A Brief Description of New Algorithms incorporated into CTH: A model for Rigid Obstacles and an Interface for Coupling with Structural Codes

David L. Littlefield
Texas Institute for Computational and Applied Mathematics
The University of Texas at Austin
email: littlefield@ticam.utexas.edu

Abstract

The implementation of two new algorithms for the Eulerian shock physics code CTH are described. The first algorithm is a rigid obstacle algorithm and allows the insertion of non-deforming bodies, either fixed or with a prescribed velocity, into the Eulerian mesh. The second algorithm is a structural code interface that allows for the coupling of loads from a Eulerian computation onto the surfaces of a finite element mesh. Presently this interface is configured to work with finite element meshes generated for DYNA3D. Together, these two new algorithms represent a significant enhancement to available computational tools for modeling blast loads onto structures. Example calculations demonstrate the utility of these algorithms for simulation of blast damage to structures.

Introduction

Researchers from TICAM (Texas Institute for Computational and Applied Mathematics) at the University of Texas at Austin have been developing new methods for the simulation of damaged structures, in a project sponsored by the DoD HPCMO (High Performance Computing Modernization Program Office) under the program called PET (Programming, Education and Training). Part of this development has centered on the development of improvements to CSM (Computational Structural Mechanics) legacy codes such as CTH.

CTH is a multi-material wave propagation code used by many analysts in the DoD user community to simulate large deformations, large strain rates, and strong shocks in solids, liquids and gases [1]. In recent years this code has become the preferred tool in the DoD for modeling large deformation, large strain rate, highly transient behavior. CTH employs a discretization of physical events based on a finite volume formulation of very general forms of the continuum equations. As such, it can be applied to a wide variety of problems. The computational mesh used in CTH is Eulerian; materials and material interfaces are permitted to flow through the mesh as the calculation proceeds.

The development and testing of a few new algorithms are described herein. The first algorithm, referred to as a *rigid obstacle algorithm*, permits the addition of non-deforming bodies of arbitrary shape into a Eulerian mesh. In its current form, the algorithm permits the insertion of obstacles that are either stationary or moving with a

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **2005**	2. REPORT TYPE	3. DATES COVERED **-**

4. TITLE AND SUBTITLE
A Brief Description of New Algorithms incorporated into CTH: A model for Rigid Obstacles and an Interface for Coupling with Structural Codes

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
DOD High Performance Computing Modernization Program Ofc,Programming Environment & Training (PET),1010 North Glebee Road Suite 510,Arlington,VA,22201

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT
The implementation of two new algorithms for the Eulerian shock physics code CTH are described. The first algorithm is a rigid obstacle algorithm and allows the insertion of non-deforming bodies, either fixed or with a prescribed velocity, into the Eulerian mesh. The second algorithm is a structural code interface that allows for the coupling of loads from a Eulerian computation onto the surfaces of a finite element mesh. Presently this interface is configured to work with finite element meshes generated for DYNA3D. Together, these two new algorithms represent a significant enhancement to available computational tools for modeling blast loads onto structures. Example calculations demonstrate the utility of these algorithms for simulation of blast damage to structures.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**13**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

prescribed velocity. For ease of implementation, the algorithm makes use of the material insertion routines already in place in CTH, with rigid obstacles being treated as a special material. The obstacles are assumed to have large masses, as such; the loads imparted on them by the surrounding deformable materials do not affect their motion. However, if the need arises for the addition of rigid body kinematics to these obstacles, hooks have been put into place for this option to be included.

The second algorithm, referred to as a *structural code interface*, allows for coupling of Eulerian and Lagrangian calculations. The coupling implemented here is unidirectional; that is, loads on the boundary of the finite element mesh determined from the Eulerian calculation are communicated to the structural code, but not vice-versa. A common application for coupling of this type is the determination of blast loads on structural members, since the motion of the structure typically occurs on a time scale much longer than the formation and propagation of the blast wave (i.e. the motion of the structure does not significantly affect the magnitude of the blast load). This is the simplest type of coupling possible for Eulerian/Lagrangian computations, and has been attempted before. However, the present treatment is somewhat unique in that the coupling is implemented in a manner that requires very little interaction from the user – the Eulerian computation automatically generates the input needed for the structural calculation. Presently the interface is designed to work with finite element models generated for DYNA3D; however, provisions have been put in place to couple with other Lagrangian finite element codes.

The Algorithms

In the development of algorithms for CTH, or any legacy code, it is prudent to consider the organization of the code and to make use existing routines to the extent possible. For example, in the present case, a great deal of effort had already been put into the development of material insertion routines for a large number of three-dimensional geometries. As such, the rigid obstacle algorithm was designed to make use of the existing material insertion coding. Another example is the tracer point routines already in place in CTH, which provides a means of extracting complete history data at discrete points in a Eulerian mesh (in fact, this feature had been used previously by analysts at ERDC [2] to extract blast loads computed from Eulerian calculations). While these routines provide a means for extracting the needed data, it was decided that the functionality of these routines was much too general for application to this specific problem. For example, history records in CTH provide complete information on the kinematic, stress and thermodynamic state at the tracer point, whereas in this case only the pressure is needed. As such, a subset of these routines was first developed to generate data for these special tracers, referred to as *coupling tracers*, and contained only the needed features, however; the existing coding was used as a guide in development of these routines.

