1-1-2014

A New Fluid-Structure Interaction Point-Projection Method

Philip Robert Ivancic

A new fluid-structure interaction point-projection method

By

Philip Robert Ivancic

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Aerospace Engineering
in the Department of. Aerospace Engineering

Mississippi State, Mississippi

December 2014

A new fluid-structure interaction point-projection method

By

Philip Robert Ivancic

Approved:

_____

J. Mark Janus
(Major Professor and Graduate Coordinator)

_____

David S. Thompson
(Committee Member)

_____

Richard A. Weed
(Committee Member)

_____

Jason M. Keith
Interim Dean
Bagley College of Engineering

Name: Philip Robert Ivancic

Date of Degree: December 13, 2014

Institution: Mississippi State University

Major Field: Aerospace Engineering

Major Professor: J. Mark Janus

Title of Study: A new fluid-structure interaction point-projection method

Pages in Study: 39

Candidate for Degree of Master of Science

A new point-projection method was developed to transfer loads and displacements in a two-way coupled fluid-structure interaction problem. The existing method involved projecting the load at each computational fluid dynamics (CFD) node onto a corresponding computation structural dynamics (CSD) element. The load is distributed to the CSD nodes on that element. However, the solution is not unique and will vary the projection. In the new method, a rigid pyramid element is built upon the CSD element and that encompasses the CFD node. Thus, the CFD load will be uniquely distributed to the CSD nodes. After the CSD code updates the CSD node location, the pyramid element can also be used to update the location of the CFD node.

This work describes a FORTRAN routine that coupled LS-DYNA to Loci/BLAST and tests conducted to test the validation, work conservation, and robustness of the routine.

ACKNOWLEDGEMENTS

I would like to thank my major professor Dr. J. Mark Janus and my other committee members Dr. David Thompson and Dr. Richard Weed for their valuable instruction and advice on this work.

I would also like to thank Dr. James Newman III, who developed the new point-projection method discussed in this thesis, for his advice for coming up with ways to test the work conservation of the method, as well as, his work in debugging the FORTRAN routine. And I would like to thank Dr. Edward Luke for his help with Loci/BLAST.

Finally, I would like to thank the Tank Automotive Research, Development, and Engineering Center (TARDEC) for providing the funding for this work and for providing a generic vehicle LS-DYNA model to test robustness of the new method.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

A major problem in an interdisciplinary engineering analysis is determining how to combine the techniques used in each individual discipline to solve the global problem. An example can be found in fluid-structure interaction problems. There are techniques within computational fluid dynamics (CFD) to solve for a flow around a body and, thereby, be able to determine pressure distribution applied to that body. Likewise, computational structure dynamics (CSD) can provide deformation and stress histories of the structure, which are usually the quantities of interest in an engineering analysis. The stress distribution could then be used to determine whether or not failure occurs.

A simple way to couple these two problems together would be to use a one-way coupling: a CFD solution is run first for the problem to get a pressure distribution history on the body, and that is then used as an input for the CSD. This process works well enough in some cases in which the time scale of the flow is much smaller than the response time of the material, but there will be other cases that require a higher fidelity, such as in explosions where deformation rates are high, and thus a two-way coupling is needed. In a two-way coupled FSI problem, the CFD simulation generates the pressure loads that are input to the CSD simulation, which generates geometrical displacements that are returned to the CFD simulation. As the CFD geometry changes, so will the fluid solution and pressure distribution. Since the pressure distribution history for the two-way

coupling could be significantly different than that computed using the one-way coupling, the stress within the structure would also be different but, potentially, more accurate. The challenge with FSI is the transfer of loads and displacements between the CFD and CSD codes. A new method was developed to transfer the loads and displacements. It is the scope of this thesis to explain this new method and provide validation for it.

CHAPTER II

BACKGROUND

A fluid-structure interaction simulation involves the solution of three coupled systems of partial differential equations: the fluid dynamics equations (e.g. Navier-Stokes), a structural dynamic equation, and a grid deformation equation [1]. While the fluid dynamics equations and the structural dynamic equations are not directly coupled, they are coupled through the grid deformation equation. Due to different numerical techniques employed to solve each of these equation sets, along with different mesh discretization requirements for the fluid and the structure equations, it becomes counterproductive to try to solve the equations simultaneously. It is advantageous to loosely couple the equations: 1) solve Navier-Stokes; 2) transfer the loads to the CSD grid; 3) solve structure dynamics; 4) update the CFD grid based on the predicted deformation of the structure. How the loads are transferred to the CSD grid and how to update the CFD grid are the important parts that need to be determined.

One common approach is to use an interpolation scheme in which the CFD pressures are transformed into CSD stresses by pairing each structural Gauss point to a fluid element. Even though this method can give reliable answers, it does have conservation problems [1]. If the CFD and CSD simulations use the same discretization, i.e. a point-to-point matching, then conservation would be guaranteed. Using this point-

to-point matching, Farhat et al. [1] proposed a conservative transfer by interpolating a fluid point to a location on the structural surface.

While node-projection schemes are conservative, inaccurate load transfers can lead to localized errors [2] [3]. For instance, the load transfer will be different depending on the exact location on the structural element the fluid point is projected. Therefore, the solution is not unique. Jaiman, et al. [2] developed a common refinement method to alleviate this problem. Sub-elements are generated based on the intersections between the two grids, and the traction vectors are calculated based on these sub-elements. It was pointed out that "for complex non-convex geometries, it may become difficult to select a reasonable mapping for all points on the reference surface to the common-refinement discretizations."

As will be seen, the method presented here is a conservative, robust technique that gives unique solutions to FSI problems.

CHAPTER III

NEW NODE PROJECTION METHOD

3.1 Elemental Stiffness Matrix

To be able to transfer the loads from a CFD node to a structural surface element, whether or not the CFD node lies on the CSD surface, tetrahedral or pyramidal elements are used to connect them. The advantage of this approach is that it allows for the load transfer to be statically equivalent, even in a statically indeterminate system by solving a finite element equation

$$[K]\{\delta\} = \{F\} \tag{3.1}$$

where [K] is the stiffness matrix, {δ} is the displacement vector, and {F} is the load vector. By applying a zero-displacement boundary condition to the bottom of the element (the solid surface), the unknown degrees of freedom that correspond to the CFD node can be determined by solving a reduced system of equations.

$$\{\delta_R\} = [K_R]^{-1}[F_R] \tag{3.2}$$

Equation 3.1 can be solved again with the newly determined CFD node displacements to get the loads on the CSD nodes.

The stiffness matrix [K] is determined by

$$[K] = \int [B]^T [D][B]\, d\Omega \tag{3.3}$$

where [D] is the elastic matrix, [B] is the strain-displacement matrix, and Ω is the domain of integration, which is a volume in this three-dimensional case. The elastic matrix is composed of the elastic modulus, E, and Poisson's ratio, ν. The elastic matrix relates the strain to the stress by $\sigma = [D]\varepsilon$. Since the elements that are being used are perfectly rigid, ν is set equal to zero. The strain-displacement matrix is defined by the Cartesian derivative of the shape functions, $[B] = \frac{dN}{dx}$. The shape functions will be elaborated on later.

As stated previously, once the unknown degrees of freedom are known, the unknown reaction forces on the structural nodes can be computed by substituting Eq. 3.2 into Eq. 3.1 giving

$$\{F\} = [K]\{\delta\} = [K][K_R]^{-1}\{F_R\}. \tag{3.4}$$

Even though an elastic matrix was used in the construction of the stiffness matrix, the product $[K][K_R]^{-1}$ is independent of the elastic modulus, but still dependent on the Poisson's ratio. However, since ν is zero, the force transfer is independent of the material properties chosen for the element.

3.1.1 Tetrahedral Elements

Tetrahedral elements are the simplest of the three-dimensional elements - three nodes being the minimum number needed to form a plane, and four to determine a solid. As such, the formulation of the stiffness matrix can be found in any finite element analysis textbook [4] [5].

The elastic matrix for a tetrahedral element is given by

$$[D] = \begin{bmatrix} c_1 & c_2 & c_2 & 0 & 0 & 0 \\ c_2 & c_1 & c_2 & 0 & 0 & 0 \\ c_2 & c_2 & c_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & c_3 \end{bmatrix} \tag{3.5}$$

where

$$c_1 = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} = 1, \tag{3.6}$$

$$c_2 = \frac{E\nu}{(1+\nu)(1-2\nu)} = 0, \tag{3.7}$$

$$c_3 = \frac{E}{2(1+\nu)} = \frac{1}{2}. \tag{3.8}$$

Recall that the force transfer is independent of the element's material properties, so the elastic modulus is arbitrarily set to one for simplicity. Also, as noted above, Poisson's ratio is zero for a rigid element.

The strain-displacement matrix is a bit more complicated to determine. The nodes must be ordered. Figure 3.1 shows how this is done. Starting with a node at the base (surface structural element), assign numbers 1, 2, and 3 to the nodes going in a counter-clockwise fashion, which is consistent with the right hand rule with an outward normal pointing towards the apex. The apex of the tetrahedral (CFD node) is assigned 4.

