

COMPARISON OF BODY-FITTED, EMBEDDED AND IMMERSED 3-D EULER PREDICTIONS FOR BLAST LOADS ON COLUMNS

Rainald Löhner, Joseph D. Baum, Eric L. Mestreau and Darren Rice

*CFD Center, Dept. of Computational and Data Science
M.S. 6A2, College of Science, George Mason University
Fairfax, VA 22030-4444, USA*

Advanced Technology Group, SAIC

US Government, Washington, D.C.

A reinforced concrete, as well as a steel column, were subjected to typical blast loads. The ensuing 3-D compressible flowfields were computed with the same 3-D edge-based FEM-FCT solver using the body-fitted, embedded-mesh and immersed body approach. The resulting pressures and impulses were recorded at stations close to the columns and compared for the different approaches. The results indicate that:

- a) The body-fitted and embedded-mesh methods converge to the same results as the grid is refined;
- b) The 2nd order embedded-mesh method is comparable in accuracy to the body-fitted results for grids of similar size;
- c) In order to obtain results of comparable quality, the 1st order embedded-mesh method requires grids with element sizes on the boundary that are approximately half of those used for the body fitted and 2nd order embedded-mesh approaches;
- d) The immersed body approach exhibits much more numerical dissipation than the body-fitted and embedded-mesh methods.

Therefore, if the manual labour required to set up a body-fitted domain is excessive, or computing resources are plentiful, the embedded-mesh approach is a very attractive option.

I. INTRODUCTION

With the advent of robust, accurate flow solvers and automatic grid generators, the task of defining quickly a flow domain and the required boundary conditions has become a key bottleneck for numerical simulations. Presently, the widespread use of blast predictions based on 3-D Euler solvers is hindered by this key issue. For so-called body fitted grids, the surface definition must be water-tight, and any kind of geometrical singularity, as well as small angles, should be avoided in order to generate a mesh of high quality. This typically presents no problems if the definition of the structure emanates from a CAD package. However, in many cases first the CSD models (i.e. abstractions of the real geometry) are built, and the ‘wetted surface’ is exported, serving, in part, as the definition of the flow domain. This data may not be water-tight, may have small strips and geometrical singularities. Therefore, even with sophisticated software toolkits, manual cleanup in most cases takes several days for a complete building. An alternative is to use grids that are not body-conforming, and simply ‘embed’ the triangulations of the wetted surfaces of the structures in them. Techniques of this kind are also known as immersed, embedded, fictitious domain or Cartesian methods. The treatment of points in the vicinity of the embedded CSD triangulations or CSD bodies is modified in such a way that the required kinetic or kinematic boundary conditions are properly imposed^{1, 8–11, 13, 15, 16, 25, 26, 28–33}.

Copyright © 2007 by the Authors. Published by the American Institute of Aeronautics and Astronautics, Inc. with permission.

An embedded technique of this type, combined with adaptive unstructured grids, has been pursued by the authors for a number of years for shock-structure interaction studies, where extremely ‘dirty’ CSD surfaces may be present²⁴. The essential elements of this technique may be summarized as follows:

- The key modification of the original, body fitted edge-based solver was the removal of all geometry-parameters (essentially the area normals) belonging to edges cut by embedded surface faces;
- Higher-order boundary conditions are achieved by duplicating crossed edges and their endpoints, or by extrapolating values and gradients from inside the domains to the surface;
- Geometric resolution and solution accuracy may be enhanced by adaptive mesh refinement that is based on the proximity to or the curvature of the embedded CSD surfaces; and
- In order to save work, user-defined or automatic deactivation for regions inside immersed solid bodies is employed.

Naturally, the question arises as to how accurate these embedded techniques are, and whether they could be used for those cases where manual labour is expected to be prohibitively high.

To this end, a typical reinforced concrete column, as well as a typical steel column, were subjected to a blast of 100 kg TNT equivalent. Both cases were analyzed with the same solver and code, exercising the body-fitted, embedded-mesh and immersed-body options. Pressure and impulse obtained at several stations was compared.

The remainder of the paper is organized as follows: Section 2 describes the overall methodology and algorithms used for blast-structure interaction calculations. Sections 3,4 detail the embedded-mesh and immersed-body techniques used. Section 5 shows the comparison of several relevant runs, and discusses the implications. Some conclusions and an outlook are given in Section 6.

II. METHODOLOGY

Any flow simulation proceeds through the following stages:

- Pre-Processing
- Grid Generation
- Flow Solver
- Post-Processing

In the pre-processing phase, the data and boundary conditions are acquired and defined, the desired mesh size is specified and all run-time files are prepared. For the applications shown here, these tasks were carried out with FECAD²². For body-fitted grids the exact geometry is imported and/or built. For embedded grids or immersed bodies, the CSD data is acquired, and the mesh in the region of the structure is automatically specified to be small enough. An advantage of the present methodology is that the embedded, immersed and body-fitted approaches may be combined in a single run. Thus, steel frames/columns may be treated embedded, while reinforced concrete columns, which typically have cleaner, watertight connections, can be treated as body-fitted. The computational domain is then filled with tetrahedral elements of specified size by using FEGEN, which incorporates the advancing front technique^{18,21}. For the flow solver, the FEM-FCT scheme¹⁷ as implemented in FEFLO^{22,23} is used. The flowfields are initialized from either user-specified tables or the output of 1-D, 2-D/axisymmetric or fine 3-D results. Post-processing is carried out using FEPOST^{6,19} as well as several x/y plotting tools such as gnuplot. All the pre- and post-processing, as well as any run requiring less than 4 Mtets elements are run on the PC, allowing for full integration under a single graphical user interface (GUI). The larger runs were carried out on a 64 processor SGI Altix system.

III. EMBEDDED MESH TECHNIQUES

In what follows, we denote by CSD faces the surface of the computational domain that is embedded. We implicitly assume that this information is given by a triangulation, which typically is obtained from a CAD package via STL files, but may also originate from remote sensing data, medical images or from a CSD

code (hence the name) in coupled fluid-structure applications. For immersed methods we assume that the embedded object is given by a tetrahedral mesh.

Embedded grids are treated by imposing appropriate **kinematic boundary conditions** for the fluid nodes close to the embedded surfaces. Depending on the required order of accuracy and simplicity, a first or second-order (higher-order) scheme may be chosen to apply the kinematic boundary conditions. Figure 1 illustrates the basic difference between these approaches. Note that in both cases the treatment of infinitely thin surfaces with fluid on both sides (e.g. fluid-structure interaction simulations with shell elements) is straightforward.

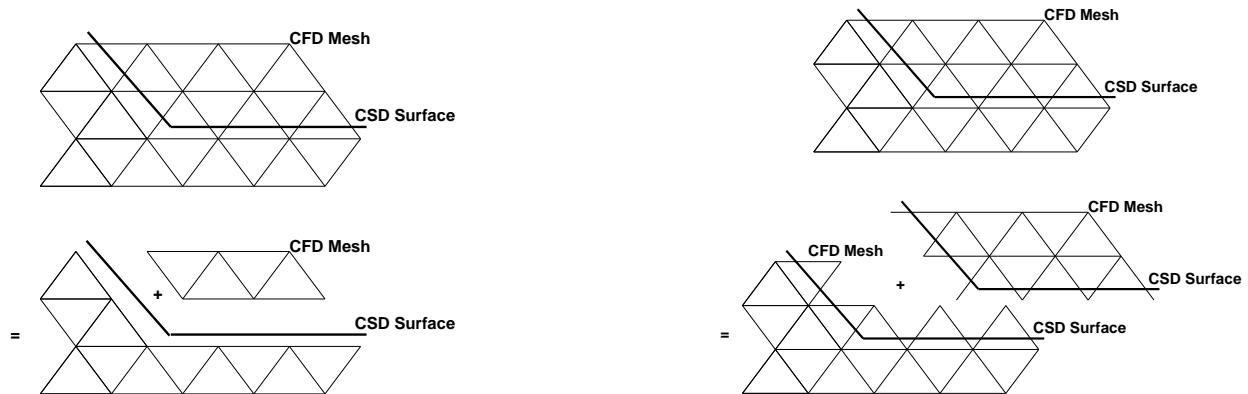


Figure 1: 1st/2nd Order Treatment of Embedded Surfaces

A first-order scheme may be achieved by:

- Eliminating the edges crossing the embedded surface;
- Forming boundary coefficients to achieve flux balance;
- Applying boundary conditions for the end-points of the crossed edges based on the normals of the embedded surface.

A second-order scheme may be obtained by:

- Duplicating the edges crossing the embedded surface;
- Duplicating the end-points of crossed edges;
- Applying boundary conditions for the end-points of the crossed edges based on the normals of the embedded surface.

Note that in either case CFD edges crossed by CSD faces are modified/duplicated.

3.1 Determination of Crossed Edges

Given the CSD triangulation and the CFD mesh, the first step is to find the CFD edges cut by CSD faces. This is performed by building a fast spatial search data structure, such as an octree or a bin for the CSD faces. Without loss of generality, let us assume an octree for the CSD faces. Then, a (parallel) loop is performed over the edges. For each edge, the bounding box of the edge is built. From the octree, all the faces in the region of the bounding box are found. This is followed by an in-depth test to determine which faces cross the given edge. The crossing face closest to each of the edge end-nodes is stored. This allows to resolve cases of thin gaps or cusps. Once the faces crossing edges are found, the closest face to the end-points of crossed edges is also stored. This information is required to apply boundary conditions for the points close to the embedded surface. For cases where the embedded surfaces only cut a small portion of the CFD edges, a considerable speedup can be realized by removing from the list of edges tested all those that fall outside the global bounding box of the CSD faces. The resulting list of edges to be tested in depth may be reduced further by removing all edges whose bounding boxes do not fall into an octree or bin covering that spatial region. One typically finds that the list of edges to be tested in detail has been reduced by an order of magnitude.

3.2 First Order Treatment

The first order scheme is the simplest to implement. Given the CSD triangulation and the CFD mesh, the CFD edges cut by CSD faces are found and deactivated. Considering an arbitrary field point i , the time-advancement of the unknowns \mathbf{u}^i for an explicit edge-based time integration scheme is given by:

$$M^i \Delta \mathbf{u}^i = \Delta t \sum_{ij\Omega} C^{ij} (F_i + F_j) \quad . \quad (1)$$

Here C, F, M denote, respectively, the edge-coefficients, fluxes and mass-matrix. For any edge ij crossed by a CSD face, the coefficients C^{ij} are set to zero. This implies that for a uniform state $\mathbf{u} = \text{const.}$ the balance of fluxes for interior points with cut edges will not vanish. This is remedied by defining a new boundary point to impose total/normal velocities, as well as adding a ‘boundary contribution’, resulting in:

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

The point-coefficients C_{Γ}^i are obtained from the condition that $\Delta \mathbf{u} = 0$ for $\mathbf{u} = \text{const.}$ Given that gradients (e.g. for limiting) are also constructed using a loop of the form given by Eqn.(18.4) as:

$$M^i \mathbf{g}^i = \sum_{ij\Omega} C^{ij} (u_i + u_j) \quad , \quad (3)$$

it would be desirable to build the C_{Γ}^i coefficients in such a way that the constant gradient of a linear function u can be obtained exactly. However, this is not possible, as the number of coefficients is too small. Therefore, the gradients at the boundary are either set to zero or extrapolated from the interior of the domain. The mass-matrix M^i of points surrounded by cut edges must be modified to reflect the reduced volume due to cut elements. The simplest possible modification of M^i is given by the so-called ‘cut edge fraction’ method.

