

2D to 3D ALE Mapping

Nicolas Aquelet

Livermore Software Technology Corporation
7374 Las Positas Road, Livermore, CA 94551, USA
aquelet@lstc.com

Mhamed Souli

Laboratoire de Mecanique de Lille
Bd Langevin, 59655 Villeneuve d'Ascq, France
mhamed.souli@univ-lille1.fr

Abstract

A 2D MMALE (Multi-Material Arbitrary Lagrange Euler) code was implemented in LS-DYNA®. Like the 3D MMALE already available each 2D computational cycle is divided in two steps. First a multi-material version of the two-dimensional shell formulations solves the physical problem on quadrangle meshes during the Lagrangian step. The 2D shell formulations are plane strain and area-weighted axisymmetric. An advection step adapted to the 2D shell approaches follows to control the mesh motion. 2D ALE data of the last cycle can be mapped on 3D ALE mesh. Data are stored in a file defined on the command line after the prompt "map=". This file is read for the 3D model with the same command line.

Introduction

The keyword to run the 2D ALE code is composed of 4 parameters:

```
*SECTION_ALE2D
$ secid, aleform, aet, elform
```

- secid : Section id called in the 2nd column of *PART
- aleform : ALE formulation
 - aleform=11: Multi-Material ALE formulation
This formulation is currently Eulerian. So during the advection step the mesh is moved back to its initial position.
 - aleform=6 : Single Eulerian formulation
- aet : Ambient element type
- elform : Element formulation
 - elform=13 : Plane strain
 - elform=14 : Axisymmetric – area weighted

The Lagrangian step is solved by routines similar to the ones called by *SECTION_SHELL with elform=13 or 14 [1]. However for aleform=11 these routines loops on the number of material groups. 2D versions of the 3D advection routines were developed for the remap cycle. Some of the 3D ALE features are also available in the 2D code (a 2D version of ALE_FSI_PROJECTION [1] was developed and the volume fractions can also be initialized around the beams with INITIAL_VOLUME_FRACTION_GEOMETRY [1]). This paper will focus on the mapping between the 2D axisymmetric MMALE formulation and the 3D MMALE

formulation. The application in this paper is the explosion of a spherical charge in water. The problem is solved with the 2D (right column in the box) and 3D (left column in the box) MMALE formulations.

<pre> *KEYWORD *TITLE Spherical charge in water \$g,cm,microsec,100Gpa *CONTROL_TERMINATION 2.0 *CONTROL_ALE 2,1,2,-1 *SECTION_ALE2D 1,11,,14 *ALE_MULTI-MATERIAL_GROUP 1,1 2,1 *MAT_HIGH_EXPLOSIVE_BURN 1,1.67,0.747,0.25,0.0 *INITIAL_DETONATION 0,0.0,0.0,0.0,0.0 *EOS_JWL 1,4.911,0.091061,4.4,1.1,0.3,0.08,1.0 *MAT_NULL 2,1.0 *EOS_GRUNEISEN 2,0.148,1.92,0,0,0.1 *PART explosive 1,1,1,1 *PART water 2,1,2,2 *ELEMENT_SHELL \$ eid,pid,n1,n2,n3,n4 </pre>	<pre> *KEYWORD *TITLE Spherical charge in water \$g,cm,microsec,100Gpa *CONTROL_TERMINATION 2.0 *CONTROL_ALE 2,1,2,-1 *SECTION_SOLID 1,11 *ALE_MULTI-MATERIAL_GROUP 1,1 2,1 *MAT_HIGH_EXPLOSIVE_BURN 1,1.67,0.747,0.25,0.0 *INITIAL_DETONATION 0,0.0,0.0,0.0,0.0 *EOS_JWL 1,4.911,0.091061,4.4,1.1,0.3,0.08,1.0 *MAT_NULL 2,1.0 *EOS_GRUNEISEN 2,0.148,1.92,0,0,0.1 *PART explosive 1,1,1,1 *PART water 2,1,2,2 *ELEMENT_SOLID \$ eid,pid,n1,n2,n3,n4,n5,n6,n7,n8 </pre>
--	--

In this box the input decks for the 2D and 3D models for the explosion problem are presented in the left and right column respectively. The only differences are the section properties and the connectivity. Let's see what is behind each card and which impact each card could have on the mapping. The plan will be the following:

- First the axisymmetric MMALE formulation will be outlined through the ALE cards presented in the left box. Then the application to a half-spherical charge exploding in water will be presented and solved with the 2 input decks. The results will be compared.
- Finally the mapping will be described and illustrated on the previous problem. We will see how the mapping interacts with the routines of the *MAT_ and *EOS_ cards and how the 2 input decks will be modified. A new card will be introduced to enable the user to map the 2D data in different locations of the 3D ALE mesh.