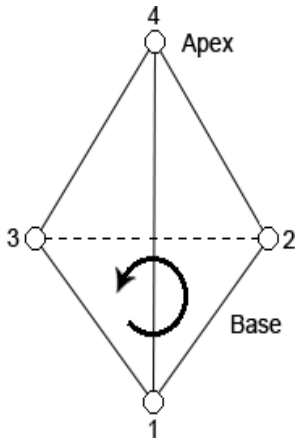

Figure 3.1 Tetrahedral element node ordering

A matrix is then formed by the distances from nodes 1, 2, and 3 to node 4 in the three Cartesian coordinate directions

$$[J] = \begin{bmatrix} x_{14} & y_{14} & z_{14} \\ x_{24} & y_{24} & z_{24} \\ x_{34} & y_{34} & z_{34} \end{bmatrix} \tag{3.9}$$

where

$$x_{i4} = x_i - x_4,$$

$$y_{i4} = y_i - y_4,$$

$$z_{i4} = z_i - z_4.$$

The determinant of [J] is calculated and is used in forming a new matrix, which represents the Cartesian derivatives of the natural coordinates and is used in the calculation of the strain-displacement matrix.

$$[a] = \frac{1}{\det[J]} \begin{bmatrix} y_{24} * z_{34} - z_{24} * y_{34} & y_{34} * z_{14} - z_{34} * y_{14} & y_{14} * z_{24} - z_{14} * y_{24} \\ z_{24} * x_{34} - x_{24} * z_{34} & z_{34} * x_{14} - x_{34} * z_{14} & z_{14} * x_{24} - x_{14} * z_{24} \\ x_{24} * y_{34} - y_{24} * x_{34} & x_{34} * y_{14} - y_{34} * x_{14} & x_{14} * y_{24} - y_{14} * x_{24} \end{bmatrix} \tag{3.10}$$

The final strain-displacement matrix, in terms of Eq. 3.10, is given by

$$[B] = \begin{bmatrix} a_{11} & 0 & 0 & a_{12} & 0 & 0 & a_{13} & 0 & 0 & b_1 & 0 & 0 \\ 0 & a_{21} & 0 & 0 & a_{22} & 0 & 0 & a_{23} & 0 & 0 & b_2 & 0 \\ 0 & 0 & a_{31} & 0 & 0 & a_{32} & 0 & 0 & a_{33} & 0 & 0 & b_3 \\ 0 & a_{31} & a_{21} & 0 & a_{32} & a_{22} & 0 & a_{33} & a_{23} & 0 & b_3 & b_2 \\ a_{31} & 0 & a_{11} & a_{32} & 0 & a_{12} & a_{33} & 0 & a_{13} & b_3 & 0 & b_1 \\ a_{21} & a_{11} & 0 & a_{22} & a_{12} & 0 & a_{23} & a_{13} & 0 & b_2 & b_1 & 0 \end{bmatrix} \quad (3.11)$$

where

$$b_i = -\sum_j a_{ij}.$$

With [B] and [D] formulated, Eq. 3.3 can now be used to calculated the elemental stiffness matrix [K].

3.1.2 Pyramid Elements

The pyramidal element construction is more involved than that for the tetrahedral element. One of the main differences is that the pyramid is constructed so that the CFD node lies within the pyramid instead of at the apex, as it did for the tetrahedral element. Given the base of the pyramid (the quadrilateral surface element), the first step to establish the pyramid apex is to find the center of the base. This is done by taking the average of the node locations. The apex is determined by a weighted normal vector of the CSD element added to the CFD node. A vector to define the apex is determined by

$$\boldsymbol{n} = \frac{|\boldsymbol{V}_{13}|}{2} \frac{(\boldsymbol{V}_{12} \times \boldsymbol{V}_{13}) + (\boldsymbol{V}_{13} \times \boldsymbol{V}_{14})}{2} \quad (3.12)$$

A direction vector is calculated from node 1 to the other three nodes on the base (like for the tetrahedral, the nodes on the base are numbered 1-4 in counter-clockwise order, and the apex of the pyramid is numbered node 5). The cross-product of $\mathbf{V}_{12}$ and $\mathbf{V}_{13}$ is taken to get the first surface normal. The same procedure is used for $\mathbf{V}_{13}$ and $\mathbf{V}_{14}$

to compute a second surface normal ($\mathbf{V}_{ij}$ is the vector pointing from node i to node j). These two normal vectors are averaged together to get an averaged unit outward normal. The apex of the pyramid is then given by a scaling term, half of the magnitude of the diagonal vector $\mathbf{V}_{13}$ multiplied by the averaged unit outward normal, added to the coordinates of the CFD node. This ensures that the CFD node lies within the pyramid.

The shape functions for the pyramid also need to be determined. A coordinate transformation is performed from Cartesian coordinates to a natural coordinate system. The natural coordinates (ζ, η, and μ) range from -1 to 1 as can be seen in Fig. 3.2. The shape functions that describe the pyramid can be expressed as

$$N_i = \frac{1}{8}(1 + \xi\xi_i)(1 + \eta\eta_i)(1 + \mu\mu_i) \tag{3.13}$$

where (ζ_i, η_i, μ_i) are the natural coordinates of the i^{th} node. Substituting the values of the pyramid nodes gives the following five shape functions

$$N_i = \begin{matrix} \frac{1}{8}(1-\xi)(1-\eta)(1-\mu) \\ \frac{1}{8}(1+\xi)(1-\eta)(1-\mu) \\ \frac{1}{8}(1+\xi)(1+\eta)(1-\mu). \\ \frac{1}{8}(1-\xi)(1+\eta)(1-\mu) \\ \frac{1}{2}(1+\mu) \end{matrix} \tag{3.14}$$

The derivatives of these five shape functions with respect to each of the natural coordinates are needed as well.

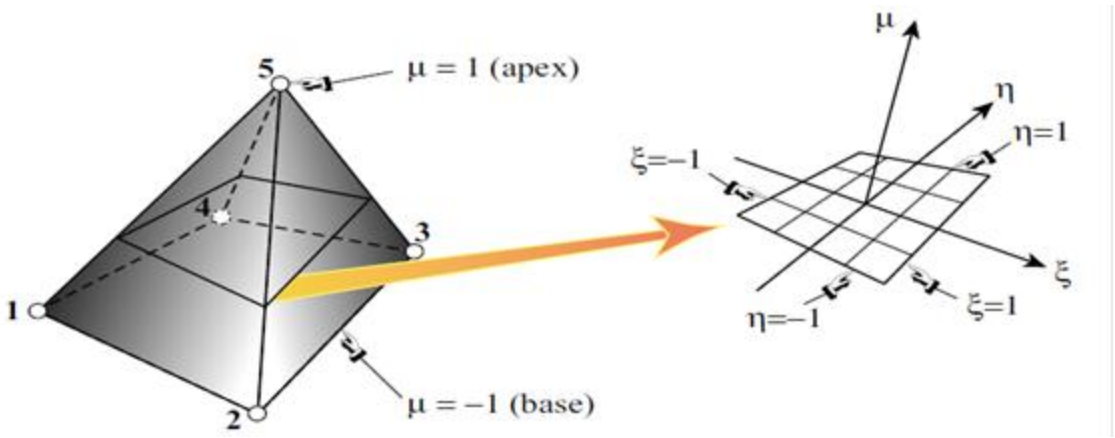

Figure 3.2 Natural coordinates of a pyramidal element [6].

The natural coordinates need to be determined for the CFD node within the pyramid. This is accomplished by guessing the coordinates (i.e. setting ζ, η, and μ equal to 0, 0 and -1 respectively), then iterating to converge to the actual values. Using the guessed values of the natural coordinates, Eq. 3.14 and derivatives are calculated. The guessed values for the natural coordinates are transformed into the Cartesian system using the shape functions. The Jacobian, $[J] = \frac{d(x,y,z)}{d(\xi,\eta,\mu)}$, is also calculated by

$$J_{kl} = \sum_i \frac{dN_i}{dl} k_i; \quad k = x, y, z \quad l = \xi, \eta, \mu \tag{3.15}$$

The Jacobian is multiplied by an update vector, {d}, to give the guessed Cartesian coordinates for the CFD node minus the actual position of the node, which will be referred to as the right-hand side (RHS). This gives

$$[J]\{d\} = RHS\ . \tag{3.16}$$

The update vector is used to update the guessed values of the natural coordinates for the CFD node. Upon convergence, the RHS will equal to zero. However, if the CFD

node and, thus, the pyramid apex, is on the opposite side of the structural element from the direction of the unit normal, performing the above routine will diverge. If this is the case, the apex is recalculated to be on the other side and the routine is repeated. It should be noted that in the current code, updating is repeated for ten iterations. Logic could be added to detect convergence to increase speed, but this step takes so little time, especially when compared to the actual dynamics steps, that it is not needed.

With the natural coordinates for the CFD node now known, the stiffness matrix can be formed. The coordinates are used to calculate the Jacobian, and, thus, the Cartesian derivatives of the shape functions, using Eq. 3.15. This is actually what is done to get the strain-displacement matrix for the tetrahedral element. With [B] and the simplified [E], numerical integration is used to solve for [k]. Gaussian quadrature is used for the integration. Since the pyramid is a linear element, only one quadrature point is required for the integration.

3.2 Routine

A routine [7] was written to couple LS-DYNA [8] [9], a commercial CSD software, with Loci/BLAST [10], a research code. The routine is divided into two parts: the initialization and the interaction. The initialization is the first thing the code does. It is only done once, and it sets up everything that is needed for the interaction part.