Rigid Obstacle Algorithm

As was previously stated, the material insertion routines already in place in CTH were re-used for the rigid obstacle algorithm to allow the insertion of three-dimensional objects of arbitrary shape into the Eulerian mesh. However, since this implementation treats rigid obstacles as a special material, the thermodynamic state must also be initialized. Here, the goal was the to minimize the influence of the rigid material on the speed and efficiency of the computation, so the density, pressure, energy and sound speed were all set to zero. This, in effect, leaves control of the maximum allowable time step to thermodynamic state of the deforming material, which significantly speeds up the wall clock time in blast loading calculations (compared to similar calculations where steel or concrete is inserted into the mesh to mimic rigid material). In one recent example the computation time was reduced by over 30% [3]. Further increases in speed could be realized if the kinematic and thermodynamic state updates for cells containing only rigid material were skipped over entirely; this will be incorporated at a later date.

The rigid obstacle algorithm was formulated in the simplest manner possible that would still yield physically meaningful results. In the algorithm, the velocities across cell faces that contain a rigid material were set equal to the velocity of the rigid material (see Fig. 1). This has the effect of requiring the deforming material to "stick" to the surface of the rigid material, not allowing deforming materials to slide tangentially to a rigid surface. Furthermore, in the case of cells containing mixed rigid and deformable materials, it also "locks" a layer of deforming material adjacent to the rigid material, this has an influence on the placement of pressure extraction points used in the structural code interface, since the pressure also does not change in this layer. However, this algorithm has the advantage of ease of implementation since changes were required only for Lagrangian phase of the computation (i.e. no remap step modifications were necessary). A more realistic algorithm would permit slip of deforming materials and would not lock deforming material next to rigid material; these additions require modifications to the remap step of the computation and will be completed at a later date.

Modifications to the mixed-cell thermodynamics routines were also necessary to determine a reasonable cell pressure. In cells with rigid material, the rigid volume was first subtracted from the cell volume, so that rigid materials do not participate in the mixed-cell energy balance. This is a consistent manner for treatment of rigid material in the mixed-cell energy balance since no volume change occurs for that material.

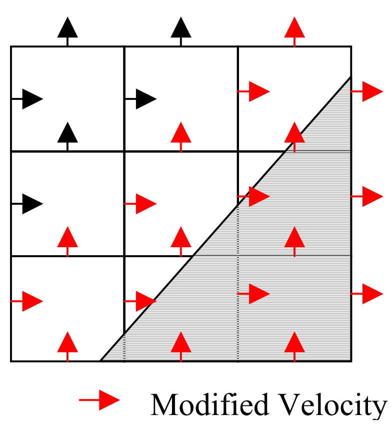


Figure 1. Representation of rigid obstacle algorithm

An illustration of the typical input required in a CTH calculation is shown in Fig. 2. In this example, Material 3 is designated rigid in the Equation of State input, so that any of Material 3 inserted during the Material Insertion input will be rigid material. This description should feel natural to the CTH user.

```
eos
  mat1 ses air
  mat2=mat1
  mat3 rigid
endeos
```

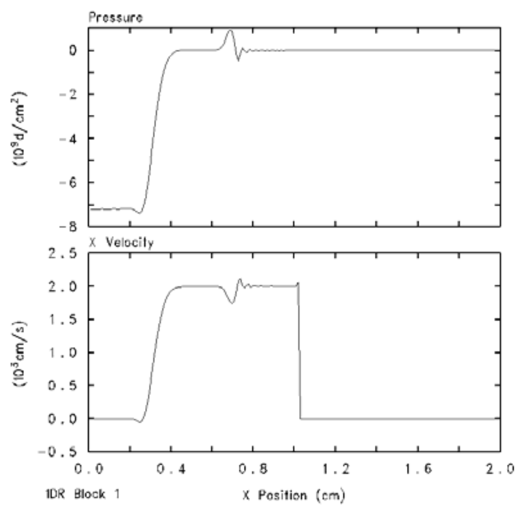
(a)

```
insertion
  block=1
*
    package column1
      material=3
      numsub=10
      insert box
        p1=0. 0. 0.
        p2=3.81 182.88 -7.62
      endinsert
    endpackage
*
    package column2
      material=3
      numsub=10
      insert box
        p1=0. 0. -144.78
        p2=3.81 182.88 -152.4
      endinsert
    endpackage
…
endblock
endinsert
```

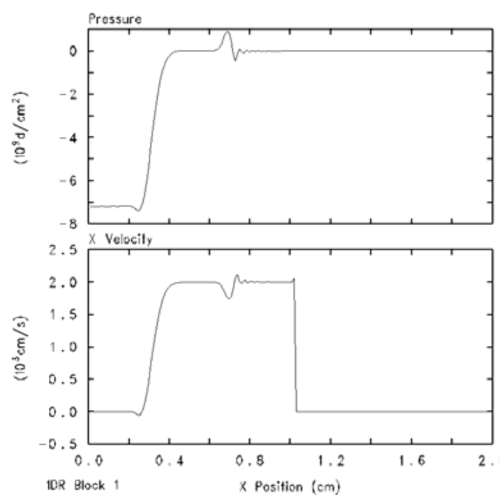
(b)

Figure 2. Input Example: (a) equation of state input and (b) material insertion input.