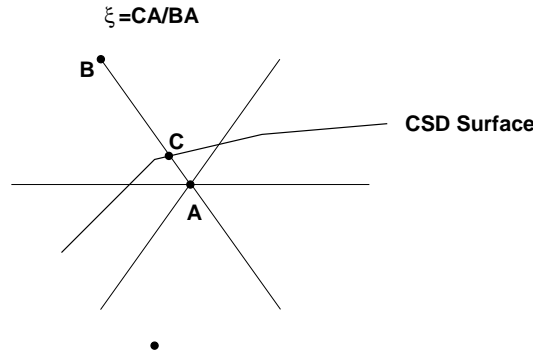


Figure 2: Cut Edge Fraction

In a pass over the edges, the smallest ‘cut edge fraction’ ξ for all the edges surrounding a point is found (see Figure 2). The modified mass-matrix is then given by:

$$M_*^i = \frac{1 + \xi_{min}}{2} M^i \quad . \quad (4)$$

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

3.2.1 Boundary Conditions

For the new boundary points belonging to cut edges the proper PDE boundary conditions are required. 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 impose 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 low-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 3, for each boundary point the closest point on the CSD face is found. Then, two (three) neighbouring 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.

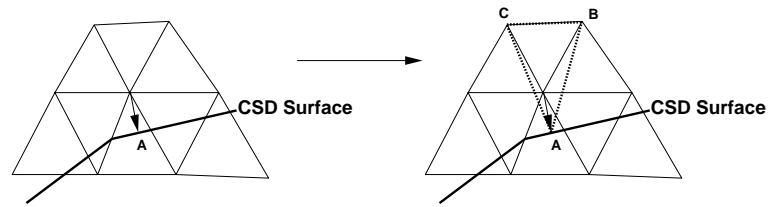


Figure 3: Extrapolation of Velocity

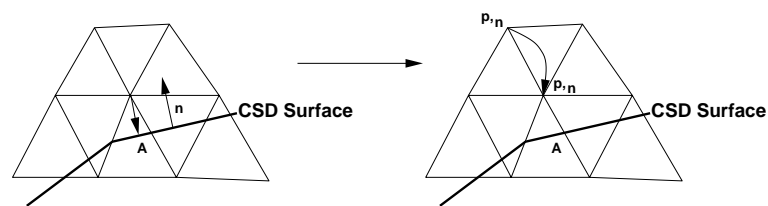


Figure 4: Extrapolation of Normal Pressure Gradient

The normal gradients at the boundary points can be improved by considering the ‘most aligned’ field (i.e., non-boundary) point to the line formed by the boundary point and the closest point on the CSD face (see Figure 4).

3.3 Higher Order Treatment

As stated before, a higher-order treatment of embedded surfaces may be achieved by using 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. On the other hand, it requires more memory due to duplication of crossed edges and points, as well as (scalar) CPU time for renumbering/reordering arrays. Particularly for moving body problems, this may represent a considerable CPU burden.

3.3.1 Boundary Conditions

By duplicating the edges, the points are treated in the same way as in the original (non-embedded) case. The boundary conditions are imposed indirectly by mirroring and interpolating the unknowns as required. Figure 5 depicts the contribution due to the edges surrounding point i . A CSD boundary crosses the CFD domain. In this particular situation point j , which lies on the opposite side of the CSD face, will have to use the flow values of its mirror image j' based on the crossed CSD face.

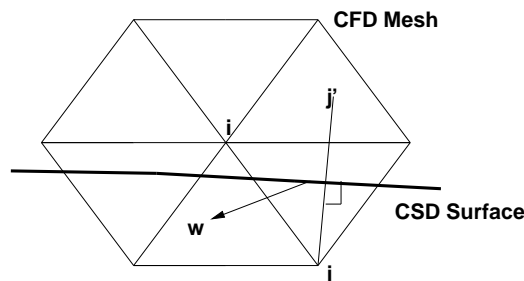


Figure 5: Higher Order Boundary Conditions

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

$$\rho_m = \rho_i \quad , \quad p_m = p_i \quad , \quad \mathbf{v}_m = \mathbf{v}_i - 2 [(\mathbf{v}_i - \mathbf{w}_{csd}) \cdot \mathbf{n}] \mathbf{n} \quad , \quad (5)$$

where \mathbf{w}_{csd} is the average velocity of the crossed CSD face, ρ the density, \mathbf{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 be crossed (Figure 6a) or not exist (Figure 6b).

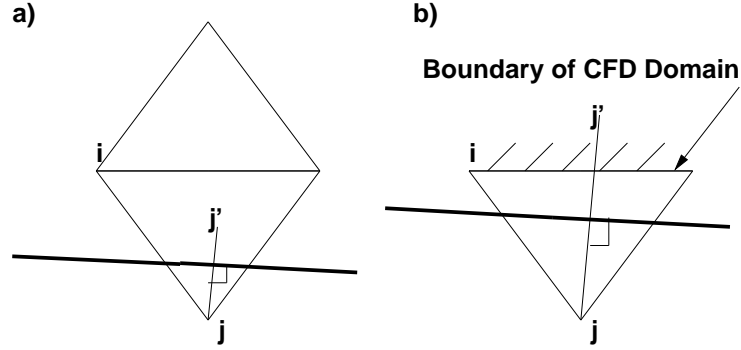


Figure 6: Problem Cases

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 surface⁹:

$$p_m = p_i - \rho_i \frac{[\mathbf{v}_i - (\mathbf{v}_i - \mathbf{w}_{csd}) \cdot \mathbf{n}]^2}{R_i} \Delta \quad , \quad \rho_m = \rho_i \left(\frac{p_m}{p_i} \right)^{\frac{1}{7}} \quad , \quad (6)$$

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.

IV. IMMERSSED BODY TECHNIQUE

The presence of immersed bodies is imposed in the flowfield by adding suitable force-functions. If we consider a rigid, closed body, as sketched in Figure 7, an obvious aim is to enforce, within the body, that the fluid velocity is the same as the body velocity, i.e. $\mathbf{v} = \mathbf{v}_b$. This may be accomplished by applying a force term of the form:

$$\mathbf{f} = -c_0(\mathbf{v}_b - \mathbf{v}) \quad (7)$$

for points that are inside of the body. This particular type of force function is known as the penalty force technique^{2,10,12,14,27}.

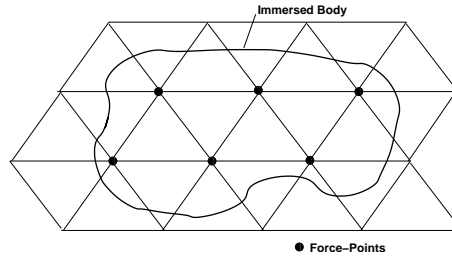


Figure 7: Kinetic Treatment of Embedded Surfaces

The penalty force technique used here follows the approach of Mohd-Yusof²⁷. The usual right-hand side for the flow equations at immersed points (or cells) is evaluated first. Then, a force is added such that the

velocity at the next timestep satisfies the kinematic boundary conditions. Writing the spatially discretized form of the momentum equations at each point (or cell) i as:

$$\mathbf{M} \frac{\Delta \mathbf{v}_i}{\Delta t} = \mathbf{r}_i + \mathbf{f}_i \quad , \quad (8)$$

\mathbf{f}^i is obtained as:

$$\mathbf{f}^i = \mathbf{M} \frac{\mathbf{w}_i^{n+1} - \mathbf{v}_i^n}{\Delta t} - \mathbf{r}^i \quad . \quad (9)$$

Here \mathbf{w}_i denotes the velocity of the immersed body at the location of point (or cell) i , and n the timestep. For explicit timestepping schemes, this force function in effect imposes the (required) velocity of the immersed body at the new timestep. Schemes of this kind have been used repeatedly in conjunction with fractional step / projection methods for incompressible flow^{2, 10–12, 14, 27, 30}, and, recently, also for compressible flows⁷. While simple to program and employ, the force-based enforcement is particularly useful if the ‘body thickness’ covers several CFD mesh elements. This is because the pressures obtained are continuous across the embedded surface/ immersed body. This implies that for thin embedded surfaces such as shells, where the pressure is different on both sides, this method will not yield satisfactory results.

4.1 Implementation Details

The search operations required for the imposition of kinetic boundary conditions are as follows:

- Given a set of CSD faces (triangulation): find the edges of the CFD mesh cut by CSD faces (and possibly 1-2 layers of neighbours);
- Given a set of CSD volumes (tetrahedrization): find the points of the CFD that fall into a CSD tetrahedron (and possibly 1-2 layers of neighbours).

The first of these is basically the same as required for embedded CSD methods, and has been dealt with extensively above. The second task may be solved in a number of ways:

Loop over the Immersed Body Elements

- Initialization:
- Store all CFD mesh points in a bin, octree, or any other similar data structure;
- Loop over the immersed body elements:
- Determine the bounding box of the element;
- Find all points in the bounding box;
- Detailed analysis to determine the shape function values.

Loop over the CFD Mesh Points

- Store all immersed body elements in a bin, modified octree, or any other similar data structure;
- Loop over the CFD mesh points:
- Obtain the elements in the vicinity of the point;
- Detailed analysis to determine the shape function values.

In both cases, if the immersed body only covers a small portion of the CFD domain, one can reduce the list of points stored or points checked via the bounding box of all immersed body points. Note also that both approaches are easily parallelized on shared memory machines.

V. RESULTS

5.1 Shock on Reinforced Concrete Column:

The first case considered is shown in Figure 8a. The top and bottom surfaces are considered as walls, while all other surfaces are treated as open boundaries. The center of the explosion is at a distance of 2.5 m from the column, and 3 ft above the ground. Surface grids for the 'fine mesh', which are indicative of the element size in the volume, are shown in Figure 8b,c. These grids used had approximately 2.4 Mtets. The same series of runs were also performed with a so-called 'superfine mesh' of 21 Mtets, and a 'coarse mesh' of 1.4 Mtets. The initial pressure distribution, obtained via interpolation from a detailed axisymmetric solution, as well as the pressure distribution upon impact on the column at 1 msec are shown in Figures 8d,e.