2D MMALE FORMULATION

Description

The MMALE formulation requires:

- *CONTROL_ALE,
- *SECTION_ALE2D for a 2D model or *SECTION_SOLID for a 3D model,
- *ALE_MULTI_MATERIAL_GROUP

Let $\Omega \in R^3$, represent the domain occupied by the explosive and water and let $\partial\Omega$ denote its boundary. If the model is 3D, this domain is meshed by solid elements (*ELEMENT_SOLID) based on the 3D MMALE formulation (*SECTION_SOLID). If the model is 2D axisymmetric, Ω is meshed by shell elements (*ELEMENT_SHELL) based on the new 2D MMALE formulation. The equations of mass, momentum and energy conservation are the same for both ALE formulations:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \rho \operatorname{div}(\vec{v}) + (\vec{v} - \vec{w}) \operatorname{grad}(\rho) &= 0 \\ \rho \frac{\partial \vec{v}}{\partial t} + \rho (\vec{v} - \vec{w}) \cdot \overline{\overline{\operatorname{grad}(\vec{v})}} &= \overline{\overline{\operatorname{div}(\vec{\sigma})}} + \vec{f} \\ \rho \frac{\partial e}{\partial t} + \rho (\vec{v} - \vec{w}) \cdot \overline{\overline{\operatorname{grad}(e)}} &= \overline{\overline{\vec{\sigma} : \operatorname{grad}(\vec{v})}} \end{aligned} \quad (1)$$

\vec{v} and \vec{w} are the fluid and mesh velocity fields respectively and they can be different. Thus the mesh motion can be independent of the material motion. This is the main characteristic of the ALE formulation. If $\vec{v} = \vec{w}$, Eqs.(1) give the Lagrangian formulation (it can be obtained in both input decks by setting the 1st parameter on the 2nd line of *CONTROL_ALE to a value larger than the computation end time). Contrary to the 3D code which can compute \vec{w} through different remapping algorithms the current version of the 2D code is Eulerian for the moment. So $\vec{w} = \vec{0}$ and Eqs.(1) can be reduced to their conservative forms:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \vec{v}) &= 0 \\ \frac{\partial \rho \vec{v}}{\partial t} + \operatorname{div}(\rho \vec{v} \otimes \vec{v}) &= \overline{\overline{\operatorname{div}(\vec{\sigma})}} + \vec{f} \\ \frac{\partial \rho e}{\partial t} + \operatorname{div}(e \vec{v}) &= \overline{\overline{\vec{\sigma} : \operatorname{grad}(\vec{v})}} \end{aligned} \quad (2)$$

Equations (2) can have the following form:

$$\frac{\partial \phi}{\partial t} + \operatorname{div}(\Phi) = S \quad (3)$$

$$\text{where } \phi = \begin{bmatrix} \rho \\ \rho \vec{v} \\ \rho e \end{bmatrix}, \Phi = \begin{bmatrix} \rho \vec{v} \\ \rho \vec{v} \otimes \vec{v} \\ e \vec{v} \end{bmatrix} \text{ and } S = \begin{bmatrix} 0 \\ \overline{\overline{\operatorname{div}(\vec{\sigma})}} + \vec{f} \\ \overline{\overline{\vec{\sigma} : \operatorname{grad}(\vec{v})}} \end{bmatrix}$$

Equation (3) is solved in 2 steps by the split approach detailed in [2],[3]: first a Lagrangian cycle and then an advection cycle.

During the Lagrangian cycle a Finite Element code solves the following equation:

$$\frac{\partial \phi}{\partial t} = S \quad (4)$$

During the advection cycle the mesh is moved back to its initial position. A Finite Volume method maps the Lagrangian solution on the new position of the mesh by solving the following transport equation:

$$\frac{\partial \phi}{\partial t} + \text{div}(\Phi) = 0 \quad (5)$$

For the Lagrangian step in 2D, when `elform=14` in `*SECTION_ALE2D`, Eq.(4) is solved with the same routines than the ones called by `*SECTION_SHELL` [1] for the same element formulation (`elform=14` see [1]). Thus Eq.(4) is integrated on an element volume $x.dx.dy$ in the x-y plane where the y-axis is the symmetric axis and the x-axis is the radial direction. When applying the Petrov-Galerkin Finite Element method to the momentum equation, the weighting functions are divided by x [4]. It's why the element formulation is called "area weighted". The routines of `elform=14` (and `elform=13`) were modified to include a loop on the number of material groups defined by `*ALE_MULTI-MATERIAL_GROUP`.

For the advection step in 2D, when `aleform=11` in `*SECTION_ALE2D`, Eq.(5) is solved with the same routines than the ones called by `*SECTION_SOLID` [1] for the Multi-Material ALE element formulation (`elform=11` see [1]). 2D versions of these routines were derived. Thus a 2D young method defines the material interfaces in the ALE quadrangles and 2D advection routines compute the material fluxes according the position of these interfaces. The flux is still volume defined even if the Galerkin method in the Lagrangian step is area-weighted (except for `elform=13` in which the flux is area-defined). In `*CONTROL_ALE`, the third parameter on the first line controls which advection method is employed to compute the fluxes. There are 2 methods :1 for the donor cell method and 2 for the Van Leer method [5]. The 2nd method is used in the application.