After implementation, a verification example was run to insure proper functionality of the routines. A one-dimensional calculation of a 1 cm thick steel bar hitting a rigid barrier at $x = 0$, with a velocity of 20 m/sec, was performed. For comparison, the rigid barrier was eliminated and replaced with a reflective boundary at $x = 0$. Results from this test are shown in Fig. 2, where the pressure and velocity are shown 5 μs after impact. In both cases the results are identical, as they must be. One artifact evident in these results is that the interface between the rigid and deforming materials unrealistically allows tensile states to exist between them. This is a well-known limitation in Eulerian simulations that use a single velocity for all materials along a cell edge. In this case the interface between the rigid and deforming materials falls directly on a cell edge, and the velocity across this edge is set equal to that of the rigid material.



(a)

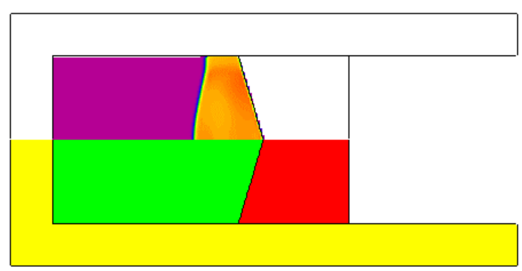


(b)

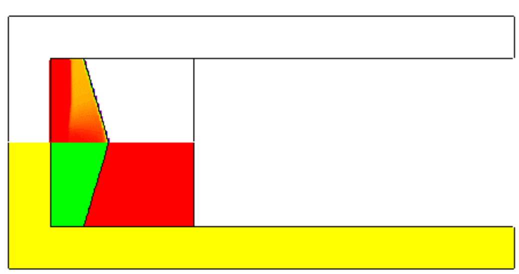
Figure 3. Code verification example: (a) rigid material for $x \leq 0$, and (b) reflective boundary condition applied at x = 0.

Results from a few sample calculations are shown in Figs. 4 and 5. Figure 4 shows the compression of air by a rigid piston, moving at 1000 m/s, in a rigid enclosure. The shock wave formed in the air at 20 μs is clearly seen. As is evident, neither the piston nor the enclosure are deforming as a result of the gas pressure. In this example the mesh was

(a)



(b)

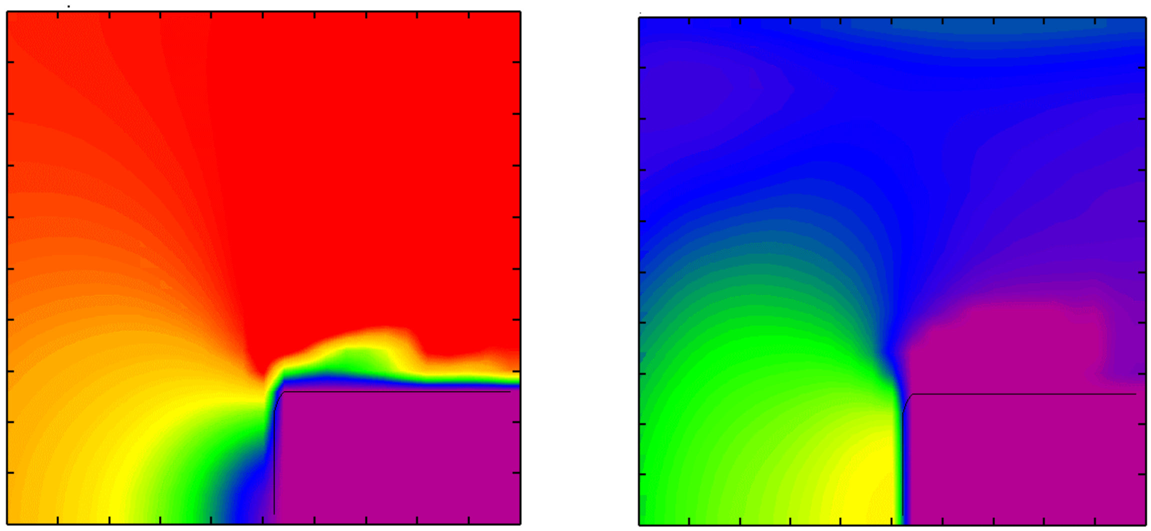


(c)

Figure 4. Compression of air in a rigid enclosure by a rigid piston moving at 1000 m/s: (a) 0 μs, (b) 20 μs, and (c) 38 μs.