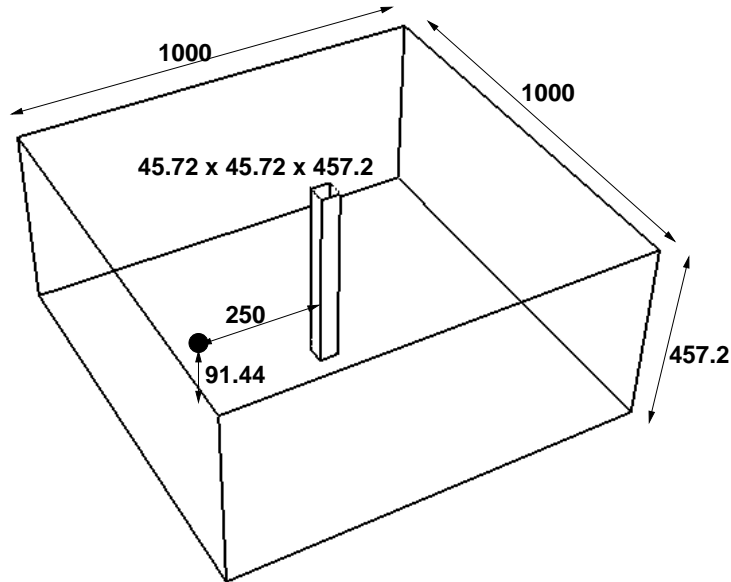
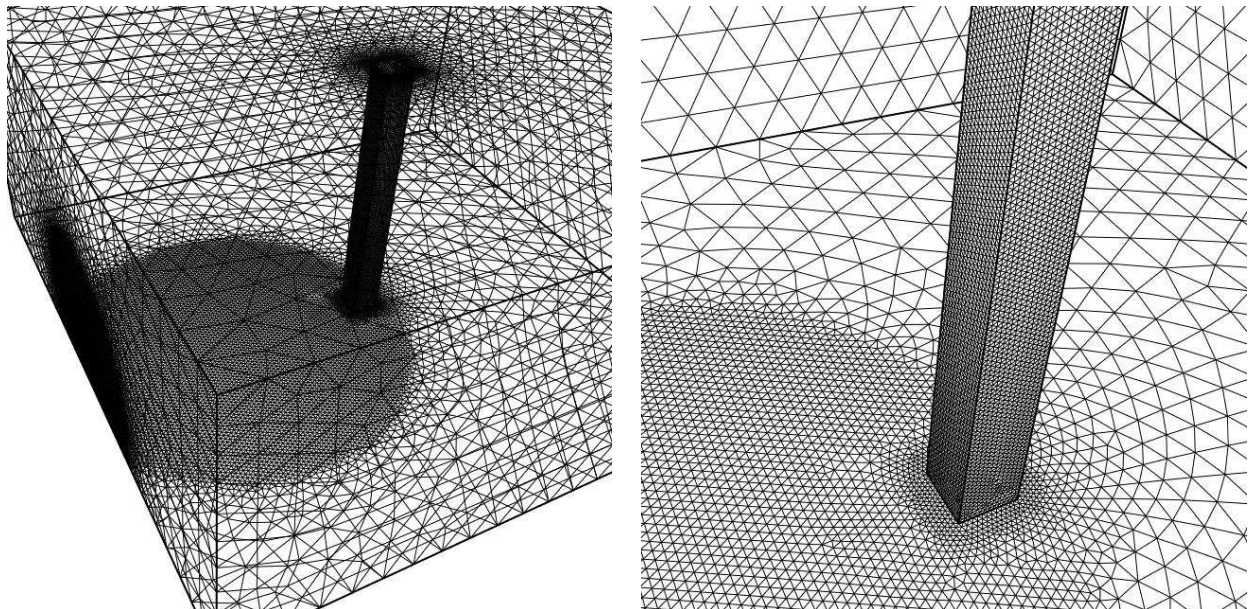
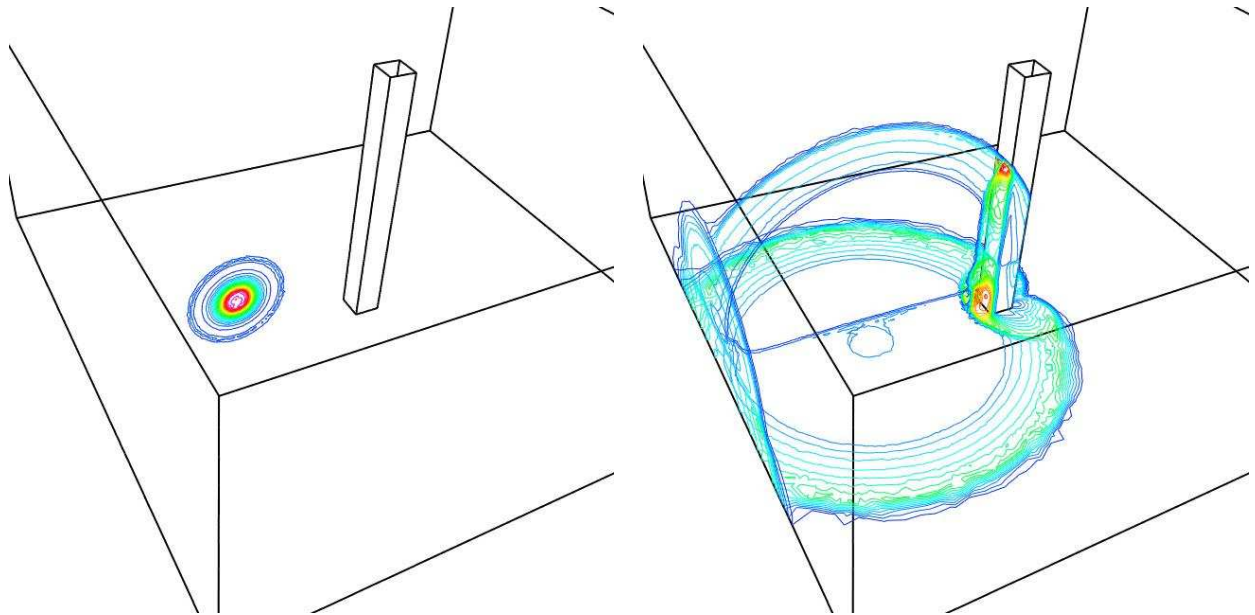


Figure 8a Shock on RC Column: Problem Definition



Figures 8b,c Shock on RC Column: Surface Grids



Figures 8d,e Shock on RC Column: Pressures

In order to gain insight into the differences between the results the pressures and impulses recorded at 6 different stations (3 in the front, going top to bottom, 3 in the back, going top to bottom) are shown in Figures 8g-q. The notation is as follows: BF: body-fitted, EMB: 1st order embedded, EMB2: 2nd order embedded, IMM: immersed body, F: fine mesh, SF: superfine mesh. For each of these 6 stations 4 comparisons are shown: fine and superfine mesh with body-fitted and 1st order embedded (BFFSF-EMBF2SF), fine mesh with body-fitted, 1st and 2nd order embedded (BFF-EMBF-EMBF2), coarse mesh with body-fitted, 1st and 2nd order embedded (BF-EMB-EMB2) and fine mesh with body-fitted, 1st order embedded and immersed body methods (BFF-EMBF-IMMF). One can see that the body-fitted and embedded-mesh results for the superfine mesh are very close, and are approaching a grid-independent solution. As expected, the superfine pressures and impulses are higher. Peak impulses are approximately 5-10% lower on the front wall, and 5-10% higher on the back wall. This simply reflects the fact that on the coarser mesh the achievable shock rise and expansion decay times are larger than on the finer mesh. The results obtained at the stations shown, as well as approximately 40 others that were placed on the column allow us to reach the following conclusions:

- a) The body-fitted and embedded-mesh methods of 1st and 2nd order converge to the same results as the grid is refined;
- b) The 2nd order embedded-mesh method is comparable in accuracy to the body-fitted results for grids of similar size;
- c) As expected, the 1st order embedded-mesh method exhibits more numerical dissipation than the 2nd order embedded-mesh method or the body-fitted approach; results of comparable accuracy require grids with element sizes on the boundary that are approximately half of those used for the other two approaches, unless a grid-independent result is already achieved;
- d) On grids of moderate size, while not as good as the body-fitted results, the 1st order embedded-mesh results provide impulses that are within 10% of the grid-converged results; given all other uncertainties in typical building response calculations (beam/girder connections, weldings, etc.), a 10% error is deemed acceptable for many cases;
- e) The immersed body approach exhibits much more numerical dissipation than the body-fitted and embedded-mesh methods.

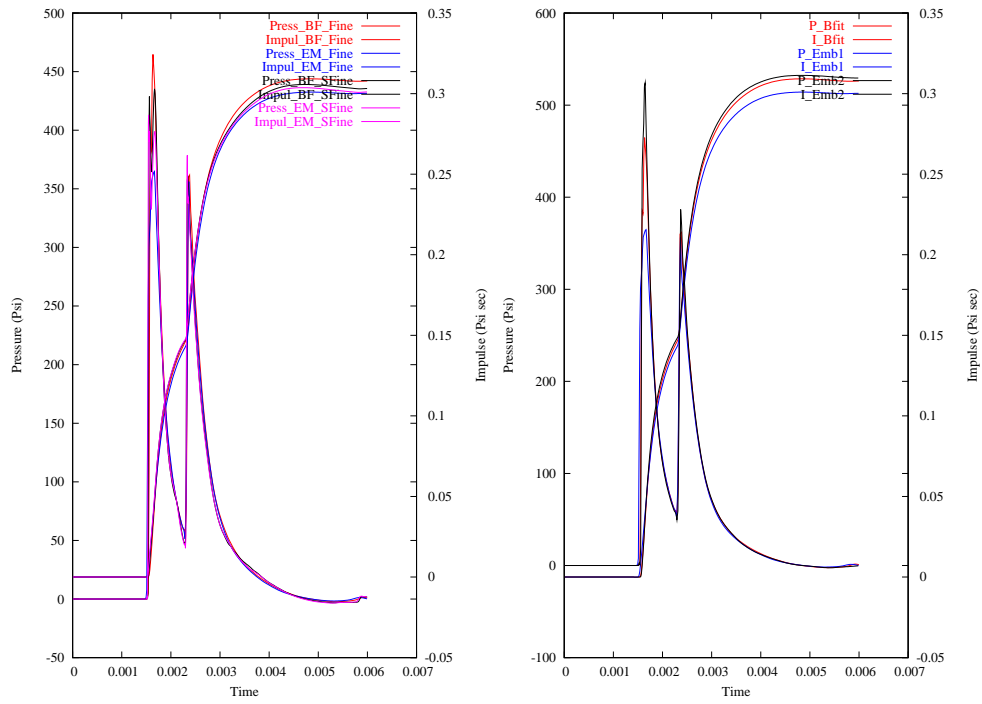


Figure 8f RC Column: BFFSF-EMBFSF and BFF-EMBF-EMBF2 (Front, Top)

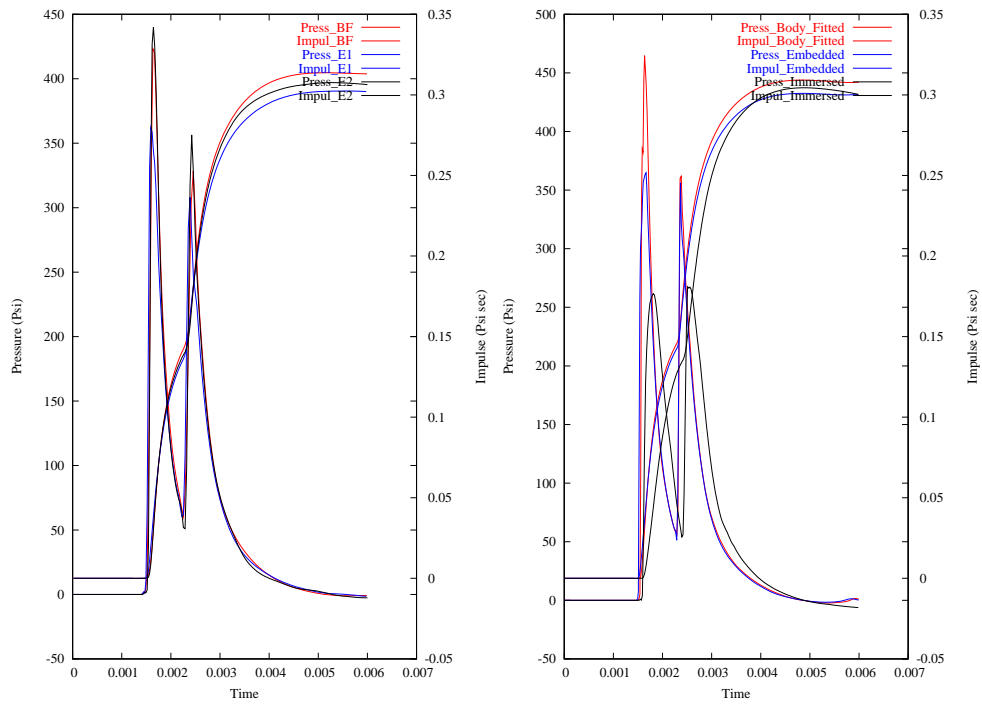


Figure 8g RC Column: BF-EMB-EMB2 and BFF-EMBF-IMMF (Front, Top)