Application

The application is showed on Fig.1 and Fig.2. The purpose is to model the water blast wave propagation due to the detonation of a pentolite charge with a radius of 2mm. The 3D mesh is a 1-cm-radius eighth sphere composed of 3500 solid elements and the 2D mesh is a 1-cm-radius quarter disk composed of 475 shell elements. The pressure histories for two different radius and along each axis (see Fig.1 and Fig.2) were plotted on Fig.3 and 4. The A and B curves agree well and show that the spherical symmetry is preserved by the 2D code. These plots are in good concordance with the 3D plots (C, D and E). This similitude shows again that the symmetry is well preserved and mostly 2D and 3D model predict the same pressure field.

The 2D model takes 57sec to terminate while the 3d one takes 670sec on a AMD Athlon(tm) 64 Dual Core Processor 5600+ (2.8GHz). So 2D is about 10 times faster than 3D for this application and finally the results are the same. Thus the idea of the mapping is to avoid the time waste of computing the wave propagation with a 3D model by replacing it by a 2D axisymmetric model. First the 2D model computes the solution until the symmetry is not respected anymore and then the 3D model takes over the computation after an initialization with the data from the last 2D cycle.

MAPPING

The mapping of the 2D ALE data on the 3D ALE mesh requires to modify the 2 input decks presented previously.

Input decks

<pre> *KEYWORD *TITLE Spherical charge in water \$g, cm, microsec, 100Gpa *CONTROL_TERMINATION 2.0 *CONTROL_ALE 2, 1, 2, -1 *SECTION_ALE2D 1, 11, , 14 *ALE_MULTI-MATERIAL_GROUP 1, 1 2, 1 *MAT_HIGH_EXPLOSIVE_BURN 1, 1.67, 0.747, 0.25, 0.0 *INITIAL_DETONATION 0, 0.0, 0.0, 0.0, 0.0 *EOS_JWL 1, 4.911, 0.091061, 4.4, 1.1, 0.3, 0.08, 1.0 *MAT_NULL 2, 1.0 *EOS_GRUNEISEN 2, 0.148, 1.92, 0, 0, 0.1 *PART explosive 1, 1, 1, 1 *PART water 2, 1, 2, 2 *ELEMENT_SHELL \$ eid, pid, n1, n2, n3, n4 </pre>	<pre> *KEYWORD *TITLE Spherical charge in water \$g, cm, microsec, 100Gpa *CONTROL_TERMINATION 2.0 *CONTROL_ALE 2, 1, 2, -1 *SECTION_SOLID 1, 11 *ALE_MULTI-MATERIAL_GROUP 1, 1 2, 1 *MAT_NULL 1, 1.67 *INITIAL_ALE_MAPPING 100, 0, 200 0, 0, 0, 300 *SET_PART 100 1, 2 *SET_MULTI-MATERIAL_GROUP_LIST 200 1, 2 *DEFINE_VECTOR 300, 0, 0, 0, 0, 0, 1 *EOS_JWL 1, 4.911, 0.091061, 4.4, 1.1, 0.3, 0.08, 1.0 *MAT_NULL 2, 1.0 *EOS_GRUNEISEN 2, 0.148, 1.92, 0, 0, 0.1 *PART explosive 1, 1, 1, 1 *PART water 2, 1, 2, 2 *ELEMENT_SOLID \$ eid, pid, n1, n2, n3, n4, n5, n6, n7, n8 </pre>
--	--

The introduction of the mapping changes the explosive material cards and introduces a new card. Let's see first how the mapping is working through the explosive and water cards and so why we need to do these modifications. The purpose of the new card introduced in the 3D input deck will be presented. Finally the explosive application will illustrate the mapping. 3D results after the mapping will be compared with the results of the previous application.

Explosive Material

The explosive modeling requires:

- *MAT_HIGH_EXPLOSIVE_BURN,
- *INITIAL_DETONATION and
- *EOS_JWL

The parameters in *INITIAL_DETONATION involves that the ignition point is applied at time $t_d = 0\mu s$ in (0,0,0) which is the center of the sphere. In this case the lighting time t_L for any explosive elements is the ratio of the distance from the center of the element to the ignition point L_d and the detonation velocity D , which is the 2nd parameter in *MAT_HIGH_EXPLOSIVE_BURN:

$$t_L = t_d + \frac{L_d}{D} \quad (6)$$

However L_d is not the same for the 2D and 3D mesh because the center positions of the 3D elements depend on the refinement in the out-of-plane direction. Moreover the burn fraction, $0 \leq F \leq 1$, which controls the release of the chemical energy, might also be different between the 2D and 3D model.:

$$F = \max(F_1, F_2)$$

$$F_1(2D) = \begin{cases} \frac{2(t-t_d)DL_{e\max}}{3A_e} & \text{if } t > t_d \\ 0 & \text{if } t \leq t_d \end{cases} \quad F_1(3D) = \begin{cases} \frac{2(t-t_d)DA_{e\max}}{3v_e} & \text{if } t > t_d \\ 0 & \text{if } t \leq t_d \end{cases} \quad (7)$$

$$F_2 = \frac{1-V}{1-V_{cj}}$$

$V = \rho_0^{he} / \rho^{he}$ is the relative volume or the ratio of the initial density and the current density of the material. V_{cj} is the Chapman-Jouguet relative volume.

