ; \vec{V}_m = \vec{V}_i - 2\left[\left(\vec{V}_i - \vec{W}_{CSD}\right)\vec{N}\right]\vec{N} ; P_m = P_i \tag{4}$$

Where W_{csd} is the average velocity of the crossed CSD face, ρ the density, V the flow velocity and P the pressure. Proper handling of the interpolation is also required as the element used for the interpolation might either not exist or be crossed (Figure 3). A more accurate formulation of the mirrored pressure and density can also be used taking into account the local radius of curvature of the CSD wetted surface.

$$P = P_i - \rho_i \frac{\left[\vec{v}_i - \left(\vec{v}_i - \vec{w}_{csd}\right)\vec{n}\right]^2}{R_i}\Delta ; \rho_m = \rho_i \left(\frac{P_m}{P_i}\right)^{1/\gamma}, V.... \tag{5}$$

where R_i is the radius of curvature and Δ the distance between the point and its mirror image. This second formulation is more complex and requires the computation of the 2 radii (3D) of curvature at each CSD point. The radius of curvature plays an important role for large elements but this influence can be diminished by the use of automatic h-refinement of the mesh.

6 Extrapolation of the solution

For problems with moving boundaries, mesh points can switch from one side of the surface to another. For these cases, the solution must be extrapolated from the proper state. The conditions that have to be met for extrapolation are as follows:

a) The edge was crossed at the previous timestep and is no longer crossed;
b) The edge has one field point (the point donating unknowns) and one boundary point (the point receiving unknowns);
c) The CSD face associated with the boundary point is aligned with the edge.

7 Adaptive mesh refinement

Adaptive mesh refinement is very often used to reduce the CPU and memory requirements without compromising the accuracy of the numerical solution. For transient problems with moving discontinuities, adaptive mesh refinement has been shown to be an essential ingredient of production codes [5][10][11]. For embedded CSD triangulations, the mesh can be refined automatically close to the surfaces. This has been done in the present case by including two further refinement indicators. The first simply looks at the edges cut by the CSD faces and refines the mesh to a certain element size or refinement level. The second, more sophisticated indicator looks at the surface curvature and refines the mesh only in regions where the element size is deemed insufficient.

8 Examples

As a large bulk of our work is to perform blast/damage assessment, the examples presented are inviscid shock propagation cases: the shock tube and a sphere deflection of a shock with impact on a back wall.

8.1 Sod shock tube

The following Figures present the comparison of results if the Sod shock-tube (ρ1 = p1 = 1.0, ρ2 = p2 = 0.1) obtained with a classical body fitted and embedded method using either low or higher-order boundary conditions. The embedded geometry can be discerned from Figure 4a. Figures 4b,c,d respectively presents snapshots of the density contours for the classical method and the embedded method using the low-order and higher-order boundary conditions. Although the embedded technique is rather primitive, the results are surprisingly good when using the higher-order boundary conditions. The main difference is a little more noise in the contact discontinuity region, which may be expected, as this is a linear discontinuity. For cases with moving bodies, all face-edge intersections checks are carried out incrementally and points that "cross states" due to CSD face movement need to be interpolated appropriately. The ends of the shock tube have been closed and Figure 5 shows the pressure time history of an off-center point. Again, after several reflections, the high-order boundary conditions produce very good results in line with the body fitted method. The low-order boundary conditions results are shifting up as if mass was increasing in time.

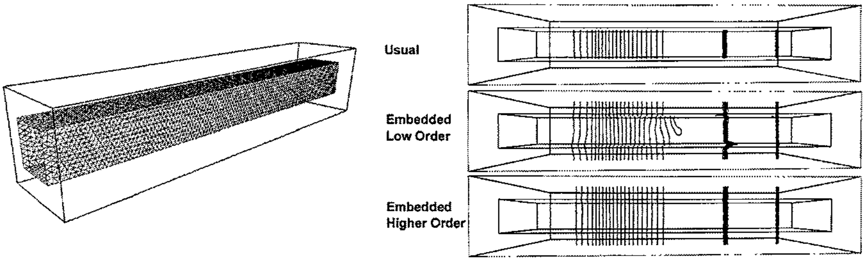


Figure 4a,b,c,d: Shock Tube, Density Contours

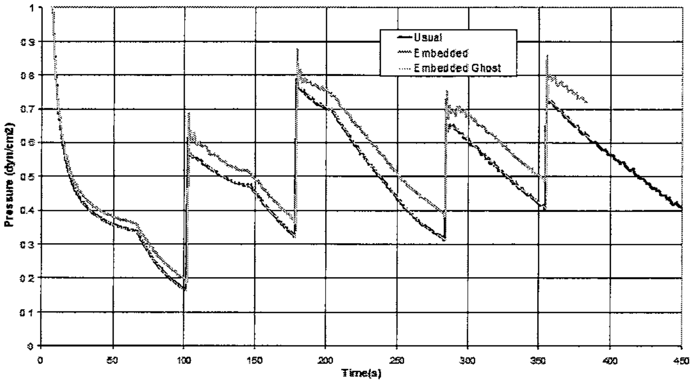


Figure 5: Time History Pressure at a Point in the Tube

8.2 EMI testcase

The EMI testcase is a micro scale benchmark used to study and design protective walls. This specific version, for which accurate measurement have been obtained, simulates a few grams charge positioned at about 1.5cm above the ground and facing a 5.5cm high wall. Figure 6 shows the setup of the simulation and a Schlieren graph of the real case. Along with various Schlieren graphs, time histories are recorded and Figure 6 presents the results obtained with the body-fitted method and the embedded method with low order and high order boundary conditions.