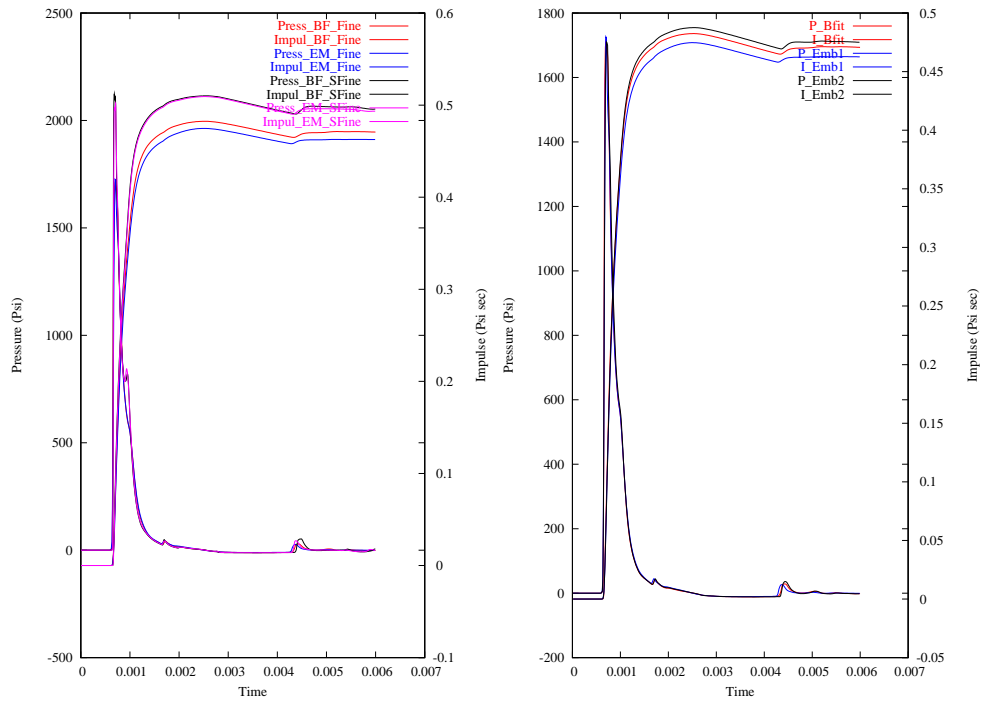


Figure 8h RC Column: BFFSF-EMBF^{SF} and BFF-EMBF-EMBF² (Front, Middle)

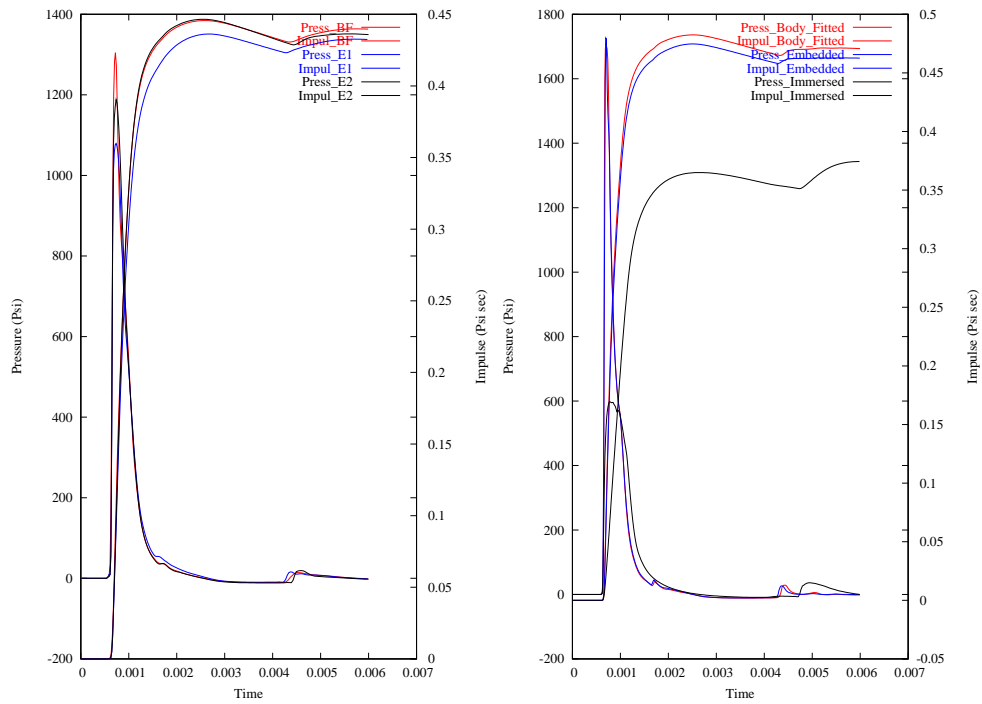


Figure 8i RC Column: BF-EMB-EMB² and BFF-EMBF-IMMF (Front, Middle)

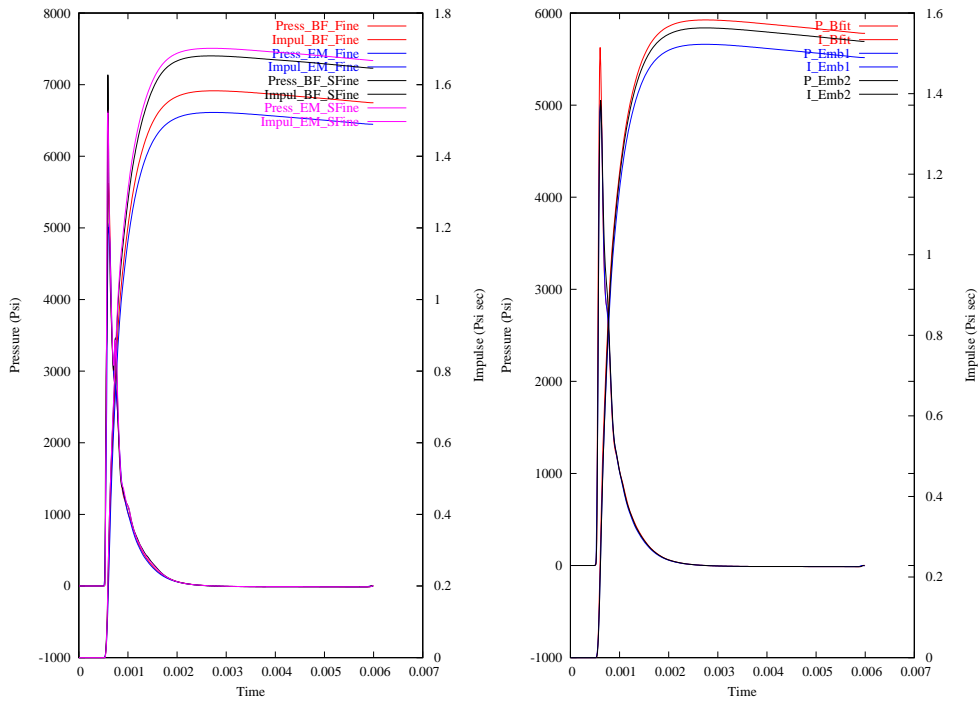


Figure 8j RC Column: BFFSF-EMBF2 and BFF-EMBF-EMBF2 (Front, Bottom)

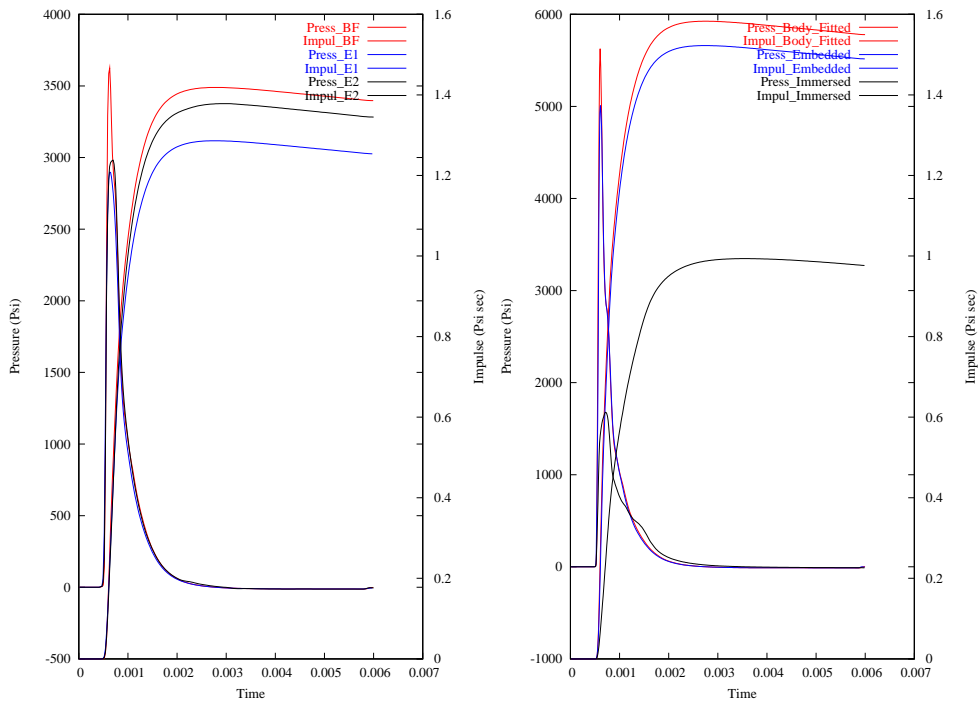


Figure 8k RC Column: BF-EMB-EMB2 and BFF-EMBF-IMMF (Front, Bottom)