When *MAT_HIGH_EXPLOSIVE_BURN is in the input deck F is computed to adjust the pressure. The difference in the computation of F comes from the definitions of F_1 in 2D and 3D set side by side in Eq.(7). The ratio in 2D of the maximum element length and area $L_{e\max} / A_e$ is generally not equal to the ratio in 3D of the maximum element area and volume $A_{e\max} / v_e$. To know what is the real impact of these 2 differences on the results, the 3D values of the previous geometric ratio and t_L were mapped on the 2D explosive elements. The comparison of waterblast pressure histories in the locations of Fig.1 and Fig.2 with and without this adjustment revealed no discrepancy. So the mapping of these 2 data is unnecessary. Furthermore the mapping should be used after the complete burn of the explosive. In this case $F = 1$. So *MAT_HIGH_EXPLOSIVE_BURN might be replaced by *MAT_NULL in the 3D input deck. This replacement is not required (anymore) because the initial time of the 3D run is shifted to match the 2D final time. Thus F is still equal to 1 and the explosive will not burn a

second time during the 3D run. The factor F controls the release of the explosive internal energy by multiplying the pressure computed by the JWL (Jones-Wilkins-Lee) equation of state *EOS_JWL:

$$p = Fp^{he} \tag{8}$$

$$p^{he} = A \left(1 - \frac{\omega}{R_1 V} \right) \exp(-R_1 V) + B \left(1 - \frac{\omega}{R_2 V} \right) \exp(-R_2 V) + \frac{\omega E^{he}}{V} \tag{9}$$

where A, B, R_1, R_2, ω are experimental constants. E^{he} is the specific internal energy. The properties of the high explosive material modeled in this paper are given in Tab.1 [6]. In this table E_0^{he} is the initial specific internal energy and P_{cj} is the Chapman Jouguet pressure.

$A(MBar)$	$B(MBar)$	R_1	R_2	ω	$E_0^{he} (MBar)$	V_0^{he}
4.911	0.091061	4.4	1.1	0.3	0.08	1
$D(cm.\mu s^{-1})$	$\rho_0^{he} (g.cm^{-3})$		$P_{cj} (MBar)$			
0.747	1.67		0.25			

Table 1. *Pentolite properties*

During the 1st Lagrangian cycle of the 3D case the JWL equation will compute the pressure. So the specific internal energy and relative volume of the last 2D time step is mapped to the 3D model to initialize E_0^{he} and V_0^{he} respectively. The density is also mapped to compute the initial nodal and element masses.

Water Material

The water modeling requires:

- *MAT_NULL and
- *EOS_GRUNEISEN (or *EOS_LINEAR_POLYNOMIAL)

The Mie-Grüneisen equation of state computes the water pressure with the following equation:

$$p^{water} = \frac{\rho_0^{water} C^2 \mu \left[1 + \left(1 - \frac{\gamma_0}{2} \right) \mu - \frac{a}{2} \mu^2 \right]}{\left[1 - (S_1 - 1) \mu - S_2 \frac{\mu^2}{\mu + 1} - S_3 \frac{\mu^3}{(\mu + 1)^2} \right]^2} + (\gamma_0^{water} + a \mu) E^{water} \tag{10}$$

in which $\mu = \frac{\rho^{water}}{\rho_0^{water}} - 1 = \frac{1}{V_0^{water}} - 1$. In the EOS card most of the parameters are zero but the ones in the following table [7]:

$\rho_0^{water} (g.cm^{-3})$	γ_0^{water}	$C(cm.\mu s^{-1})$	S_1	V_0^{he}
1	0.1	0.148	0.08	1.92

Table 2. *Water properties*

γ_0^{water} is the Gruneisen gamma. C and S_1 are parameters in the shock velocity (v_s) – particle velocity (v_p) relation: $v_p = C + S_1 v_s$. Here again for the mapping after the 1st 3D Lagrangian cycle the pressure will be computed from the relative volume V_0^{water} and internal energy E_0^{water} mapped from their last 2D values. For an explosion the bulk forces are much more larger than the viscous forces. It's why *MAT_NULL only gives the initial density: $\rho_0^{water} = 1g.cm^{-3}$. As for the explosive the 2D density field is mapped on the 3D mesh to compute the initial nodal and element masses.

*Initial_ALE_mapping

```
*INITIAL_ALE_MAPPING
$ id, styp, ammsid
  100,    0,    200
$ x0,  y0,    z0, vid
   0,   0,    0, 300
```

- id : Part id or part set id
- styp : Set type
 - styp=0: Part set
 - styp=1: Part
- ammsid : ALE Multi-Material set id
- X0,Y0,Z0 : Origin of the coordinates system
- vid : Vector id

The routines of this card need to know which mesh will be initialized with the 2D data and more specifically which multi-material groups. The first 2 parameters (id and styp) defines the mesh and the third one (ammsid) calls the *SET_MULTI-MATERIAL_GROUP_LIST card. This card will define a list of 3D material groups. The rank in this list should match the rank of the multi-material groups (as a reminder the ranks of multi-material groups are defined by *ALE_MULTI-MATERIAL_GROUP). Let's take an example. For instance if the 2D model has 3 groups, the 3D one has 5 groups and the following mapping is wanted:

- The 1st group in 2D is mapped on the 3rd one in 3D,
- The 2nd group in 2D is mapped on the 5th one in 3D,
- The 3rd group in 2D is mapped on the 4th one in 3D

The *SET_MULTI-MATERIAL_GROUP_LIST card should be set as follows:

```
*SET_MULTI-MATERIAL_GROUP_LIST
300
3,5,4
```

The 2D data can be mapped in different parts of the 3D mesh by defining the origin of the coordinate system (x0,y0,z0) and the direction of the axisymmetric axis (vid).