3.2.1 Initialization

The number and the locations of the CFD nodes are read in and stored. The same is done for the CSD nodes, but the connectivity is also read in so that the surface elements can be determined. The LS-DYNA connectivity file always lists four points in

the element. If the third and fourth points are the same, then the element is a triangle. This is how the code distinguishes between the two different element types.

A k-d tree routine is used to determine which CSD element is closest to each CFD node. A vector is drawn from the CFD node to the centroid of every CSD element. The centroid is determined by simply taking the average of locations of each CSD node on that element. The tree chooses the shortest vector and, thus, the closest element. There is a chance that if there is a large element next to a small element, that the CFD node could be linked to the smaller when it would be better to link it to the larger, but that situation is bad practice in finite element analysis, and should be avoided. However, if a CFD node is linked to a "wrong" CSD element, the routine will still work but the force transfer is no longer unique.

The CFD nodes and their corresponding CSD surfaces are connected by a fictional pyramidal or tetrahedral element. The stiffness matrices for the elements are formulated as described above.

3.2.2 Interaction

The interaction part is the implementation of the FSI algorithm that actually does the calculation and transfer of loads and displacements between LS-DYNA and Loci/BLAST.

Loads are transferred from LOCI/Blast to LS-DYNA. With the given loads on the CFD nodes, Eq. 3.1 is solved yielding the distributed loads on the CSD nodes. In general, multiple CFD nodes will be mapped to any given CSD surface element, so the distributed load contribution from each CFD node is added at on each CSD node. If a CSD element does not have any CFD nodes mapped to it, then the resulting forces on

those CSD nodes are set to be zero. These loads are then read into LS-DYNA by the User Defined Loading [11] routine.

The LS-DYNA execution gives an updated surface geometry. Equation 3.1 is then solved again, except this time the displacements of the CSD nodes are known and the CFD node displacement is found. The displacements are added to the original locations to obtain the updated CFD locations for Loci/BLAST to use. The nodal velocities and accelerations are also transferred to Loci/BLAST so that it can properly calculate the loads.

This part of the routine is repeated until LS-DYNA is finished running. If LS-DYNA runs longer than Loci/BLAST does, then the transferred forces are zeroed out, leaving LS-DYNA to run by itself with the residual motion of the CSD structural elements.

CHAPTER IV

VALIDATION

To make sure that the new node projection method is properly validated, it should provide reasonable answers compared to an experiment, the load and displacement transfers should be work conservative, and the method should be robust enough to work on complicated geometry.

4.1 Experimental Comparison

The Defense Research and Development Canada (DRDC) conducted experimental tests in May of 2000 to study the effects of mine blasts on armor plates [12]. The simplicity of the model and the availability of the experimental results were the reasons for choosing this case as a validation case.

A 6 ft (182.88 cm) square aluminum plate was suspended 16 in (40.64 cm) above the ground by corner supports. A box support rested on top of the plate to help hold it in place, leaving a 4 ft (121.92cm) square opening over the middle of the plate. An additional 10,620 kg mass rested on top of the box frame to resist movement during the mine detonation. The experimental set-up diagram can be seen in Fig. 4.1.

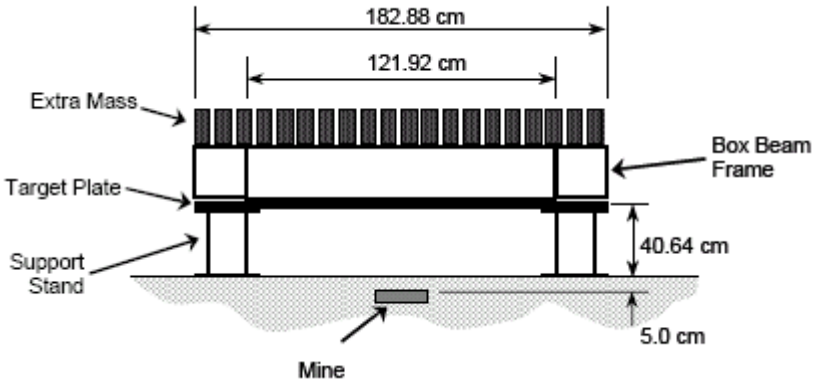

Figure 4.1 DRDC plate experimental set-up [13].

A six kilogram cylindrical C-4 charge was used. The mine was buried directly underneath the center of the plate, with a 5 cm depth of burial, measured to the top of the mine. The thickness to diameter ratio of the charge was 0.3. The density of the soil was assumed to be 2301 kg/m^3. These properties are summarized in Table 4.1.

Table 4.1 Mine and soil conditions [13]

Property	Value
Mine Mass	6 kg
Mine Diameter	25.4 cm
Mine Thickness	7.62 cm
Depth of Burial	8.81 cm
Soil Density	2301 kg/m^3

To speed up the run time of the simulation, Loci/BLAST was only coupled to LS-DYNA after blast from the mine almost reached the structure. Until this time, the blast simulation was treated as axisymmetric, and used a uniform, 0.5 mm grid. This pre-interaction simulation ran for about 0.05 ms.

The result from the axisymmetric run was interpolated onto a more coarse grid (two discretizations were used) to continue the run with the coupling turned on. The coarser grid resolution was 8 mm x 18 mm x 18 mm, with the 8 mm being along a ground-to-plate vector. The finer grid is one-half of coarser grid's spacing in each direction: 4 mm x 9 mm x 9 mm.

Symmetry was also exploited in the coupled part of the simulation; however, due to Loci's grid deformation [14], only one plane of symmetry can be used even though the plate geometry has two. The plate was made up of 3200, 22.86 mm square elements. The frame holding the plate is made out of rigid elements so the discretization is not important. The model can be seen in Fig. 4.2. This step of the simulation ran coupled for 2 ms (Fig. 4.3), and LS-DYNA continued to run to generate the response of the plate up to 5 ms. However, Loci/BLAST crashed and terminated prematurely on the finer mesh; the continued response start at about 1.5 ms. The finer mesh got tangled during the mesh deformation step of the algorithm. The mesh deformation updates are based on the LS-DYNA time step. On a finer mesh, each grid point is moving farther, relative to the grid size. If the LS-DYNA had more time steps, then perhaps the finer mesh would not have crashed, but further investigation would be required to find the cause.

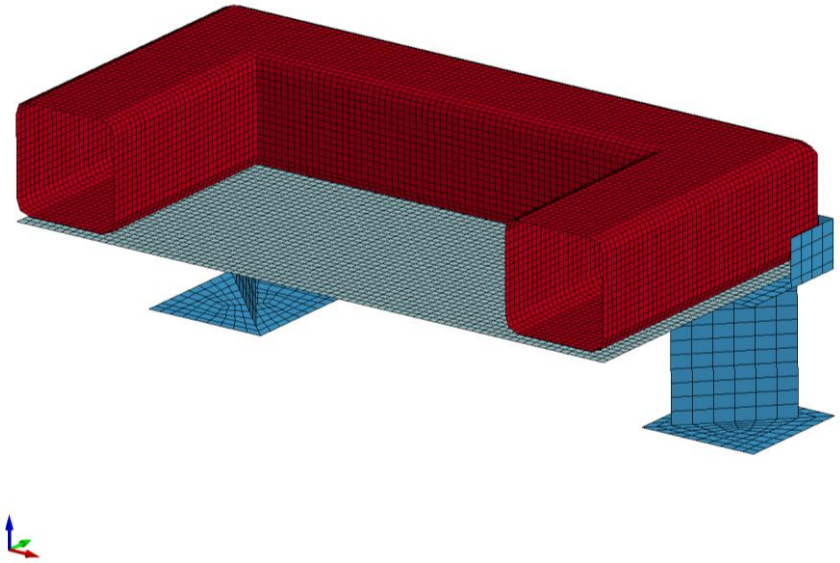

Figure 4.2 DRDC plate LS-DYNA original model

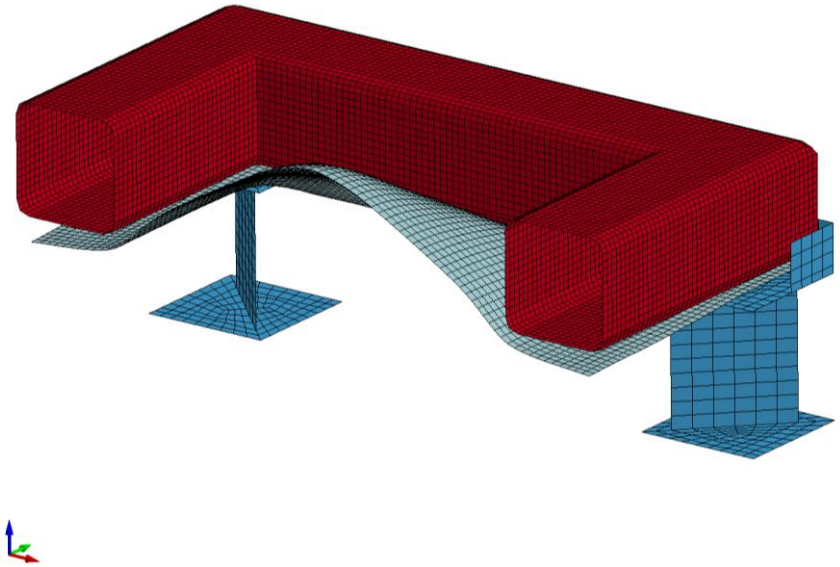

Figure 4.3 Deformed DRDC plate (2 ms).