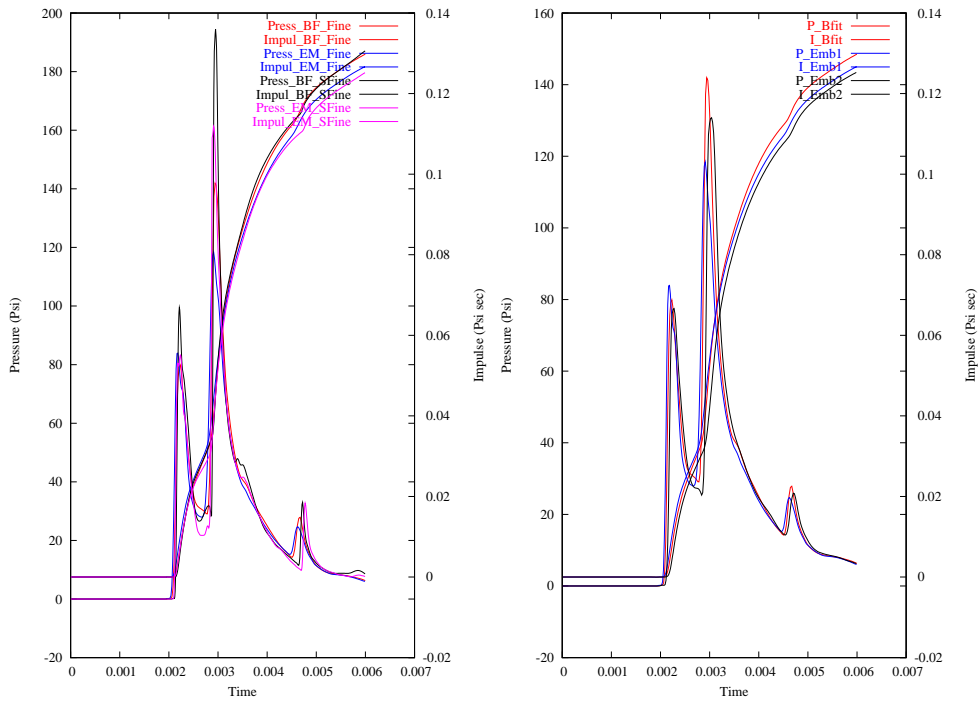


Figure 8l RC Column: BFFSF-EMBFSF and BFF-EMBF-EMBF2 (Back, Top)

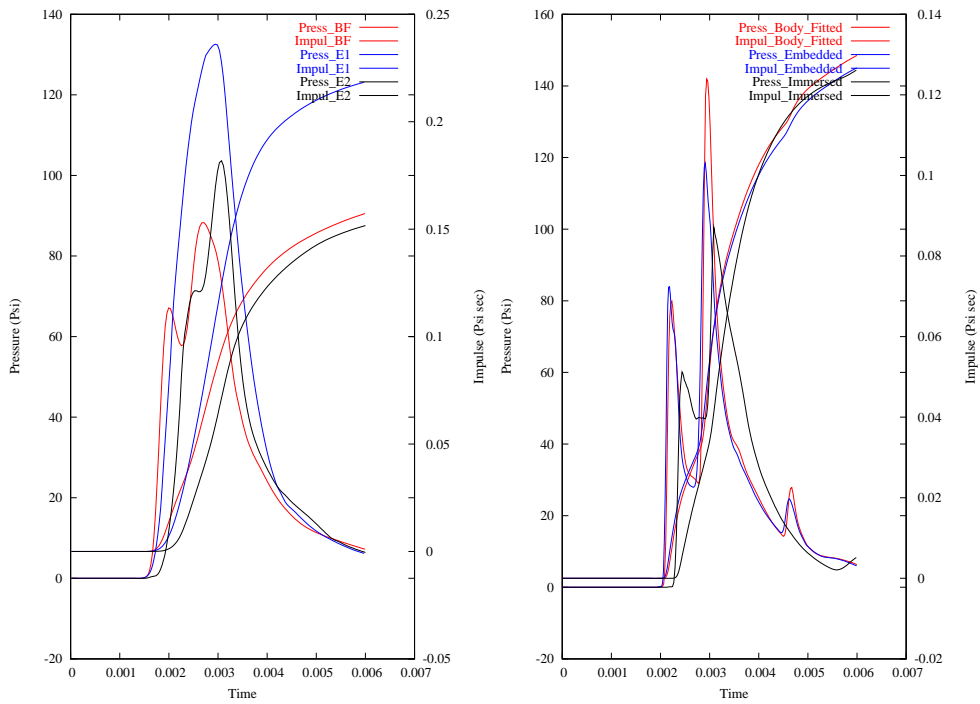


Figure 8m RC Column: BF-EMB-EMB2 and BFF-EMBF-IMMF (Back, Top)

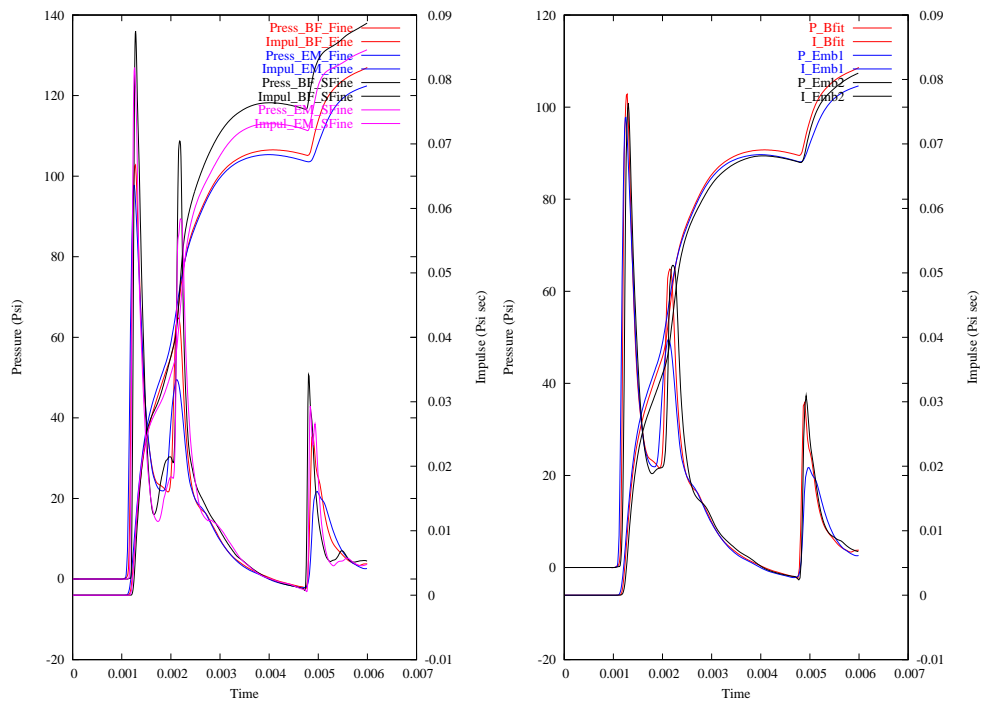


Figure 8n RC Column: BFFSF-EMBFSF and BFF-EMBF-EMBF2 (Back, Middle)

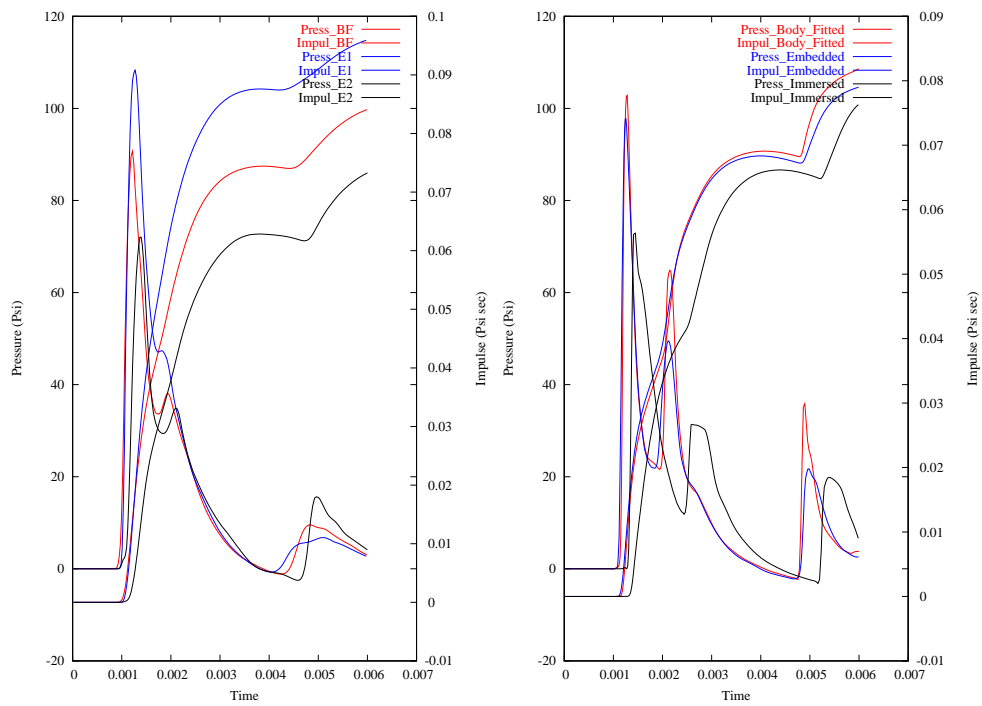


Figure 8o RC Column: BF-EMB-EMB2 and BFF-EMBF-IMMF (Back, Middle)

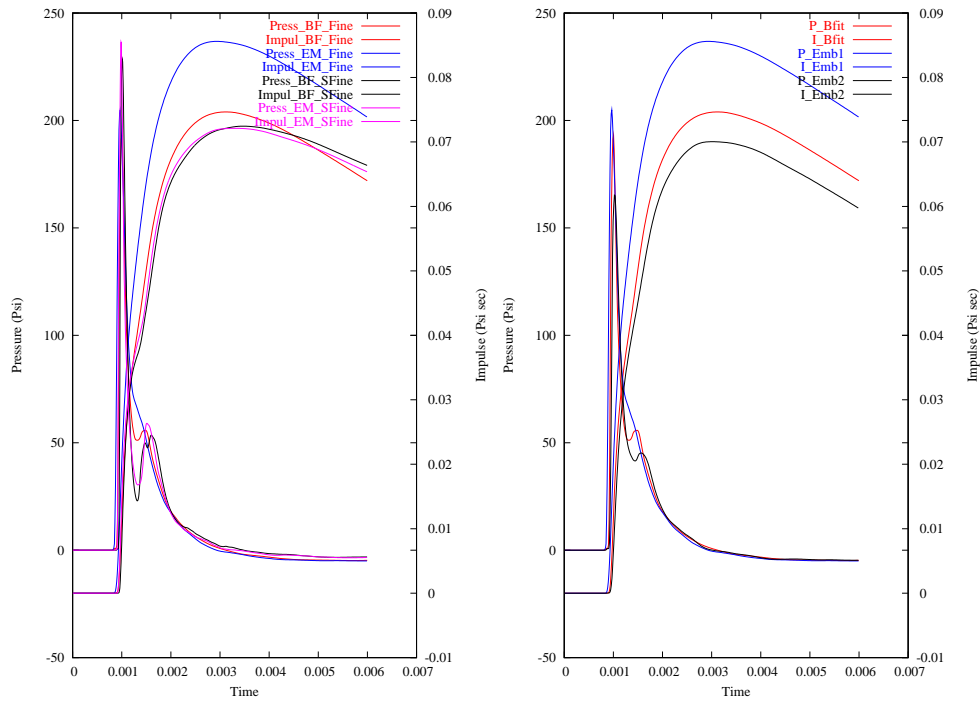


Figure 8p RC Column: BFFSF-EMBF2 and BFF-EMBF-EMBF2 (Back, Bottom)