Application

The same application as previously will show how the mapping is working. First the 2D model runs from 0 to 2microsec with the following command:

```
ls971 i=explo.k map=2dto3d
```


The data of the last cycle are saved in a binary file called 2dto3d.

Second the 3D model runs from 2microsec to the end with the same command. The 3D model reads the data in 2dto3d and initializes the 3D model by interpolating the 2D values. So the 2D mesh should be large enough to “include” the 3D mesh. If a node or an element of the 3D mesh is out of the interpolation range defined by the size of the 2D mesh, the mapping will skip it. The 3D mesh can be smaller than 2D but there should be no velocity on the 2D boundary nodes (except along the axis). The 2D model of this paper was created from the x-y face of the 3D model through Lsprepost > 2 > Elgen > shell > Shell By: Solid_face. The positions of the 2D nodes on the circular bound were scaled up by 10% through Lsprepost > 2 > Scale.

Figures 5 to 6 give the pressure histories at the locations showed on Fig.1 and 2. On Fig.5 the x-cross curves are computed by the 2D code before the mapping at t=2microsec. The +-cross curves are the histories given by the 3D code after the mapping time. We can see a good agreement between these plots and the B curves computed previously by the 3D code. Figure 6 compares plots from the 3D model. The difference between the A-curve and the other ones is the mapping. The B, C and D pressure histories are obtained after the mapping at t = 2microsec while the A pressure was computed by the 3D code only. The B, C and D pressure histories are located on the radius along each axis. The four curves agree well. So the spherical symmetry is preserved in the 3D model after the mapping. This comparison shows also that a 3D run after the mapping is equivalent to a simple 3D run. The 2D run takes 28sec and the 3D run, 343 sec. So the overall computation time is 371sec whereas the 3D model by itself took 670sec. The 2D computational time is so small that the speedup clearly depends on when the mapping time is done. The latter the 3D code takes over the computation, the higher the speedup is. So the mapping is interesting only if the blast wave or any other axisymmetric phenomena takes CPU time in a 3D model before reaching the symmetry limits.

CONCLUSIONS

A new 2D ALE code was presented and applied to an axisymmetric blast wave propagation. The 2D and 3D results met a good agreement. Since the 2D code was faster than the 3D code, it was showed that computation time can be saved if the 2D code took care of the blast wave propagation phase. Then if the axisymmetry condition was no more respected because of a coupling with a structure for instance the 3D ALE code could take over the computation through a mapping. The next step of these developments will be the mpp version of the 2D code.

References

- [1] “LS-DYNA Keyword User’s Manual version 971”, Livermore Software Technology Corporation, May 2007.
- [2] Benson D.J., 1992, “Computational methods in Lagrangian and Eulerian hydrocodes,” *Computer Methods in Applied Mechanics and Engineering*, **99**(2), pp.235-394.
- [3] Hughes, T.J.R., Liu, W.K., Zimmerman, T.K., 1981, “Lagrangian Eulerian finite element formulation for viscous flows,” *J. Comput. Methods Appl. Mech. Engrg.*, **21**, pp.329-349.
- [4] Hallquist J.O., “LS-DYNA, Theoretical Manual,” LSTC, Livermore, 1998.
- [5] Van Leer, B., “Towards the Ultimate Conservative Difference Scheme. IV. A New Approach to Numerical Convection”, *Journal of Computational Physics* 23 (1977) p276-299
- [6] Alia, A., Aquelet, N., Souli, M., Olovsson, L., “A delayed remap technique in multi-material ALE methods”, *European Journal of Computational Mechanics*, Vol 15 No.5 (2006)
- [7] Mahmadi, K., Itoh, S., Hamada, T., Aquelet, N., Souli, M., “Numerical Studies of Wave Generation using Spiral Detonating Cord”, *Materials Science Forum* Vols. 465-466 (2004) pp. 439-444

LS-DYNA keyword deck by LS-PRE
 Time = 2
 Contours of Pressure
 max ipt. value
 min=-0, at elem# 3706
 max=0.00687146, at elem# 3685