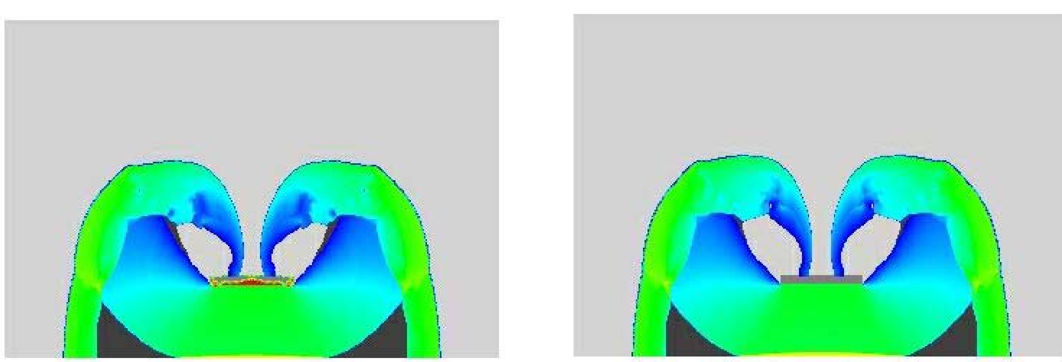
aligned with rigid interfaces, this is a requirement since CTH uses a single velocity for all materials in a given cell. Generalizing this code to permit multi-material velocities would require a major undertaking and was not attempted in this work.

Figure 5 shows pressure and velocity profiles for the flow of water at 200 m/s past a rigid plate. It is evident from the figures that the plate remains rigid, also apparent is a layer for deformable material that is "locked" to the rigid surface. This is an artifact of the algorithm mentioned earlier; this layer is motionless and its pressure is also zero.



(a) (b)

Figure 5. Flow of water at 200 m/s past a rigid protrusion: (a) Velocity magnitude and (b) Pressure.



(a) (b)

Figure 6. Simulation of blast wave impinging on a barrier, 4 ms after detonation, for (a) a steel barrier, and (b) a perfectly rigid barrier.

Dr. Richard Weed installed a developmental version of CTH, containing these algorithms, at the ERDC MSRC. Shown in Fig. 6 are results from a test calculation he performed on the SGI Origin 3000 (ruby.wes.hpc.mil) using 16 processors. In this two-dimensional example, a blast wave emanating from the center of the bottom edge, impinging on a barrier positioned above it, is simulated. Figure 6a shows pressure contours for a steel barrier, 4 ms after detonation, and Figure 6b for a perfectly rigid barrier. As is expected, the pressure distribution is nearly identical in the two cases.

Structural Code Interface

As was previously stated, routines for generating load curves on the faces of a finite element structural model were developed using the existing CTH tracer particle routines as a guideline. New routines for the coupling tracers contain only a subset of the data generated from the existing tracer particle package; this greatly reduces memory requirements and size of the data arrays needed for the coupling.

The structural code interface also makes use of the rigid obstacle algorithm previously described in this paper. To insure proper functionality of the interface, geometry for the structure must be reproduced and inserted as rigid material into the Eulerian mesh. In its current form, the interface requires the user to perform this step manually, that is; the geometry from the finite element model must be reproduced in the CTH input. A planned improvement to the interface is to automate this step, so that rigid volumes are inserted automatically by reading input from the finite element model.

Setup of the coupling tracer locations is done automatically by first reading the finite element model input file and placing a tracer on the face of each element. The storage required for these tracers is allocated dynamically and not restricted to the 1000-particle maximum hard-coded into the existing CTH tracer routines. The cell used to record the pressure history, however, was not coincident with the location of the coupling tracer. It was necessary to use and adjacent cell because, as was described earlier, the rigid obstacle algorithm "locks" a layer of deforming material directly adjacent to interface between the rigid and deforming material. The cell chosen to record the pressure was the first cell containing no rigid material along a direction coincident with the outward unit normal to the rigid material interface, as described by the diagram in Fig. 7.

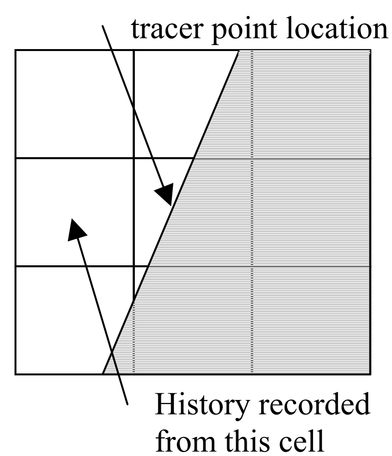


Figure 7. Location of history record for coupling tracers

The final step is generation of load curves from the pressure data collected at the coupling tracers. Surfaces in the finite element model used to generate the load curves are specified in the structural code input; a future improvement is to automate this step so that the location of surfaces for the load curves are determined automatically. Pressure histories along these surfaces are integrated over the load surface using data from the coupling tracers, so that the proper impulse is maintained. Since DYNA3D interpolates these load curves to give the pressure on an element face at the time it is needed, two additional history points with zero pressure were added to the end of each load curve.

```
fem
  ingrid
    units uscs
    infile 'ingrido'
    outfile 'preloads'
    tend 1.0
    dt 1.e-5
  endi
endf
```