During the DRDC experiment, probes were placed on the plate at key locations: at the center and different locations down each of the center lines toward the edges of the plate. These probes provided displacement data at a time of 5 ms. The mesh refinements

of Loci/BLAST – LS-DYNA coupling were compared to experimental data (Fig. 4.4), as well as the TAMCO model discussed in Williams [13]. It can be seen that the finer mesh did a better job predicting the deformation. And, while the TAMCO model predicted the upper bound of the deformation data, the Loci/BLAST – LS-DYNA predicted the lower bound fairly well. It was also noted that during the mesh refinement simulations, the plate lifted slightly off of the frame (Fig. 4.5), so the plate was constrained to the frame in the LS-DYNA model and another run was performed on the coarser mesh. The work that went into lifting the plate off of the frame in the original run produced more deformation in the plate instead, which can be seen in Fig. 4.6. One explanation for why the simulation still did not correspond precisely with the experimental data is that only the soil density was given. While soil density plays a major role in the plate deformation, so does the moisture in the soil. The soil moisture affects the cohesion of the soil, which, in turn, factors into the impulse the soil imparts on the plate. Since that data was not given, the soil was assumed to be dry.

Data from the DRDC experiment was also provided to the displacement history for a location 30.5 cm from the plate center along a centerline seen in Fig. 4.7. There was no displacement for the Loci/BLAST – LS-DYNA runs for a small section of time, which was removed so that the displacement starts at t=0. This was done because of the time for the blast wave to reach the plate and because the TAMCO model assumes that all the loading occurs over the plate at the same time, so it predicts displacement at t=0. The coupled runs follow the experimental values well for all of the cases. The runs match the TAMCO model well, until about the 2 ms mark. The deviation could be due to the fact that Loci/BLAST only ran for 2 ms, so after that point, the plate starts rebound back

while the TAMCO model is still applying the forces to keep the plate deformed. This also applies to why the TAMCO model predicted the upper bound of the experimental data and Loci/BLAST – LS-DYNA coupling predicted the lower bound.

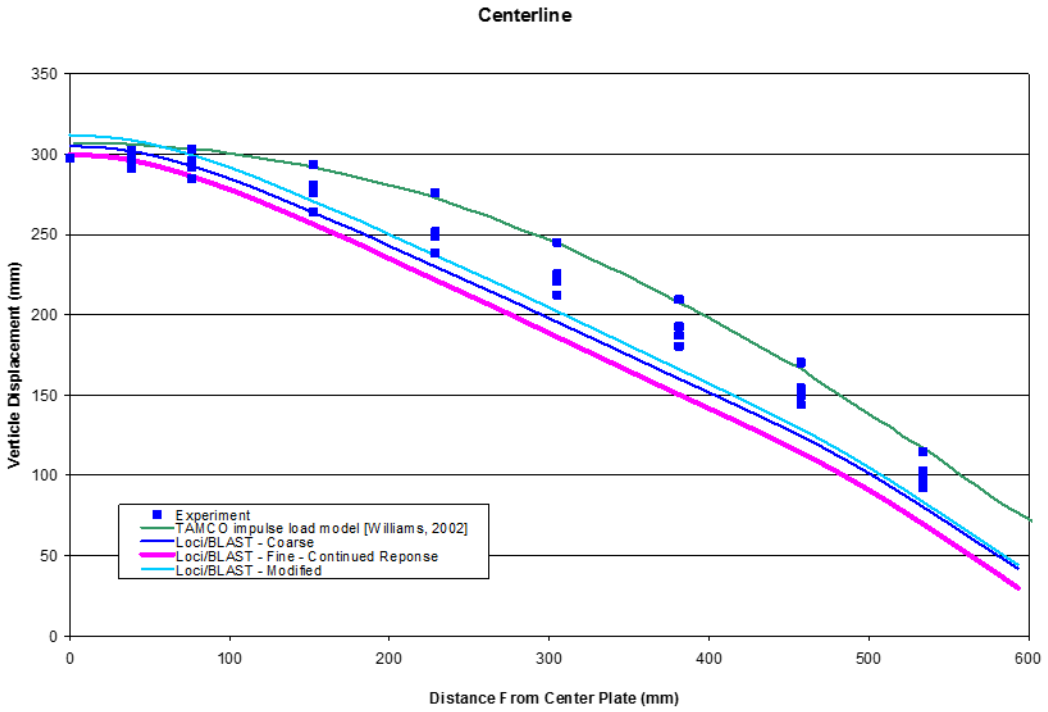

Figure 4.4 DRDC plate centerline displacement

As one more point of comparison, Williams shows a figure of a model that correlates the soil density to the deflection of the center of the plate. Figure 4.8 shows this graph along with two data points from the Loci/BLAST – LS-DYNA: one using dry sand and one using the soil2300. Both of these points lie near the model and provide evidence that the coupled codes follow this model as well.