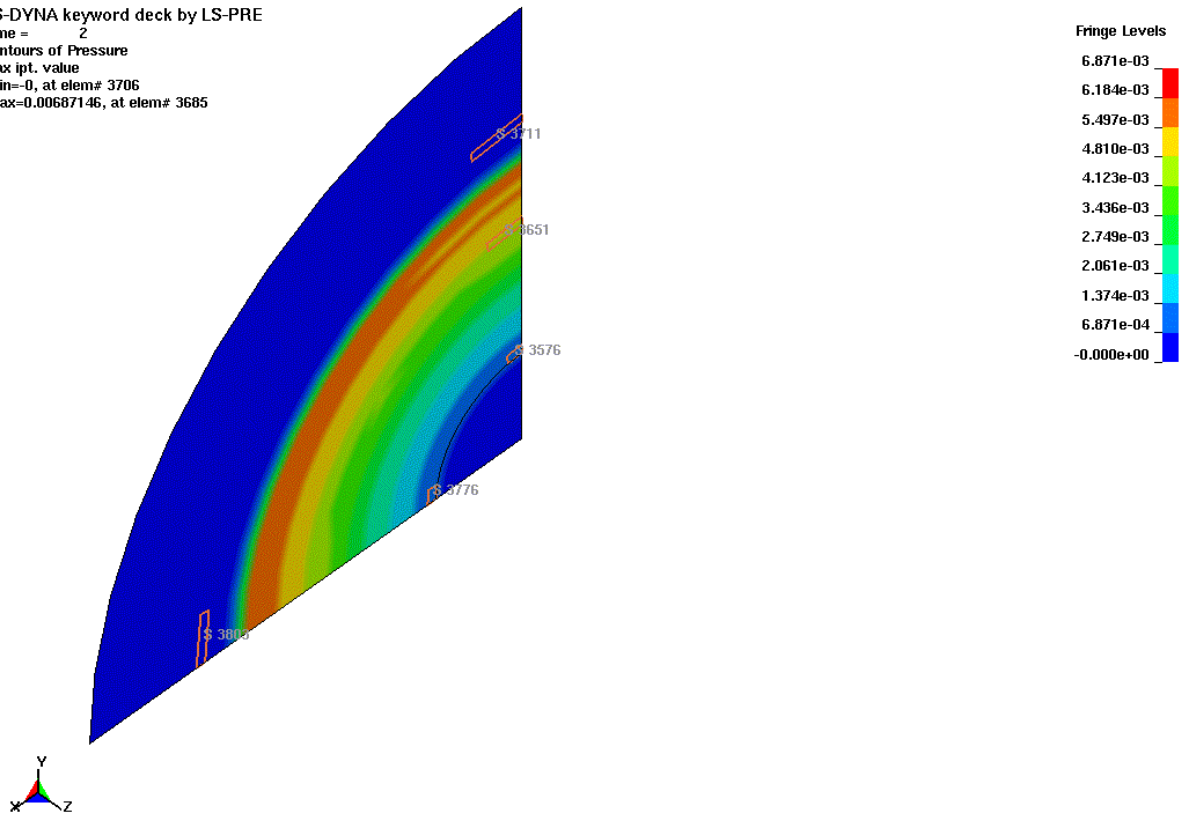


Fig.1: 2D Pressure field before the mapping at $t = 2.\mu s$. Location of the pressure histories

Spherical charge in water
 Time = 2
 Contours of Pressure
 min=-0, at elem# 901
 max=0.00687144, at elem# 785