(a)

```
tswg model for David Littlefield                                        88
large
*                                                                       88
large
*
*----------------- ANALYSIS INPUT DATA FOR DYNA3D 88 ------------------*
*
* Generated by Ingrid - Version # 1996e    (08/05/96)

…

*
*-------------------------- NODE DEFINITIONS --------------------------*
*
       1   7. 0.000000E+00        0.000000E+00        0.000000E+00          7.
       2   7. 0.000000E+00        0.000000E+00       -9.999999E-01          7.
       3   7. 0.000000E+00        0.000000E+00       -2.000000E+00          7.
       4   7. 0.000000E+00        0.000000E+00       -3.000000E+00          7.

…

*
*----------------- PRESSURE BOUNDARY CONDITION CARDS ------------------*
*
    1       1      73      77       5 1.000E+00 1.000E+00 1.000E+00 1.000E+00
    1       5      77      81       9 1.000E+00 1.000E+00 1.000E+00 1.000E+00
    1       9      81      85      13 1.000E+00 1.000E+00 1.000E+00 1.000E+00
    1      13      85      89      17 1.000E+00 1.000E+00 1.000E+00 1.000E+00
    1      17      89      93      21 1.000E+00 1.000E+00 1.000E+00 1.000E+00
…
```

(b)

Figure 8. Input Example: (a) Structural Code interface input and (b) Structural Code input file

Figure 8 illustrates the typical input required in CTH as well as the structure of a typical DYNA3D input file used in the coupling. As is evident, additional input required to CTH is minimal since the majority of the coupling tasks are performed without any user intervention.

Results from a sample calculation are summarized in Figs. 9 – 14. In this example, a simple blast/structure interaction problem was simulated. Setup for the geometry of the structure is depicted in Fig. 9, where an image of half the structure is shown (the left edge of the structure is a plane of symmetry). A one-eighth sphere shown in red in the figure represents the explosive charge; symmetry is assumed on the three orthogonal planes passing though the center of this sphere. This structure is a generic model that has been used in many structure/blast interaction tests at ERDC [4]. J. T. Baylot provided the author with a simple finite element

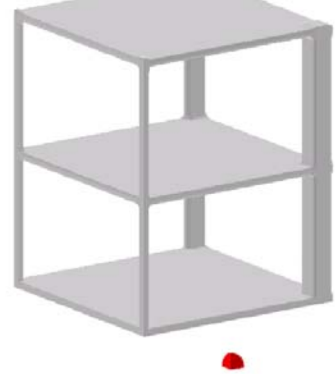


Figure 9. Blast-structure interaction problem setup

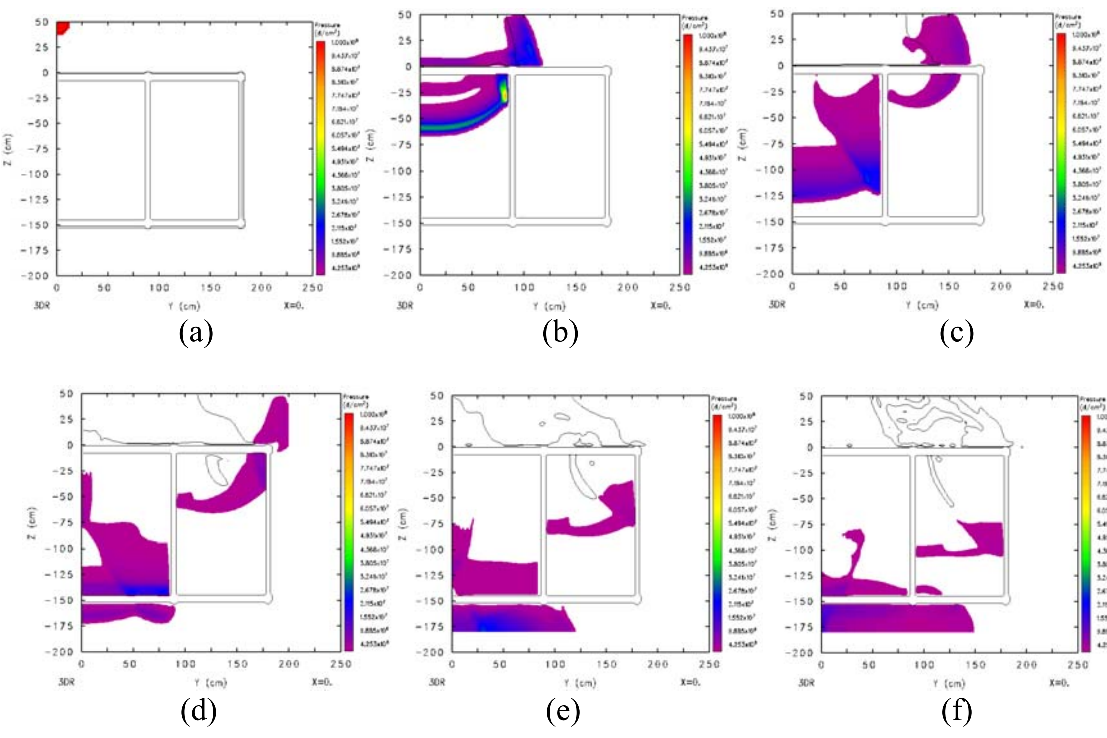


Figure 10. Two-Dimensional slices of pressure for *x* = 0 at various times: (a) 0 μs, (b) 400 μs, (c) 800 μs, (d) 1200 μs, (e) 1600 μs and (f) 2000 μs.

model of this structure, containing 25380 nodes and 19908 hexahedral elements.