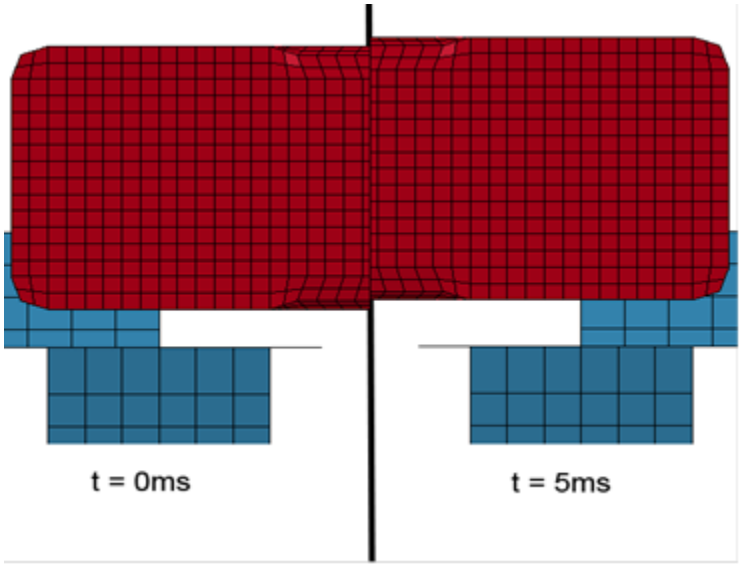

Figure 4.5 Frame deflection

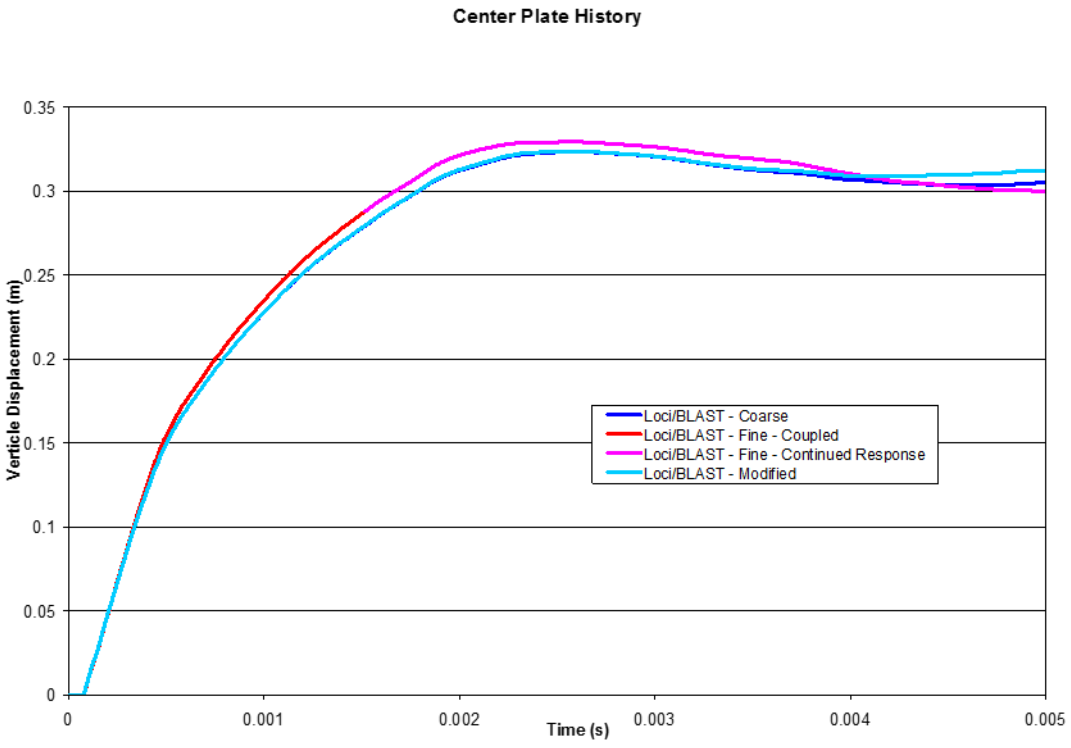

Figure 4.6 DRDC plate center time history

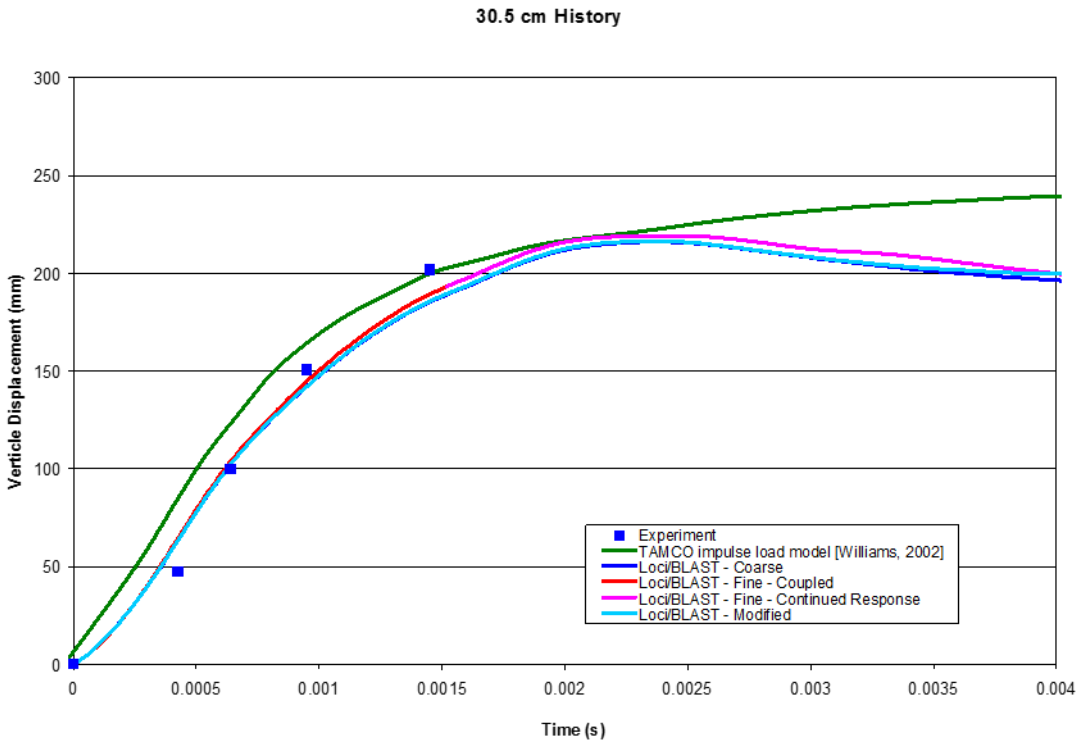

Figure 4.7 DRDC plate time history

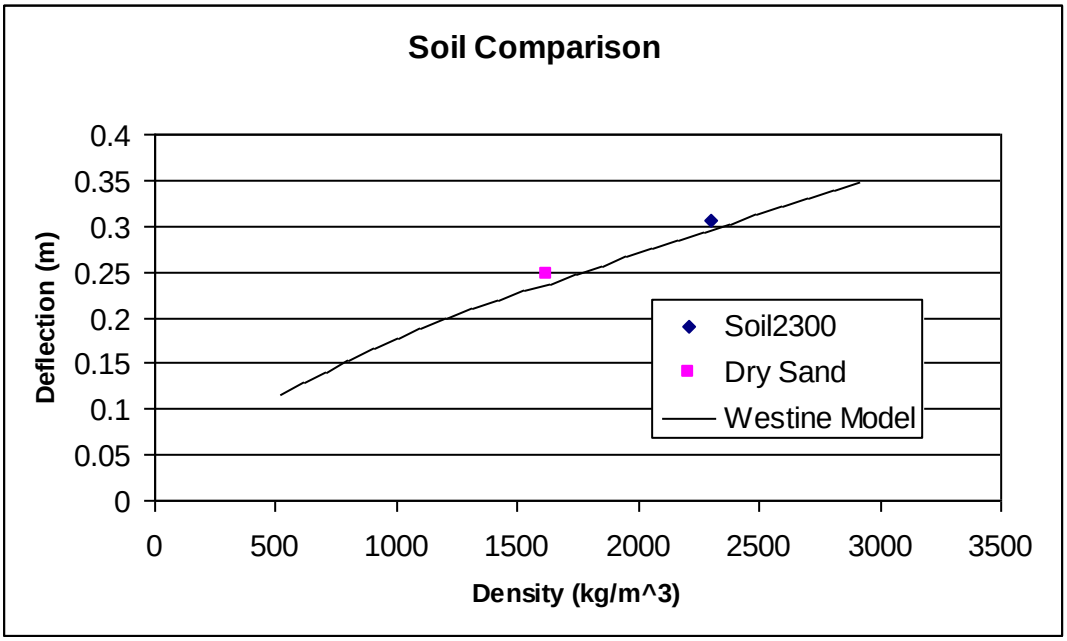

Figure 4.8 Density – deflection correlation

4.2 Work Conservation

As was pointed out earlier, Farhat, et al. [1] showed that, just because a technique provides reasonable, if not accurate, results, it may not actually conserve the work. Therefore, it is not enough to just show this new node-projection scheme provides reliable answers to problems, but conservation needs to be proven, even though Jaiman, et al. [2] showed that node-projection schemes are inherently conservative.

To prove that the method does provide conservation of work, the work done on the CFD mesh and the CSD mesh are compared. The FORTRAN fluid-structure interaction (FSI) routine did not explicitly calculate the work done on each mesh. However, the CFD nodal forces are known from Loci/BLAST, which are transformed into the CSD nodal forces, and the displacements from LS-DYNA are used to update the node locations on the CFD mesh. For each time step, the work on each mesh can be calculated by summing the dot products of the force and the displacement vectors for each node. A routine was added to the code to do just that and keep track of the total work done on each mesh with respect to time. At each time step, the routine calculates the work on each mesh by

$$W = \sum_i \{F\}_i \cdot \{\delta\}_{i,} \qquad i = 1, \dots, N \tag{4.1}$$

where N is the total number of nodes in that particular mesh. The displacement for each node is calculated by subtracting the node location at the previous time step from the location at the current time step,

$$\{\delta\}_i = \{x\}_i^t - \{x\}_i^{t-1} . \tag{4.2}$$

4.2.1 Static Case

The first case chosen to show the conservation of work is a static case of a horizontal pipe holding water. For a static case, the forces on the CFD nodes are not going to change with time, so there is no need to run Loci/BLAST. This case requires only a one-way coupling. The communication with Loci/BLAST was turned off in the FSI subroutine. Normally, Loci/BLAST determines the CFD nodal forces, but since it is turned off in the one-way coupling case, files with the original CFD mesh and forces were and read in by the FSI routine. The routine will still give an updated CFD mesh location that is used in the work calculation.

A 1 m long section of a 2 m diameter pipe with 1.5 cm wall thickness was modeled. Since the test case was to be a simple case, the material properties for steel were used, and the problem was assumed to be perfectly elastic. These assumptions, along with the geometry of the pipe section, more than likely, will not generate a valid physical result, but since the aim of this test case is to show the conservative nature of this new node-projection scheme, the "correctness" of the CSD assumptions is of no consequence.

The CSD model was modeled very coarsely (again, the accuracy of the actual solution is not needed for this part) with ten grid point around the circumference of the pipe and four grid points down the length for forty points total. Since the CFD nodal forces needed to be manually specified, the number of CFD nodes was relatively small; the CFD mesh was twice as refined with twenty points around the circumference and eight nodes down the length.

As said before, the pipe was simulated to be holding water so that the load would not be evenly distributed around the pipe but be more concentrated is the bottom and less on the top due to hydrostatic pressure. A hydrostatic pressure was calculated along the depth of the pipe, with the top of the pipe having 0 m depth, and the bottom a 2 m depth. Just so that there would not be any location in the pipe that had a zero pressure, an additional 500 Pa was added to the hydrostatic pressure. Knowing the CFD element size, the pressure at each of the CFD nodes was converted into Cartesian force components to be read by the FSI routine.

Figure 4.9 shows the cross-section meshes of the pipe, both before and after the loading. It can be seen that due to the loading, the pipe in slightly compressed along the y-axis and wider in the z-axis. But, perhaps, a more important observation is made in that the CFD and the CSD nodes that are coincident in the original mesh remain coincident after the deformation. If work is conserved between the two meshes, then coincident points should stay coincident after deformation.

The work done by the fluid on the structure for the entire pipe in this case is about 800 J. The error between the CFD work and the CSD work is about 8.07e-7%. Although this error is small enough to imply that the routine is conservative, it would be more convincing with a dynamic case with multiple time steps.