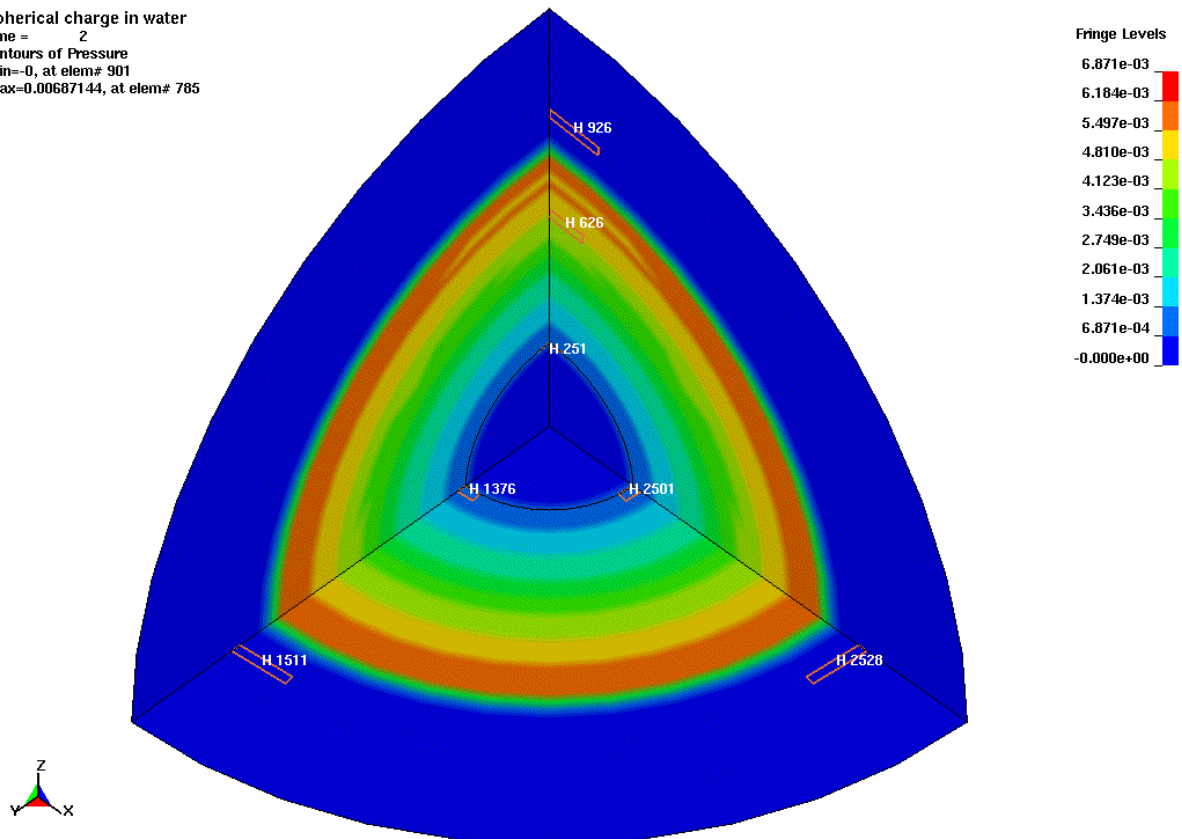


Fig.2: 3D Pressure field after the mapping at $t = 2.\mu s$. Location of the pressure histories

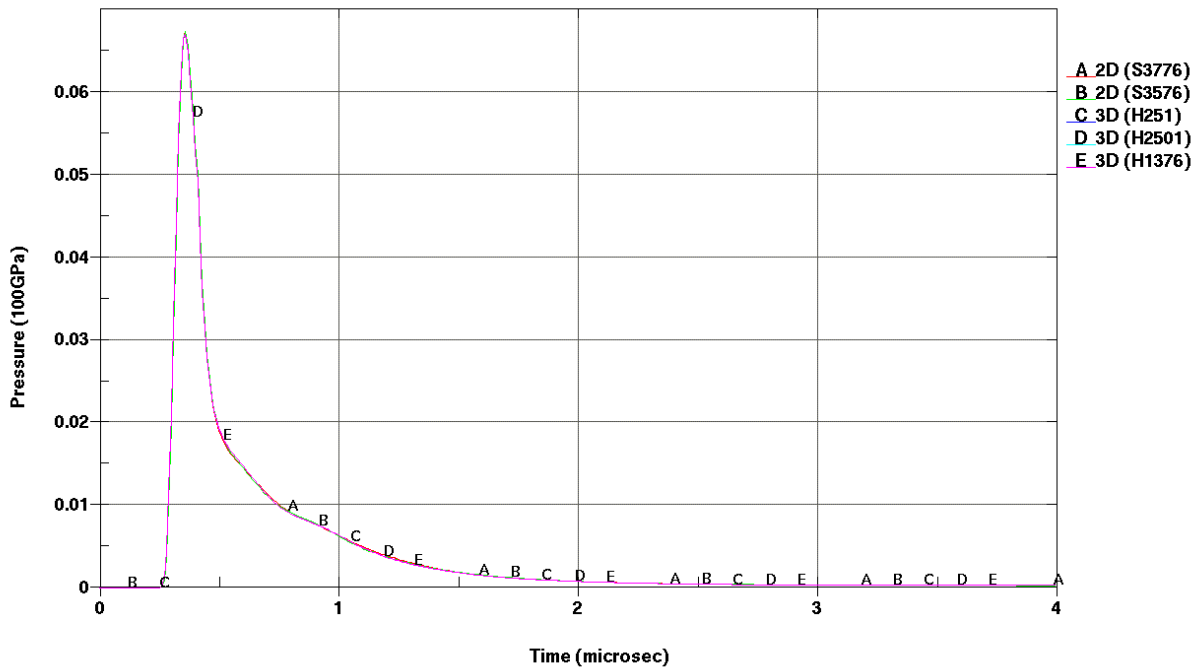


Fig.3: Pressure histories for the 2D (A and B) and 3D (C,D and E) models. The pressures were measured at the same distance from the origin and along each axis (see Fig1 and 2: shell 3776 and solid 2501 for x, shell 3576 and solid 1676 for y, solid 251 for z)