Results from the simulation of the blast wave interaction on the structure are given in Figs. 10 – 12. The images are pressure contours and material interfaces from two-dimensional slices at three points through the cross section of the structure, shown at times up to 2 ms. The orientation of the structure in these images is different from that shown in Fig. 9; in this sequence the bottom floor of the structure appears on the left side of the image. Pressures as high as 70 bar are seen along the bottom edge of the second floor, but in general the overpressures are less than 10 bar. A three-dimensional view of the blast wave interaction is shown in Fig. 13, where material interfaces are shown at times up to 2 ms. In this image red depicts the explosive interface and blue the structure interface. The material interface propagates much like the blast wave except at a much slower velocity (this can also be seen from the material interface boundaries shown as black lines in Figs. 10 – 12).

Load curves generated from this calculation were used to compute the structural response in DYNA3D. The response of the structure to these loads is depicted in Fig. 14, where element surfaces are shown at various times. The constitutive behavior was assumed to be isotropic elastic, with density ρ, Poisson's ratio ν and Young's Modulus *E* of 6.89 x 10^{-3} lbm/in^3, 0.15 and 1.5 x 10^3 psi, respectively. These values are typical of concrete with the exception of *E*, which was artificially lowered by a factor of 1000 to exaggerate

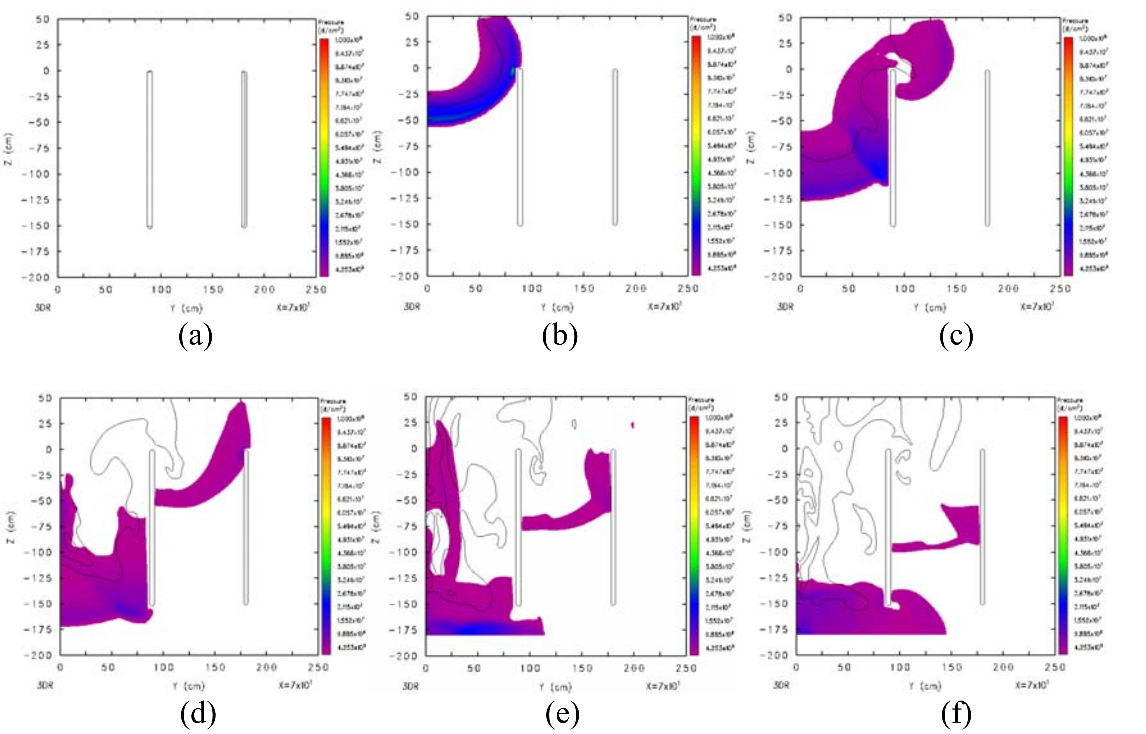


Figure 11. Two-Dimensional slices of pressure for *x* = 70 cm at various times: (a) 0 μs, (b) 400 μs, (c) 800 μs, (d) 1200 μs, (e) 1600 μs and (f) 2000 μs.