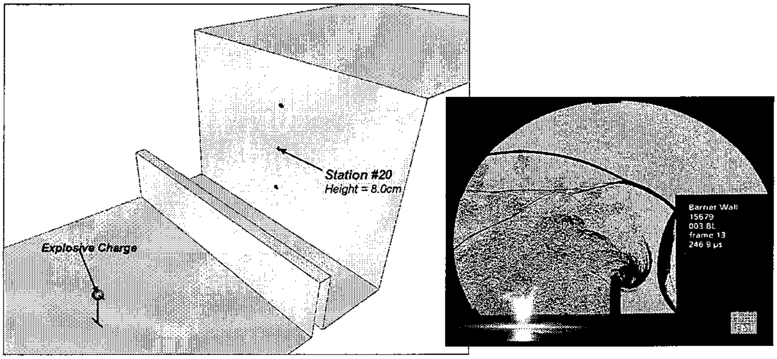


Figure 6: Testcase Setup

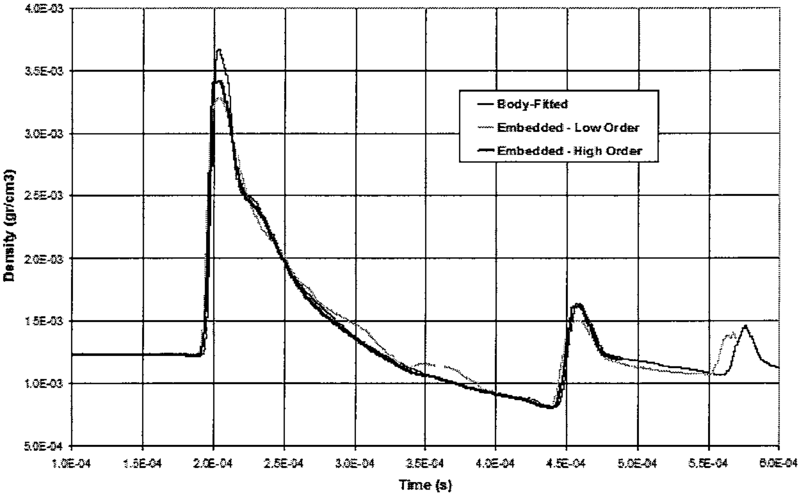


Figure 7: EMI Testcase, Density Time History

Transactions on the Built Environment vol 71, © 2004 WIT Press, www.witpress.com, ISSN 1743-3509

The results of the embedded method with high order boundary conditions are matching the body-fitted method, despite the fact that the peak pressure is slightly underevaluated. This can be explained by the relative coarseness of the mesh used in the vicinity of the wall and the building. The lower order boundary condition, similarly to the shock tube, does not provide results accurate enough. Bumps and slight sift in time are also visible.

8.3 Blast interaction with a generic ship hull

Figure 8 presents the interaction of an explosion with a generic ship hull. For this fully coupled CFD/CSD simulation, the structure is modeled with quadrilateral shell elements, the fluid as mixture of high explosive and air. The embedded technique is used. The structure elements are assumed to fail once the average strain exceeds 60%. As the shell elements fail, the blast diffracts through the holes. Figures 8a-d show the structure as well as the pressure contours in a cut plane at two times during the run. The influence of the bulkheads on surface velocity is clearly visible.