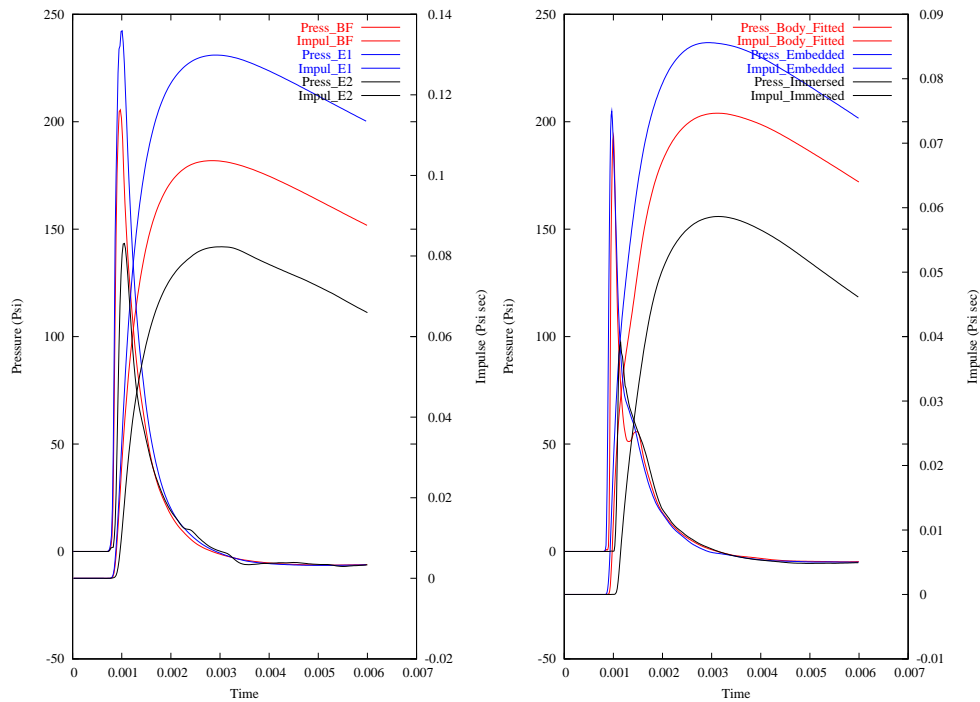


Figure 8q RC Column: BF-EMB-EMB2 and BFF-EMBF-IMMF (Back, Bottom)

5.2 Shock on Steel Column:

The second case considered is basically the same as the first case, except that the reinforced concrete column has been replaced by a steel column. The problem definition is shown in Figure 12a. Surface grids, which are indicative of the element size in the volume, are shown in Figure 12b,c. The grid size distribution is almost identical to the one used for the first example (grid size: 1,4, 2.4, 21 Mtets). The initial pressure distribution, obtained via interpolation from a detailed axisymmetric solution, as well as the pressure distribution upon

impact on the column at 1 msec are shown in Figures 12d,e. As this is a structure that is basically composed of thin shells, the immersed body approach was not attempted.

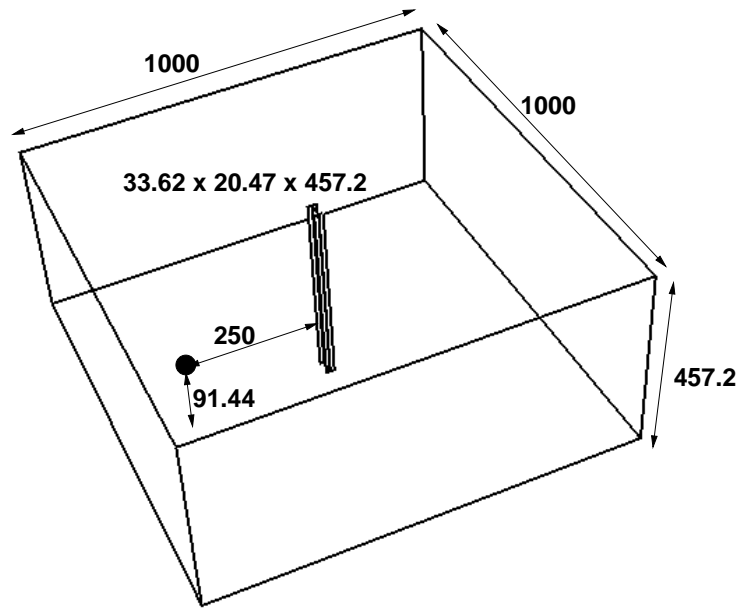
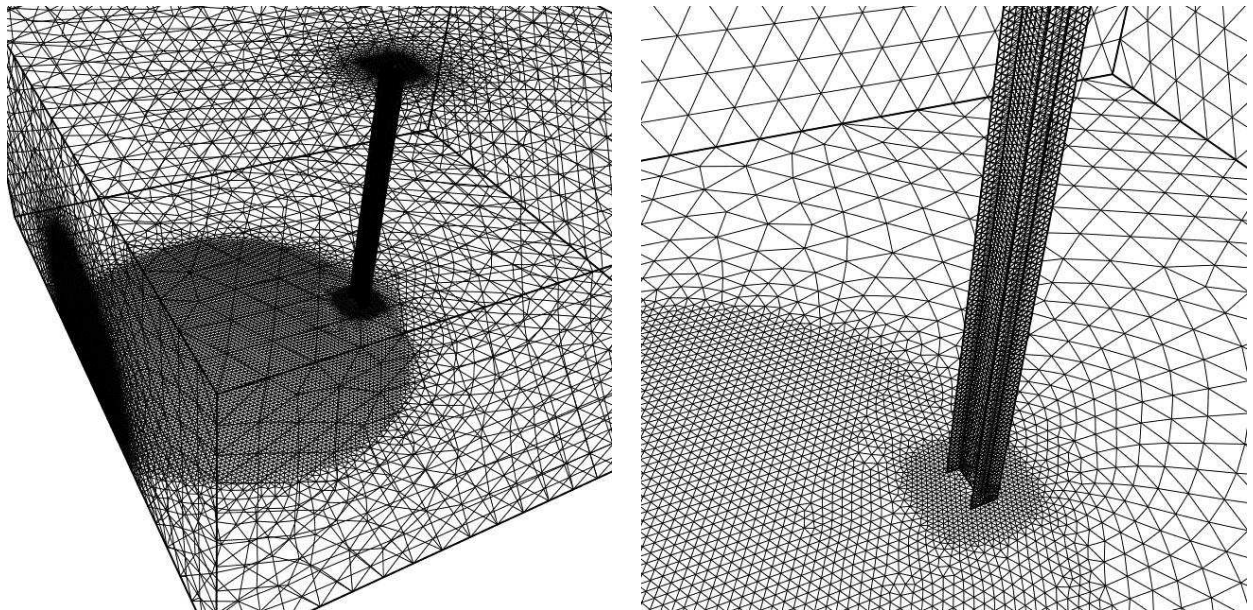
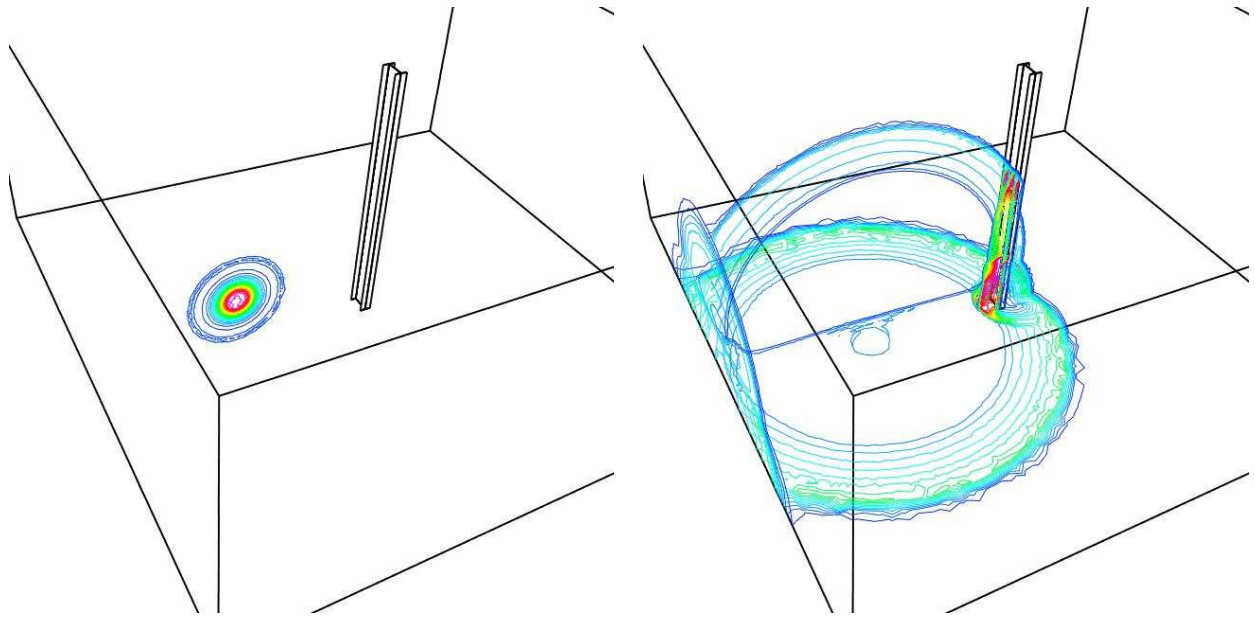


Figure 2a Shock on Steel Column: Problem Definition



Figures 2b,c Shock on Steel Column: Surface Grids



Figures 1d,e Shock on Steel Column: Pressures

As before, the pressures and impulses recorded at 6 different stations (3 in the front, going top to bottom, 3 in the back, going top to bottom) are shown in Figures 2g-l. Note that due to the internal reflections on the steel column walls, the pressure time histories show a larger number of peaks and subsequent expansions. The results obtained at the stations shown, as well as approximately 50 others that were placed on the column allow us to reach the following conclusions, which are essentially the same as with the previous example:

- a) The body-fitted and embedded-mesh methods of 1st and 2nd order converge to the same results as the grid is refined;
- b) The 2nd order embedded-mesh method is comparable in accuracy to the body-fitted results for grids of similar size;
- c) As expected, the 1st order embedded-mesh method exhibits more numerical dissipation than the 2nd order embedded-mesh method or the body-fitted approach; results of comparable accuracy require grids with element sizes on the boundary that are approximately half of those used for the other two approaches, unless a grid-independent result is already achieved;
- d) On grids of moderate size, while not as good as the body-fitted results, the 1st order embedded-mesh results provide impulse data that is within 10% of the grid-converged results;