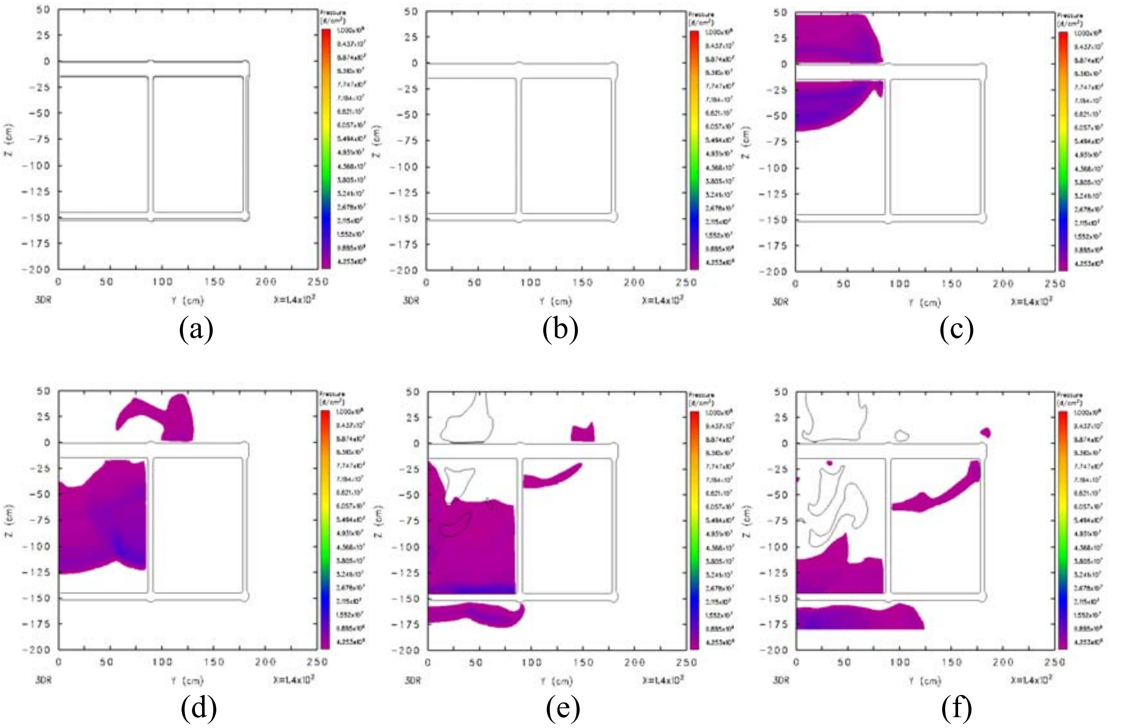


Figure 12. Two-Dimensional slices of pressure for x = 140 cm at various times: (a) 0 μs, (b) 400 μs, (c) 800 μs, (d) 1200 μs, (e) 1600 μs and (f) 2000 μs.

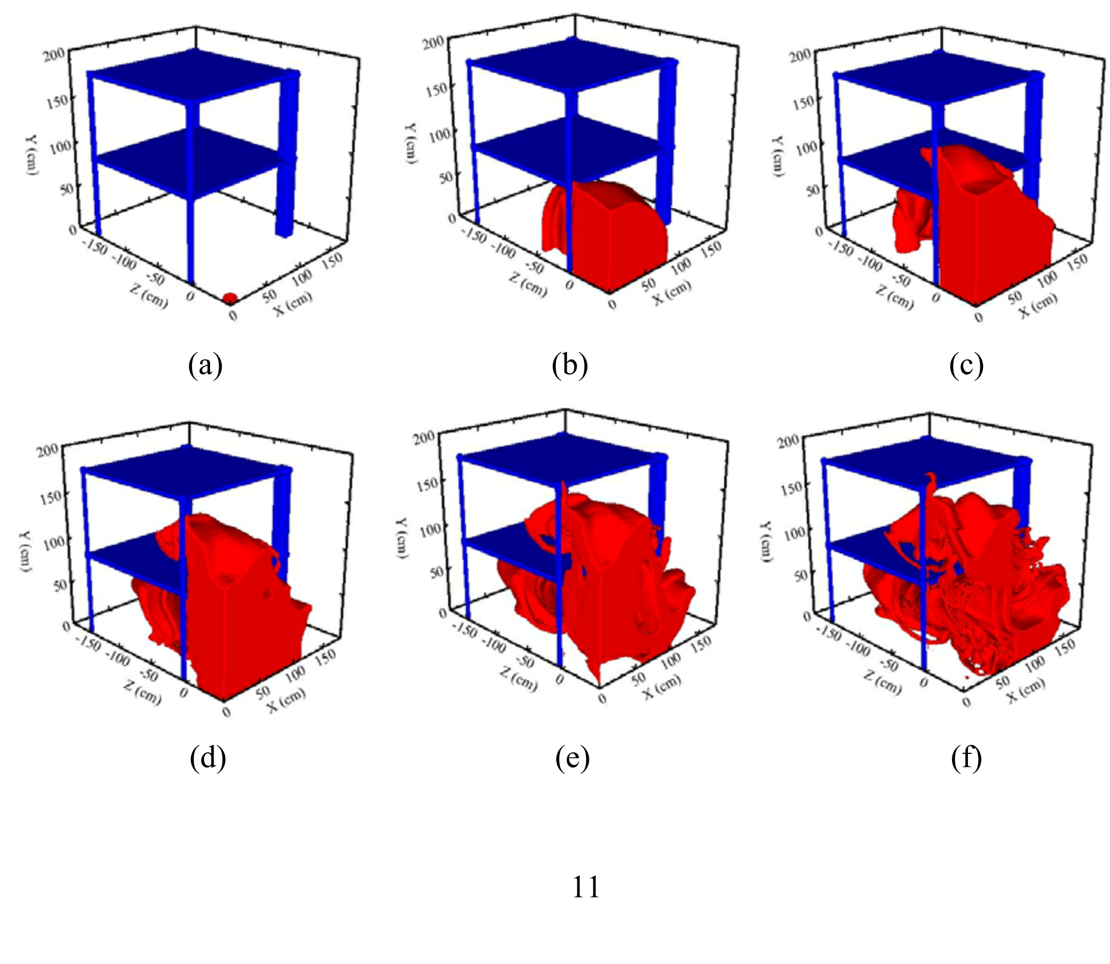


Figure 13. Material contours at (a) 0 μs, (b) 400 μs, (c) 800 μs, (d) 1200 μs, (e) 1600 μs and (f) 2000 μs.

the deformation. As is evident from this sequence of images, the majority of deformation occurs on the second floor; this is where the largest blast loads were seen in the Eulerian computation. It is also evident that the time scale of the deformation is much longer for the structure than for the blast wave propagation; the structural response computation was run for 120 ms compared to 2 ms for the blast wave calculation. It is this behavior that permits coupling of this type, where one-way communication occurs between the two calculations.