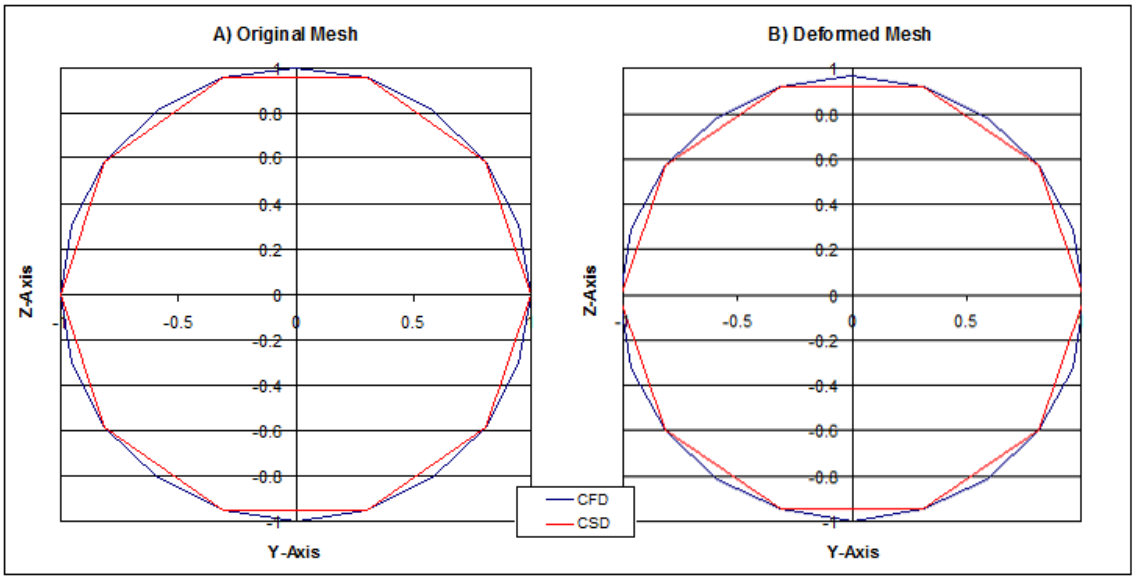

Figure 4.9 Static pipe cross-sectional meshes

4.2.2 Dynamic Case

A shock tube case was chosen to be the dynamic test case. Again, as with the static case, it is preferred that the pressure distribution not be axisymmetric. To achieve the desired nonuniformity, the shock tube slightly curved. Oblique shocks and expansion waves are formed to turn the flow around this slight curve resulting in a non-uniform pressure.

The shock tube is 0.25 m in radius and has 2 cm shell thickness. The first 0.5 m of the tube is straight to allow the shock to develop before it reaches the bent section of the tube. The curved portion of the tube is 5.5 m long and bends through a 10 degree turn. Perfectly elastic steel properties were used for the material, except that the elastic modulus used was 200 MPa. The three orders of magnitude reduction in the modulus allows for a larger displacement and thus a larger work.

Figure 4.10 shows the discretization of the original CSD mesh. Sixteen elements were used to discretize the radially around the tube, while sixty were used in the lateral direction for a total of 960 CSD nodes. The CFD surface mesh was slightly more refined with about twenty-two surface elements around the circumference (the grid is unstructured so this number could vary at different locations along the tube) and about seventy elements laterally. This resulted in the CFD mesh having slightly more than double the number of surface nodes of the CSD mesh. The CFD mesh was initially finer, but it caused the solution to fail. The grid deformation technique used within Loci/BLAST generated a negative area if the elements are too small, possibly due to grid tangling explained in Section 4.1, but further investigation may be necessary. However, the coarser CFD mesh can be used to prove conservation of work because only a few of the CFD nodes and the CSD nodes are coincident

.

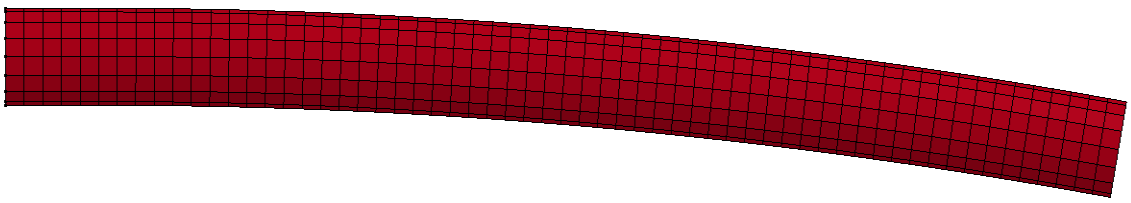

Figure 4.10 Undeformed shock tube CSD mesh.

The initial air conditions in the tube, as well as the outlet conditions were: 1 atm pressure, 300 K temperature, and 0 m/s velocity. The inlet conditions, including the first 0.1 m of the tube, were: 8 atm pressure, 685.7 K temperature, and 642.5 m/s velocity in the direction down the tube. The initial pressure distribution in the shock tube can be seen in Fig. 4.11. These conditions introduced a shock that will travel all the way through the tube in about 6.5 ms, and cause the air behind the shock to be supersonic. Even though it only takes 6.5 ms for the shock to travel through tube, the simulation ran for 8 ms to allow for further deformation of the pipe.

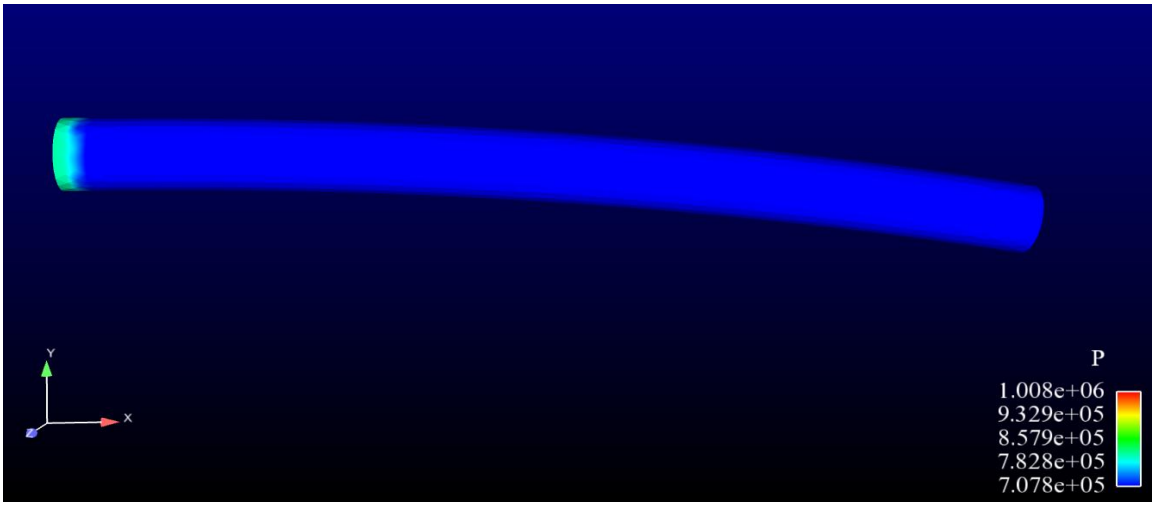

Figure 4.11 Pressure state of shock tube just after initialization.

Figure 4.12 shows the time history of the total work done on both meshes. Both curves lay on top one another all the way from 0 ms to about 7.8 ms. (Note: Due to the way Loci/BLAST and LS-Dyna were coupled, the work history stopped at 7.83 ms even though the simulation ran for the full 8 ms.) To get a better grasp on the work conservation, the time history of the relative work error was also plotted as seen in Fig.

4.13. The maximum error in this dynamic case was about 2.07e-7%, which is the same order of magnitude of error as the static case. While the error seems to converge before a small spike around 7 ms (which may have something to do the shock reflecting off the open end of the tube), the "converged" error still is not much less than the maximum error. This small error suggests that this node-projection scheme is conservative.

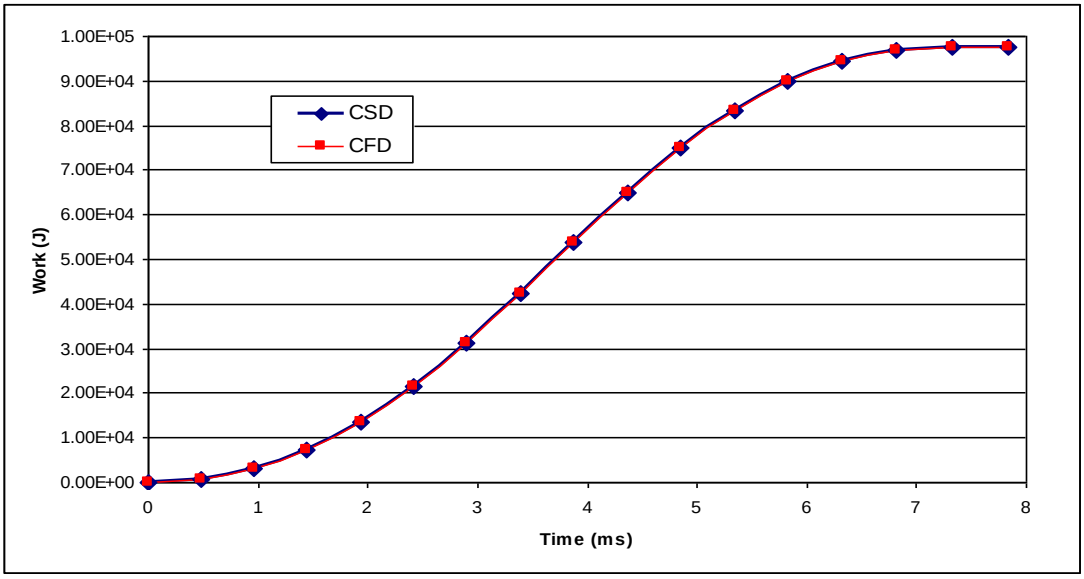

Figure 4.12 Work history on both meshes.