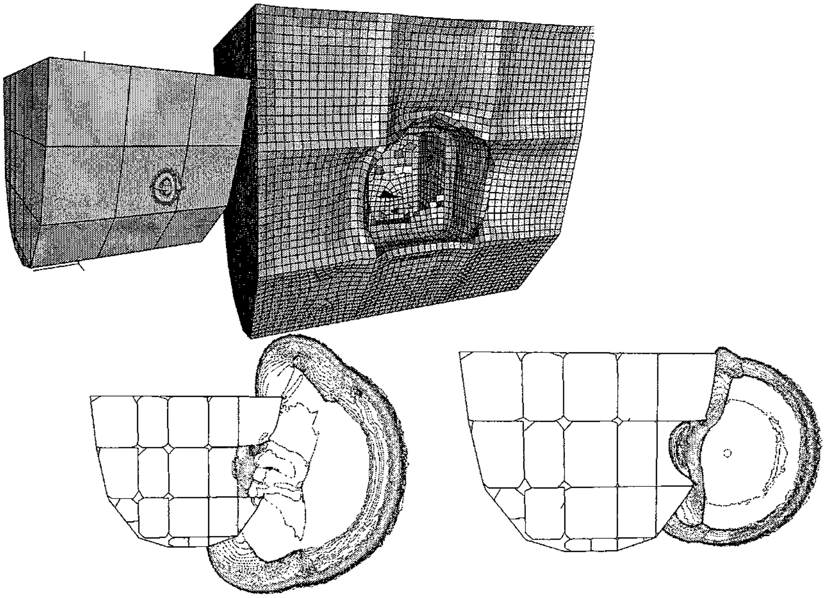


Figure 8: Surface of Generic Ship Hull and Pressure in Cut-Plane

9 Conclusion and outlook

A CFD solver based unstructured, body conforming grids has been extended to treat embedded or immersed CSD surfaces given by triangulations. The key modification of the original, edge-based solver is to handle all geometry-parameters belonging to edges cut by CFD faces. In order to guarantee a minimum level of accuracy, additional boundary points or ghost points are introduced. The higher-order ghost points provide a more accurate solution as long as the mesh is not too coarse. Adaptive mesh refinement based on proximity or the curvature of the embedded CSD surfaces is used to enhance the accuracy of the solution. Several examples have been included that show the viability of the approach for inviscid flow problems.

Future work will center on improved boundary conditions, as well as faster crossing checks for transient problems.

Acknowledgements

This work was partially supported by DTRA and AFOSR. The technical monitors were Drs. Michael Giltrud, Young Sohn and Leonidas Sakell.

References

[1] M.J. Aftosmis, M.J. Berger and G. Adomavicius – A Parallel Multilevel Method for Adaptively Refined Cartesian Grids with Embedded Boundaries; *AIAA-00-0808 (2000).*

[2] J.D. Baum, H. Luo and R. Löhner – Validation of New ALE, Adaptive Unstructured Moving Body Methodology for Multi-Store Ejection Simulations; AIAA-95-1792 (1995).

[3] J.D. Baum, H. Luo, R. Löhner, C. Yang, D. Pelessone and C. Charman – A Coupled Fluid/Structure Modeling of Shock Interaction with a Truck; *AIAA-96-0795 (1996).*

[4] J.D. Baum, H. Luo, R. Löhner, E. Goldberg and A. Feldhun – Application of Unstructured Adaptive Moving Body Methodology to the Simulation of Fuel Tank Separation from an F-16 C/D Fighter; *AIAA-97-0166 (1997).*

[5] J.D. Baum, H. Luo, E. Mestreau, R. Löhner, D. Pelessone and C. Charman – A Coupled CFD/CSD Methodology for Modeling Weapon Detonation and Fragmentation; *AIAA-99-0794 (1999).*

[6] P.L. George and H. Borouchaki – Delaunay Triangulation and Meshing; *Editions Hermes, Paris (1998).*

[7] S.L. Karman – SPLITFLOW: A 3-D Unstructured Cartesian / Prismatic Grid CFD Code for Complex Geometries; *AIAA-95-0343 (1995).*

[8] A.M. Landsberg and J.P. Boris – The Virtual Cell Embedding Method: A Simple Approach for Gridding Complex Geometries; *AIAA-97-1982 (1997).*

[9] R. Löhner, C. Yang, J.D. Baum, H. Luo, D. Pelessone and C. Charman – The Numerical Simulation of Strongly Unsteady Flows With Hundreds of Moving Bodies; *Int J. Num. Meth. Fluids 31, 113-120 (1999).*

[10] R. Löhner and C. Charman – Fluid/Structure Interaction Algorithms for Rupture and Topology Change; *Proc. 1999 JSME Computational Mechanics Division Meeting, Matsuyama, Japan, November 1999.*

[11] J.E. Melton, M.J. Berger and M.J. Aftosmis – 3-D Applications of Cartesian Grid Euler Method; *AIAA-93-0853-CP (1993).*

[12] R.B. Pember, J.B. Bell, P. Colella, W.Y. Crutchfield and M.L. Welcome – An Adaptive Cartesian Grid Method for Unsteady Compressible Flow in Irregular Regions – *J. Comp. Phys. 120,278 (1995).*

[13] J.J Quirk – A Cartesian Grid Approach with Hierarchical Refinement for Compressible Flows; *NASA CR-194938, ICASE Report No. 94-51, (1994).*

[14] D. Sharov, H. Luo, J.D. Baum and R. Löhner – Time-Accurate Implicit ALE Algorithm for Shared-Memory Parallel Computers; *First International Conference on Computational Fluid Dynamics, Kyoto, Japan, July 10-14 (2000).*

[15] D. de Zeeuw and K. Powell – An Adaptive-Refined Cartesian Mesh Solver for the Euler Equations; *AIAA-91-1542 (1991).*