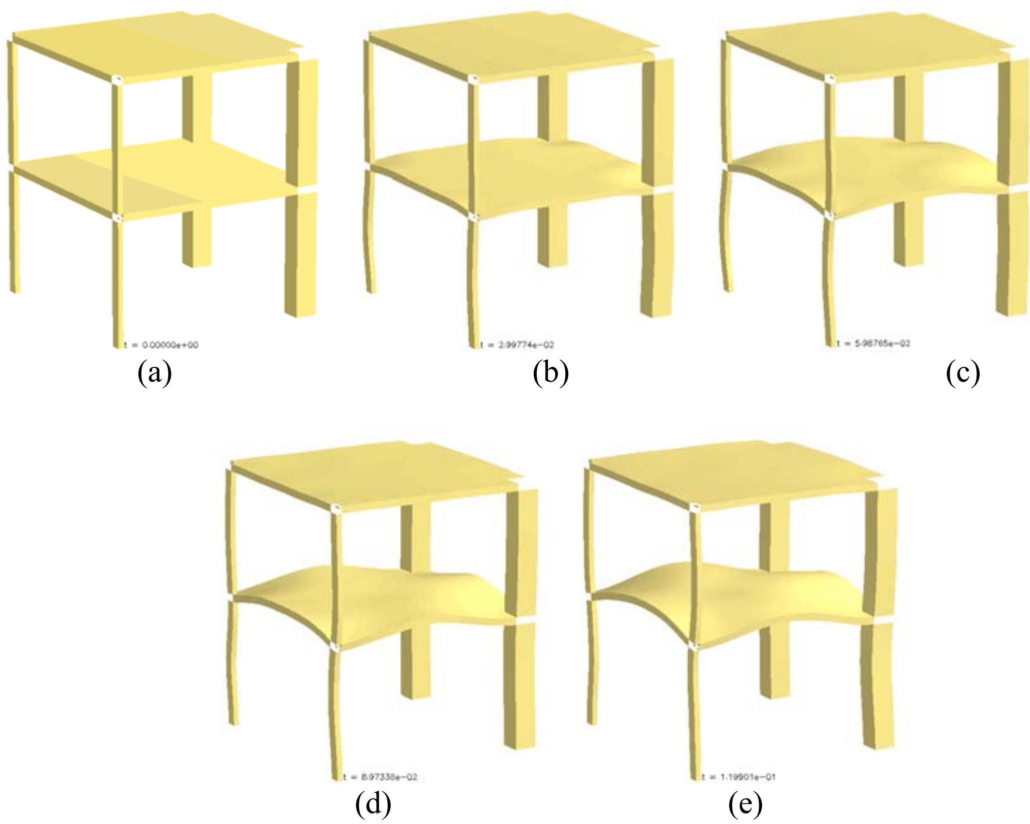


Figure 14. Structural Response at (a) 0 ms, (b) 30 ms, (c) 60 ms, (d) 90 ms and (e) 120 ms.

Conclusion

The implementation of two new algorithms for the Eulerian shock physics code CTH has been described in this paper. The first algorithm, referred to as a *rigid obstacle algorithm*, permits the insertion of non-deforming bodies into the Eulerian mesh. The second algorithm, referred to as a *structural code interface*, allows the coupling of loads from a Eulerian computation onto the surfaces of a finite element mesh. Example calculations have demonstrated the utility of these algorithms for modeling blast/structure interactions.

Presently these algorithms are not available in the production version of CTH, but will be made available soon in a special version installed at the ERDC MSRC. The POC at ERDC for this version is Dr. Richard Weed. Please contact the Dr. Weed by email at Richard.A.Weed@erdc.usace.army.mil if you are interested in using this developmental version.

Acknowledgements

The author thanks Dr. Richard Weed for many useful discussions, installation and testing of code at the ERDC MSRC, and debugging assistance. Dr. James Baylot provided the finite element model used in the example for the structural code interface. This work was supported by the DoD High Performance Computing Modernization Program EDRC Major Shared Resource Center through Programming Environment and Training (PET).

References

[1] McGlaun, J. M., Thompson, S. L. and Elrick, M. G. 1990. "CTH: A three dimensional shock wave physics code", *Int. J. Impact Engng.*, Vol. 10, 351 – 360.

[2] Private communication, T. L. Bevins, February 2001.

[3] Bevins, T. L., Baylot, J. T. and Littlefield, D. L, "Blast Barrier Effectiveness Simulations", Proceedings of the DoD HPCMP User's Group Conference, Biloxi, MS, June 18 – 21, 2001

[4] Private communication, J. T. Baylot, February 2001.