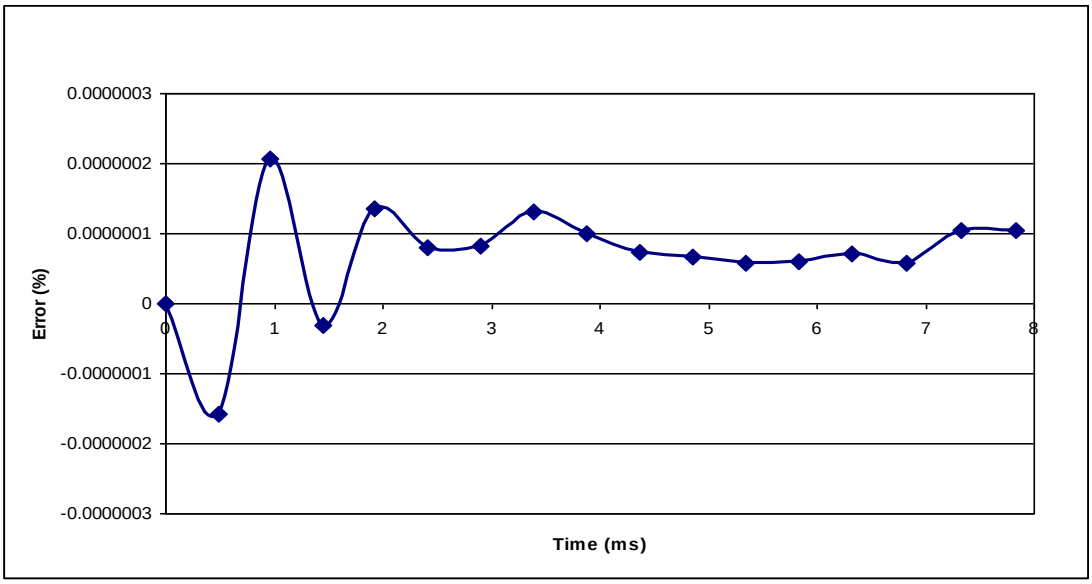

Figure 4.13 Relative percent error history.

4.3 Grid Refinement Study

Just because the present method can give valid results and is conservative, it does not necessarily mean that the method is useful if it can only be used with simple geometries. To be useful, this method has to be robust enough to work on a variety of different, and possibly complex, geometries, so a grid refinement study was conducted on a more complex geometry. The case that was chosen was a buried explosive charge detonating underneath a structure, similar to the DRDC plate used in the validation, except the structure was a generic vehicle hull instead of a flat plate.

The LS-Dyna model used for the vehicle hull, Fig. 4.14, was supplied by TARDEC. While it is not a perfect to test the robustness of the transfer algorithm (for example, it does not have any concave sections), it does have differently sized elements, both triangular and quad elements, and sharp corners. A few simplifications to the

geometry were made for the CFD model, for example, removing the door hinges, as well as, only having a grid exterior to the hull.

This was a purely simulated case with no experimental data with which to compare, so it was assumed that a 6 kg cylindrical charge of C4 was buried 2 in underneath dry sand. The simulation was carried out for 5 ms. Since there was no experimental data, a grid refinement was conducted using local body CFD resolution of 5 cm, 2.5 cm and 1.25 cm. Considering the computations took about 15 wall hours for the coarsest simulation to complete on 192 processors, the refined meshes were locally refined around the hull while coarsening in the far field so that the number of elements remained essentially constant.

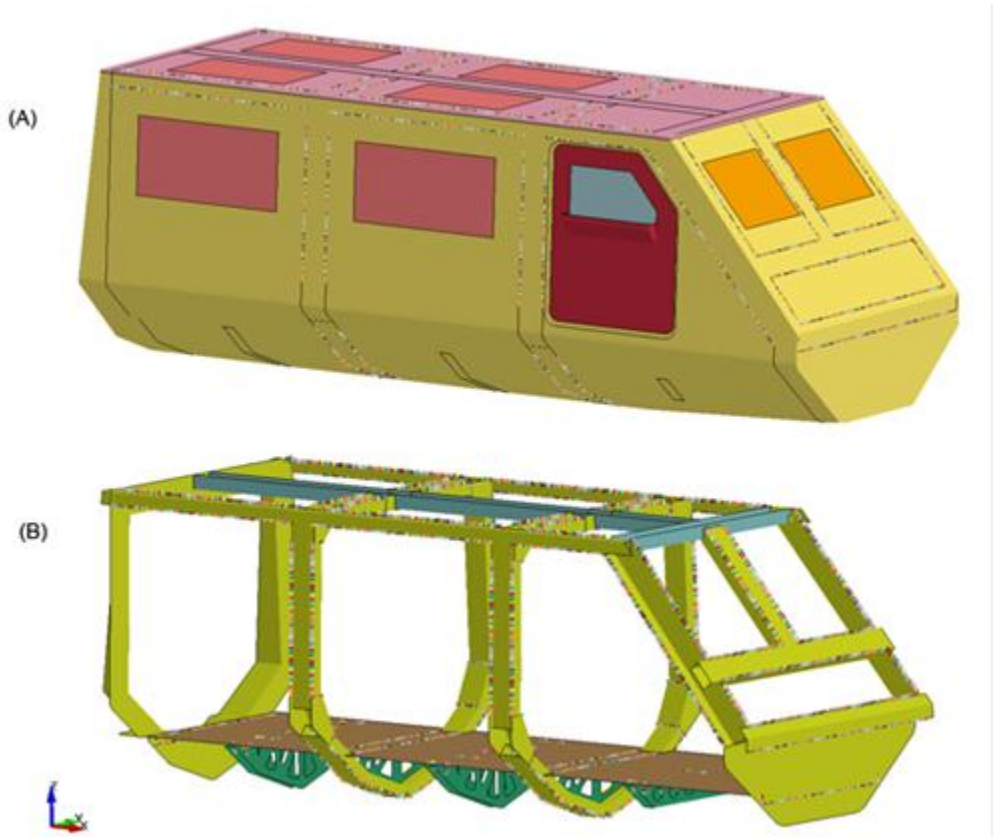

Figure 4.14 LS-Dyna generic hull model. A) outer shell and B) inner frame.

Two points (Fig. 4.15) were chosen to compare the history of the vertical displacement among the mesh refinements. The first node chosen (node 645358) was on the skin on the bottom of the hull, right above where the explosive was buried. The second node (node 612928) was on the frame on the bottom of the hull slightly behind the charge.

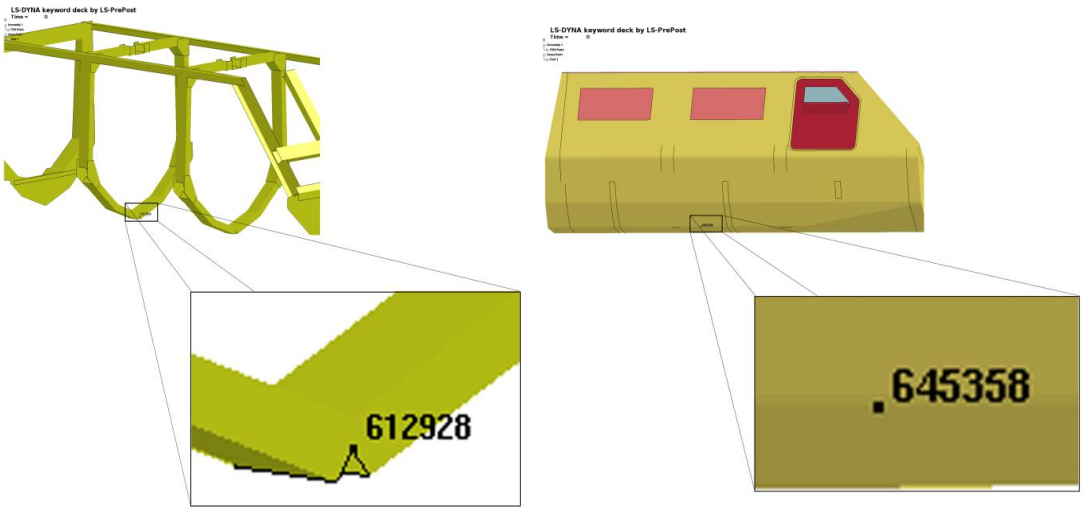

Figure 4.15 Locations of chosen CSD nodes.

The vertical node time histories for the selected skin node and frame node are shown in Figs. 4.16 and 4.17, respectively. In both cases, the finer mesh soulutions produced more displacement in the nodes. However, the frame node seemed to converge better with the mesh refinement. It should be noted that Loci/BLAST failed for the finest of the meshes around 2 ms, perhaps for the reasons previously discussed. After the fluid code failed, LS-DYNA continued running for the remainder of the 5 ms. The decoupled responses had the same shapes as the coupled responses of the coarser meshes. It could be that most of the explosive energy is imparted in the first 2 ms of the explosion (at least to the bottom of the hull where these two points were located) and the material response plays a bigger role for the rest of the time. Also, it could be that as the mesh becomes finer, the more concentrated the blast and, thus, the imparted pressure becomes localized. That would explain why the frame node, which is behind the blast, converges faster than the skin node, which is directly above the blast.