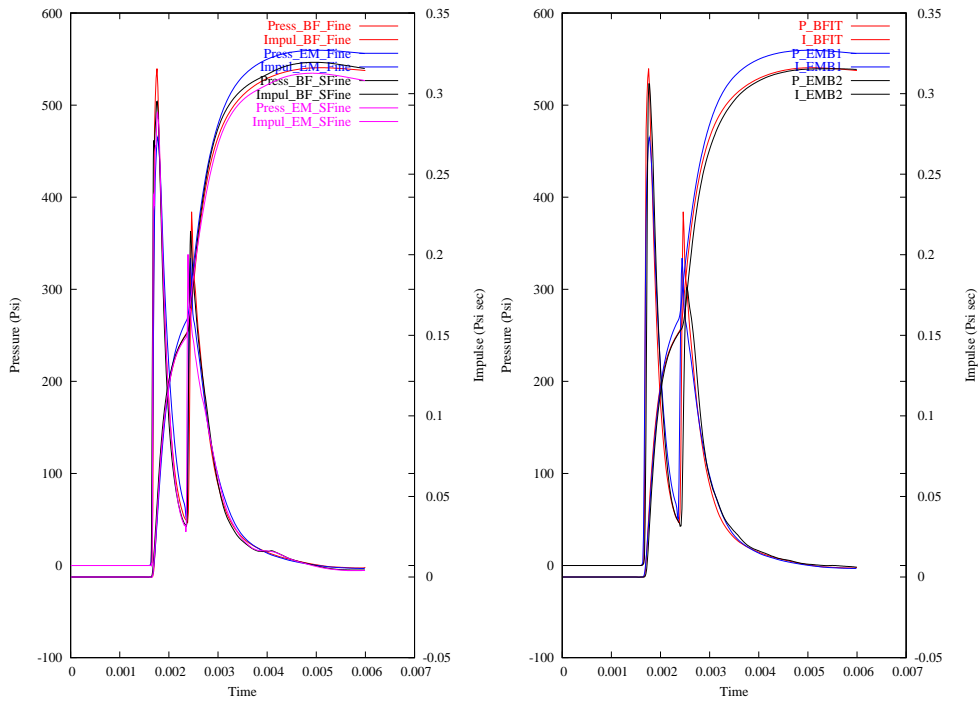


Figure 8f ST Column: BFFSF-EMBSF and BFF-EMBF-EMBF2 (Front, Top)

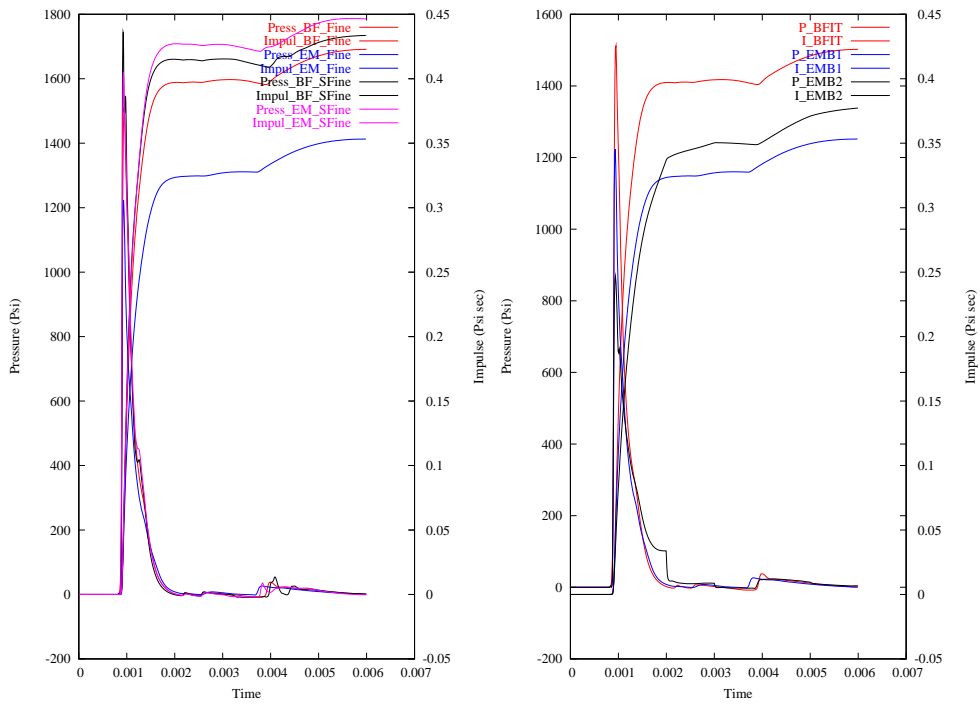


Figure 8f ST Column: BFFSF-EMBSF and BFF-EMBF-EMBF2 (Front, Middle)

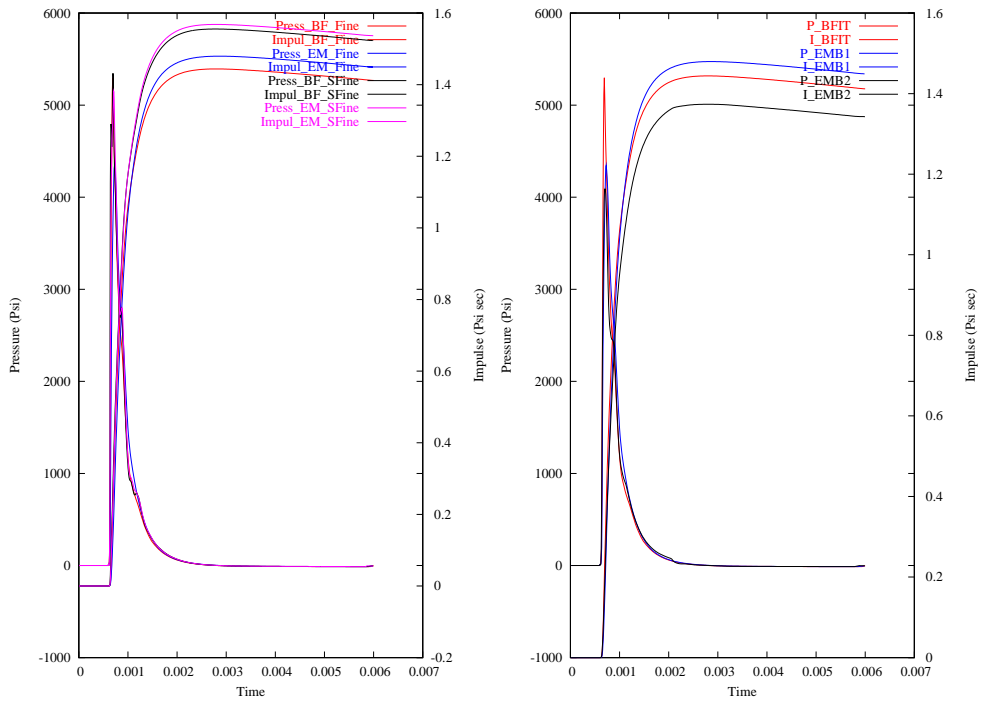


Figure 8f ST Column: BFFSF-EMBF2 and BFF-EMBF-EMBF2 (Front, Bottom)

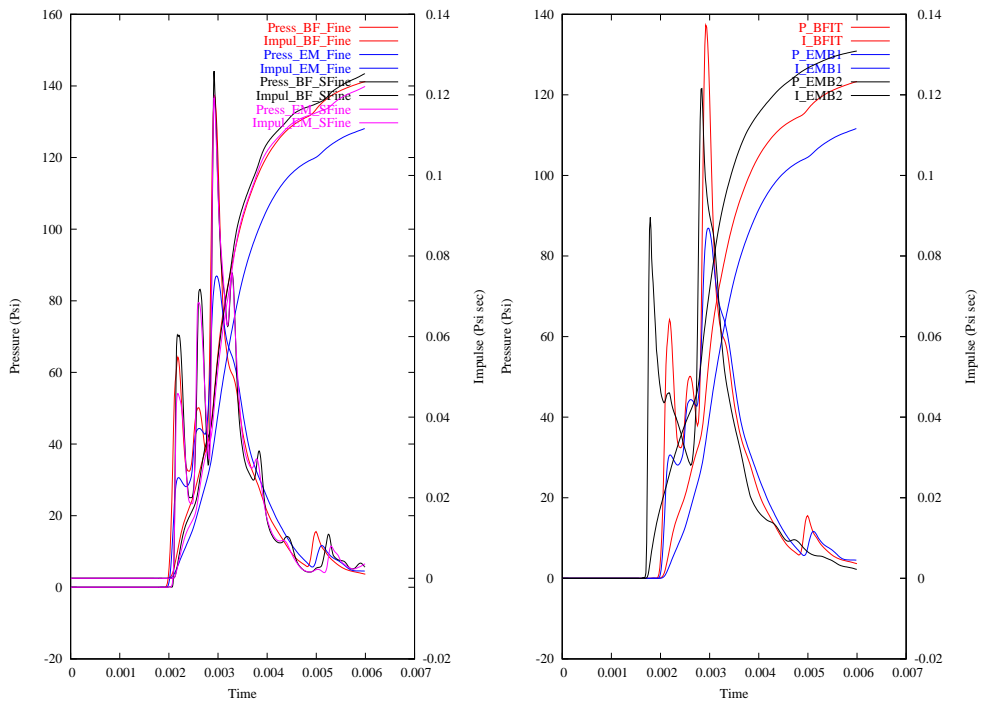


Figure 8f ST Column: BFFSF-EMBF2 and BFF-EMBF-EMBF2 (Back, Top)

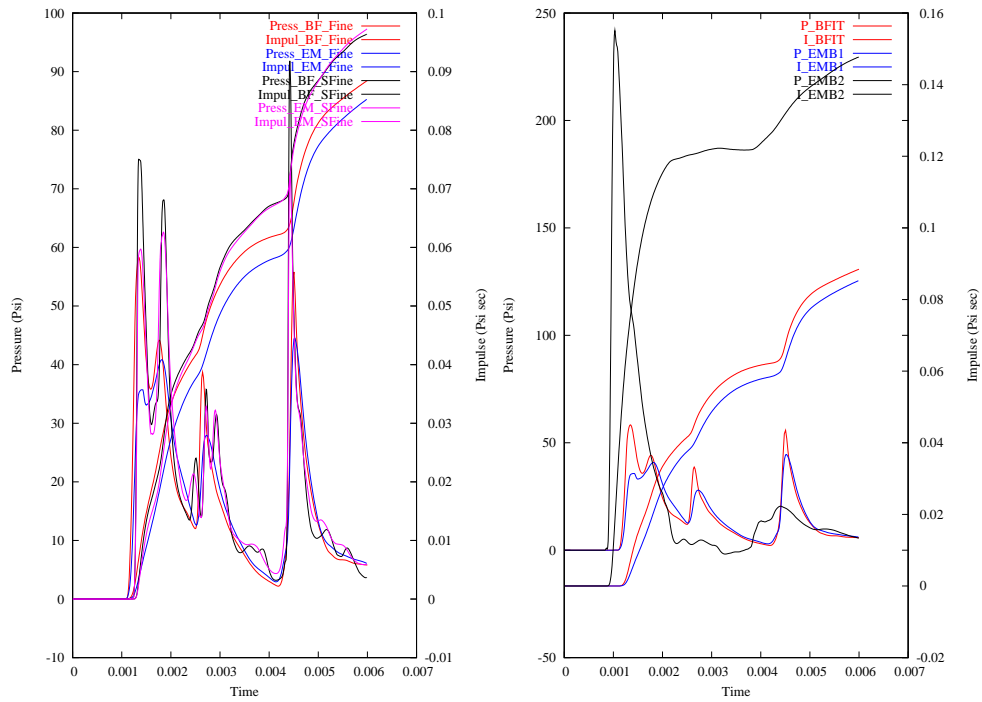


Figure 8f ST Column: BFFSF-EMBSF and BFF-EMBF-EMBF2 (Back, Middle)

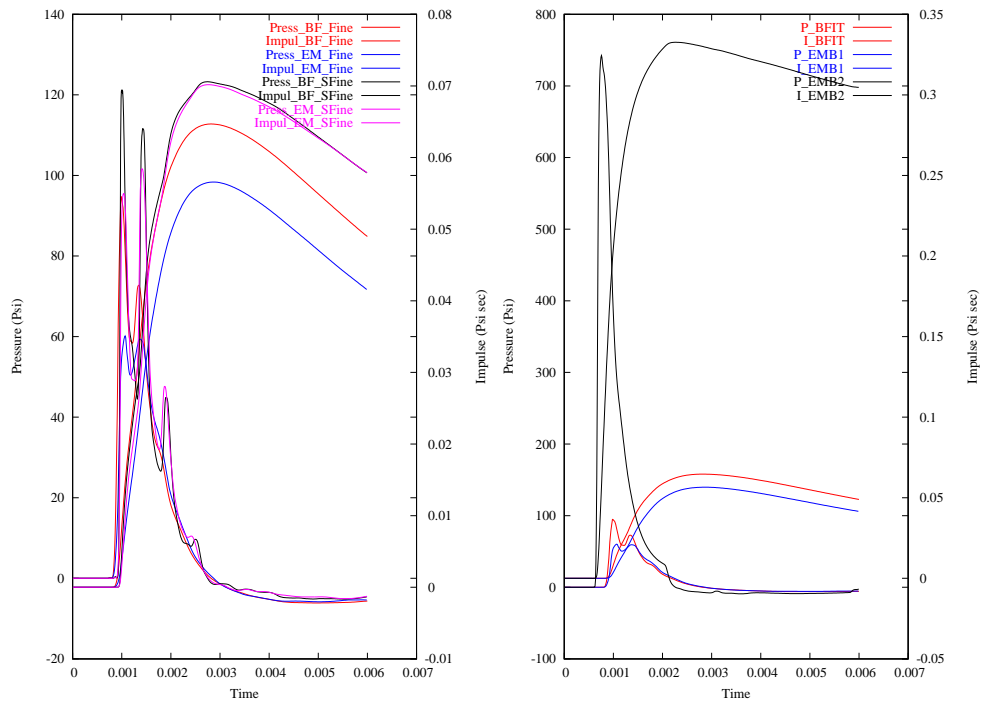


Figure 8f ST Column: BFFSF-EMBSF and BFF-EMBF-EMBF2 (Back, Bottom)

VI. CONCLUSIONS AND OUTLOOK

A reinforced concrete, as well as a steel column have been subjected to typical blast loads. The ensuing 3-D compressible flowfields have been computed with the same 3-D edge-based FEM-FCT solver using the body-fitted and embedded-mesh approach. The resulting pressures and impulses have been recorded at

stations close to the columns and compared for both approaches.

The results shown above, as well as others carried out over the years, lead us to the following conclusions:

- a) The body-fitted and embedded-mesh methods converge to the same results as the grid is refined;
- b) The body-fitted results exhibit faster convergence to the grid-independent (superfine mesh) result; therefore, if the manual labour required to set up a body-fitted domain is not excessive, and computing resources are scarce, this represents the recommended option;
- c) Conversely, if the manual labour required to set up a body-fitted domain is excessive, or computing resources are plentiful, the embedded-mesh approach is the recommended option;
- d) On grids of moderate size, while not as good as the body-fitted results, the embedded-mesh results are within 10% of the grid-converged results; given all other uncertainties in typical building response calculations (beam/girder connections, weldings, etc.), a 10% error is deemed acceptable for many cases;
- e) The fact that the body-fitted results exhibit lower numerical dissipation and higher accuracy should motivate the development of higher order boundary conditions for embedded-mesh methods. A considerable body of work exists in this regard [Lev99, Dad02, Mur03, Gil05]. However, some of these approaches become cumbersome and expensive when bodies or surfaces immersed in the flowfield start to move or break.

Given that computer performance and memory will continue to increase, the prospects for the embedded-mesh methods are very bright. Development of improved boundary condition treatment remains an active area of research, and should yield considerable improvements in the future. For blast-structure interaction problems, the embedded approach will undoubtedly gain favour as the treatment of complex beam connections, and the simultaneous treatment of 1-D (beams), 2-D (shells) and 3-D (solids) Finite Elements with breakup and spallation presents no further difficulties. This is in sharp contrast to body-fitted CFD approaches, where watertight surfaces need to be provided, something that is both difficult and error-prone.

References

- ¹M.J. Aftosmis, M.J. Berger and G. Adomavicius - A Parallel Multilevel Method for Adaptively Refined Cartesian Grids with Embedded Boundaries; *AIAA-00-0808* (2000).
- ²P. Angot, C.-H. Bruneau and P. Fabrie - A Penalization Method to Take Into Account Obstacles in Incompressible Viscous Flows; *Numerische Mathematik* 81, 497-520 (1999).
- ³J.D. Baum, H. Luo and R. Löhner - Numerical Simulation of a Blast Withing a Multi-Room Shelter; pp. 451-463 in *Proc. MABS-13 Conf.* The Hague, Netherlands, September (1993).
- ⁴J.D. Baum, H. Luo and R. Löhner - Numerical Simulation of Blast in the World Trade Center; *AIAA-95-0085* (1995).
- ⁵J.D. Baum, E. Mestreau, H. Luo, R. Löhner, D. Pelessone and Ch. Charman - Modeling Structural Response to Blast Loading Using a Coupled CFD/CSD Methodology; *Proc. Des. An. Prot. Struct. Impact/ Impulsive/ Shock Loads (DAPSIL)*, Tokyo, Japan, December (2003).
- ⁶J.R. Cezral and R. Löhner - Interactive On-Line Visualization and Collaboration for Parallel Unstructured Multidisciplinary Applications; *AIAA-98-0077* (1998).
- ⁷Y. Cho, S. Boluriaan and P.J. Morris - Immersed Boundary Method for Viscous Flow Around Moving Bodies; *AIAA-06-1089* (2006).
- ⁸D.K. Clarke, H.A. Hassan and M.D. Salas - Euler Calculations for Multielement Airfoils Using Cartesian Grids; *AIAA-85-0291* (1985).
- ⁹A. Dadone and B. Grossman - An Immersed Boundary Methodology for Inviscid Flows on Cartesian Grids; *AIAA-02-1059* (2002).
- ¹⁰E.A. Fadlun, R. Verzicco, P. Orlando and J. Moud-Yusof - Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional Complex Flow Simulations; *J. Comp. Phys.* 161, 33-60 (2000).
- ¹¹A. Gilmanov and F. Sotiropoulos - A Hybrid Cartesian/Immersed Boundary Method for Simulating Flows with 3-D, Geometrically Complex Moving Bodies; *J. Comp. Phys.* 207, 2, 457-492 (2005).
- ¹²D. Goldstein, R. Handler and L. Sirovich - Modeling a No-Slip Flow Boundary with an External Force Field; *J. Comp. Phys.* 105, 354366 (1993).
- ¹³S.L. Karman - SPLITFLOW: A 3-D Unstructured Cartesian/ Prismatic Grid CFD Code for Complex Geometries; *AIAA-95-0343* (1995).
- ¹⁴J. Kim, D. Kim and H. Choi - An Immersed-Boundary Finite-Volume Method for Simulation of Flow in Complex Geometries; *J. Comp. Phys.* 171, 132-150 (2001).
- ¹⁵A.M. Landsberg and J.P. Boris - The Virtual Cell Embedding Method: A Simple Approach for Gridding Complex Geometries; *AIAA-97-1982* (1997).

- ¹⁶R.J. LeVeque and D. Calhoun - Cartesian Grid Methods for Fluid Flow in Complex Geometries; pp. 117-143 in *Computational Modeling in Biological Fluid Dynamics* (L. J. Fauci and S. Gueron, eds.), IMA Volumes in Mathematics and its Applications 124, Springer-Verlag (2001).
- ¹⁷R. Löhner, K. Morgan, J. Peraire and M. Vahdati - Finite Element Flux-Corrected Transport (FEM-FCT) for the Euler and Navier-Stokes Equations; ICASE Rep. 87-4, *Int. J. Num. Meth. Fluids* 7, 1093-1109 (1987).
- ¹⁸R. Löhner and P. Parikh - Three-Dimensional Grid Generation by the Advancing Front Method; *Int. J. Num. Meth. Fluids* 8, 1135-1149 (1988).
- ¹⁹R. Löhner, P. Parikh and C. Gumbert - Some Algorithmic Problems of Plotting Codes for Unstructured Grids; AIAA-89-1981-CP (1989).
- ²⁰R. Löhner and J.D. Baum - Adaptive H-Refinement on 3-D Unstructured Grids for Transient Problems; *Int. J. Num. Meth. Fluids* 14, 1407-1419 (1992).
- ²¹R. Löhner - Extensions and Improvements of the Advancing Front Grid Generation Technique; *Comm. Num. Meth. Eng.* 12, 683-702 (1996).
- ²²R. Löhner, Chi Yang, J. Cebal, O. Soto, F. Camelli, J.D. Baum, H. Luo, E. Mestreau, D. Sharov, R. Ramamurti, W. Sandberg and Ch. Oh - Advances in FEFLO; AIAA-01-0592 (2001).
- ²³R. Löhner, Chi Yang, J. Cebal, O. Soto, F. Camelli, J.D. Baum, H. Luo, E. Mestreau and D. Sharov - Advances in FEFLO; AIAA-02-1024 (2002).
- ²⁴R. Löhner, J.D. Baum, E. Mestreau, D. Sharov, C. Charman and D. Pelessone - Adaptive Embedded Unstructured Grid Methods; *Int. J. Num. Meth. Eng.* 60, 641-660 (2004).
- ²⁵J.E. Melton, M.J. Berger and M.J. Aftosmis - 3-D Applications of a Cartesian Grid Euler Method; AIAA-93-0853-CP (1993).
- ²⁶R. Mittal and G. Iaccarino - Immersed Boundary Methods; *Annu. Rev. Fluid Mech.* 37, 239-261 (2005).
- ²⁷J. Mohd-Yusof - Combined Immersed-Boundary/B-Spline Methods for Simulations of Flow in Complex Geometries; *CTR Annual Research Briefs*, NASA Ames Research Center/ Stanford Univ., 317-327 (1997).
- ²⁸S.M. Murman, M.J. Aftosmis and M.J. Berger - Implicit Approaches for Moving Boundaries in a 3-D Cartesian Method; AIAA-03-1119 (2003).
- ²⁹R.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).
- ³⁰C.S. Peskin - The Immersed Boundary Method; *Acta Numerica* 11, 479-517 (2002).
- ³¹S. Del Pino and O. Pironneau - Fictitious Domain Methods and Freefem3d; *Proc. ECCOMAS CFD Conf.*, Swansea, Wales (2001).
- ³²J.J. Quirk - A Cartesian Grid Approach with Hierarchical Refinement for Compressible Flows; *NASA CR-194938, ICASE Report No. 94-51*, (1994).
- ³³D. de Zeeuw and K. Powell - An Adaptively-Refined Cartesian Mesh Solver for the Euler Equations; AIAA-91-1542 (1991).