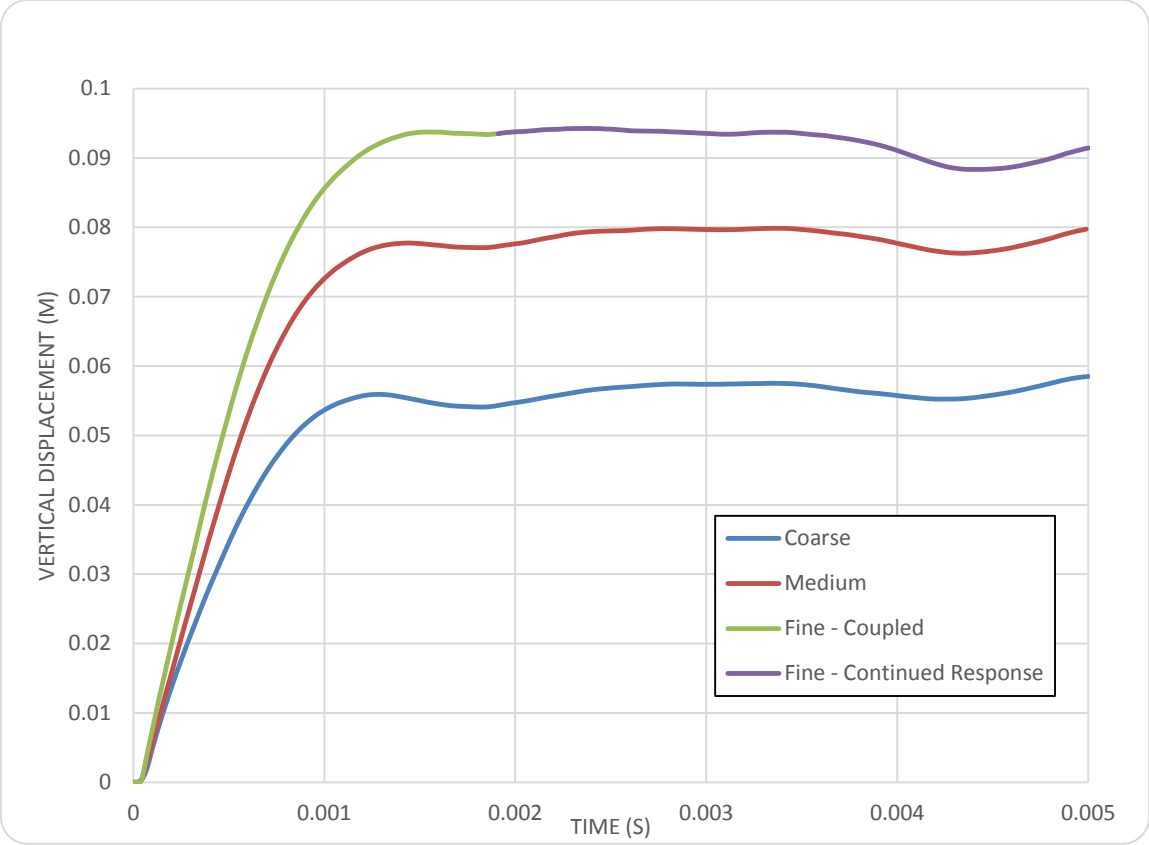

Figure 4.16 Skin-node deflection time history (645358).

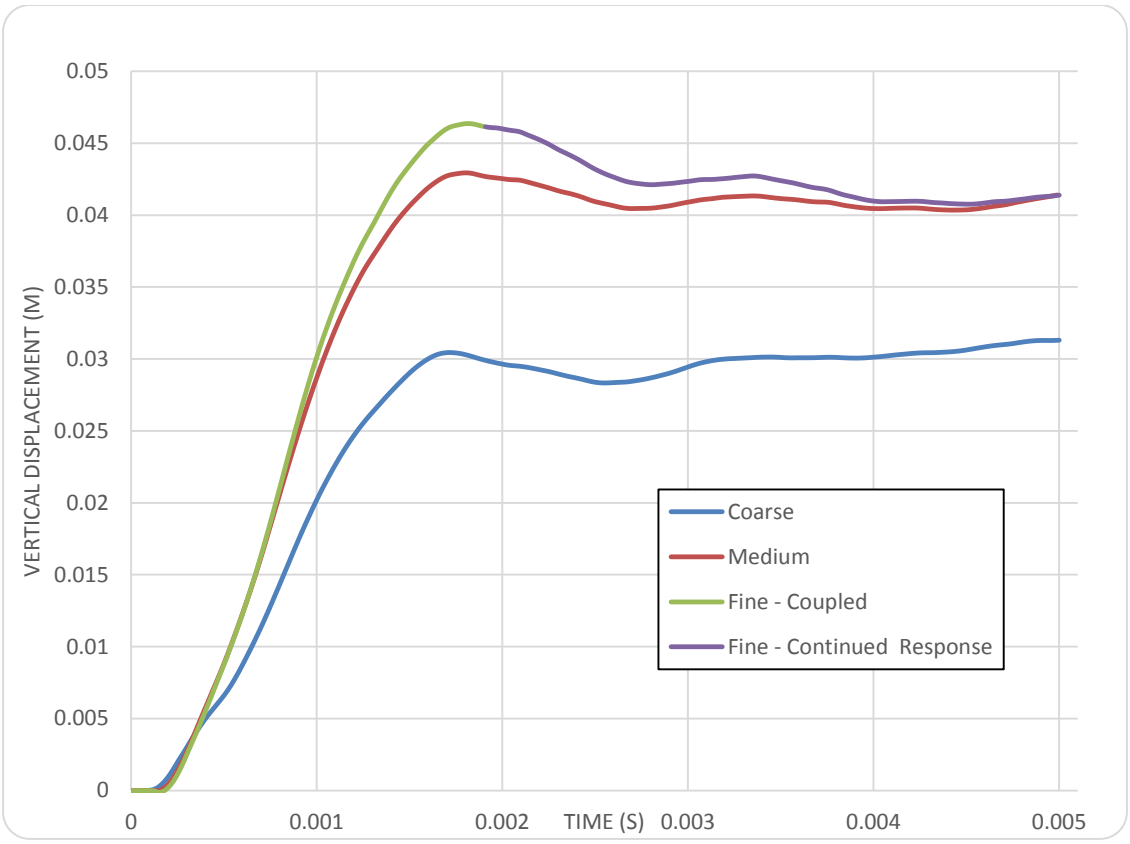

Figure 4.17 Frame-node deflection time history (612928).

CHAPTER V

CONCLUSION

A new point-projection method to transfer loads and displacements in a fluid-structure interaction problem was described. Unlike other point-projection methods, the method presented here gives unique solutions. This method, when coupling Loci/BLAST to LS-DYNA, was shown to give accurate results compared to experiments, to be work-conservative, and to be robust enough to work on complicated geometries. Problems were experienced when the CFD meshes would sometimes tangle during the mesh deformation process. It was proposed that this could be resolved by decreasing the LS-DYNA time step, though this hypothesis was not tested. Further investigation into the mesh tangling problem is required.

It would be beneficial if the validation and conservation tests were performed on the same problem with a more complicated geometry. The validation and conservation only tested the routine with quadrilateral CSD elements, so there could possibly be errors introduced with triangular CSD elements, which may or may not be due to non-conserved work. If experimental data could be found for a complex geometry (like the generic hull), then this new method could be compared to that experimental data, and the work-conservation routine could also be run at the same time to show how the works vary in the case. If the simulation results did not match experimental data, then the work-

conservation history could be consulted to determine if there is an error within the FSI routine.

REFERENCES

[1] Farhat, C., et al., "Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and applications to areoelasticity," *Computer Methods in Applied Mechanics and Engineering,* Vol. 157 (1998), pp. 95-114.

[2] Jaiman, R. K., et al., "Conservative load transfer along curved fluid-solid interface with non-matching meshes," *Journal of Computational Physics,* Vol. 218 (2006), pp. 372-397.

[3] Cebral, J. R. and R. Löhner, "Conservative Load Projection and Tracking for Fluid-Structure Problems," *AIAA Journal,* Vol. 35 No. 4 (1997).

[4] Fish, J. and T. Belytschko, *A First Course in Finite Element Analysis*. Chichester: John Wiley & Sons, 2007.

[5] Reddy, J. N., *An Introduction to the Finite Element Analysis*. 3rd Ed. New York: McGraw-Hill, 2006.

[6] "Chapter 12 Pyramid Solid Elements," *Advanced Finite Elements Department of Aerospace Engineering Science University of Colorado at Boulder.* Last updated 2013.

[7] P. Ivancic, J.M. Janus, E. Luke, D. Thompson, R. Weed, and J. Kang, "A conformal, Fully-Conservative Approach for Predicting Blast Effects on Ground Vehicles," NATO Paper MP-AVT-221-05, *Design and Protection Technologies for Land and Amphibious NATO Vehicles,* Copenhagen, DK, Apr 7-9, 2014.

[8] Livermore Software Technology Corporation. "LS-DYNA Keyword User's Manual Volume I." Version 971. May 2007.

[9] Livermore Software Technology Corporation. "LS-DYNA Keyword User's Manual Volume II." Version 971. May 2007.

[10] Luke, E. "BLAST 0.3: A Finite Volume Blast Simulation Tool – The User Guide." 17 Jan 2013.

[11] "LS-PRE/POST v1.0." Beta. 27 August 2002.

[12] Canada. Defense Research and Development Canada. *Comparison of ALE and SPH methods for simulating mine blast effects on structures.* DRDC Valcartier TR, 2010.

[13] Williams, K., et al., "Validation of a Loading Model for Simulating Blast Mine Effects on Armored Vehicles," *7th International LS-DYNA Users Conference.*

[14] Luke, E., et al., "A fast mesh deformation method using explicit interpolation," *Journal of Computational Physics,* Vol. 231 (2012), pp. 586-601.