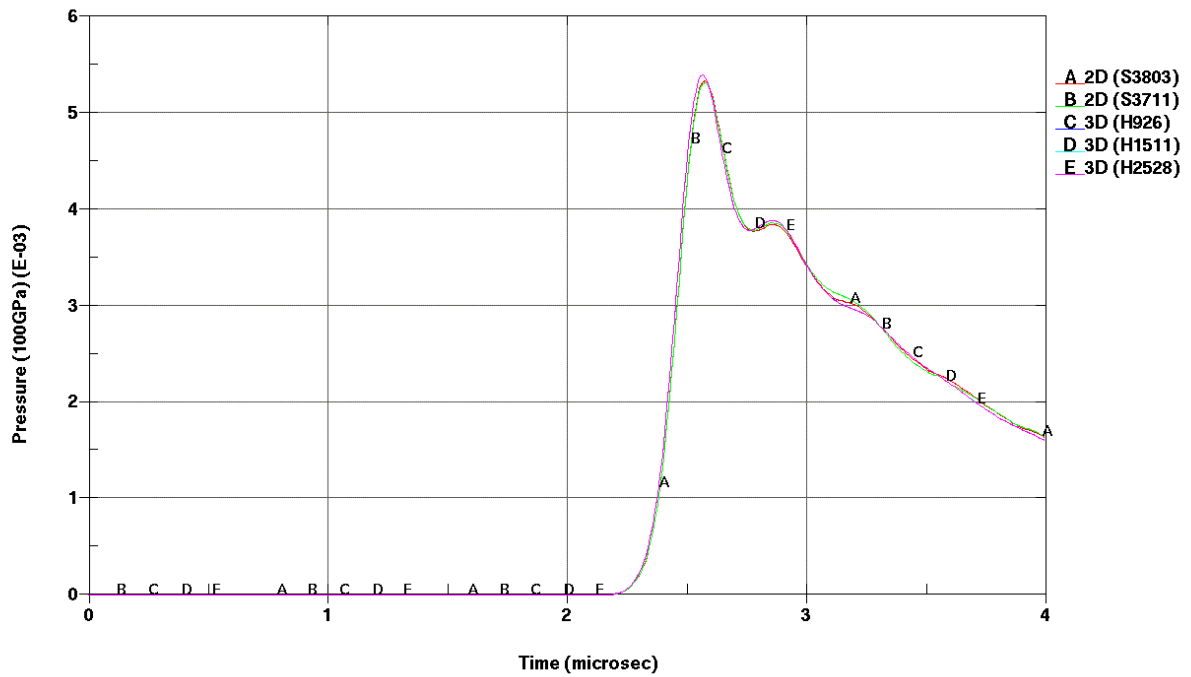


Fig.4: Pressure histories for the 2D (A and B) and 3D (C,D and E) models. The pressures were measured at the same distance from the origin and along each axis (see Fig1 and 2: shell 3803 and solid 2528 for x, shell 3711 and solid 1511 for y , solid 926 for z)

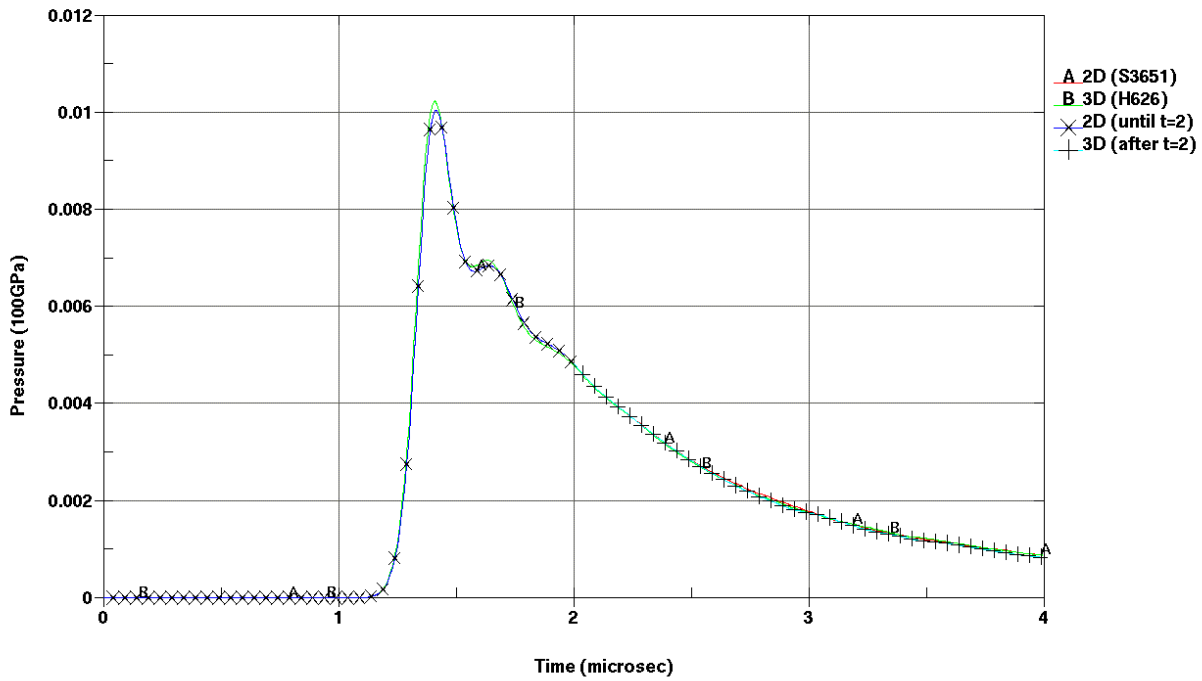


Fig.5: Pressure histories for the 2D (A and x) and 3D (B and +) models. The 2D pressures were measured in the shell 3651 (see Fig.1). The 3D pressures were measured in the solid 626 (see Fig.2). The x curve is the 2D pressure history before the mapping at t=2microsec. The + curve is the 3D pressure history after the mapping at t=2microsec

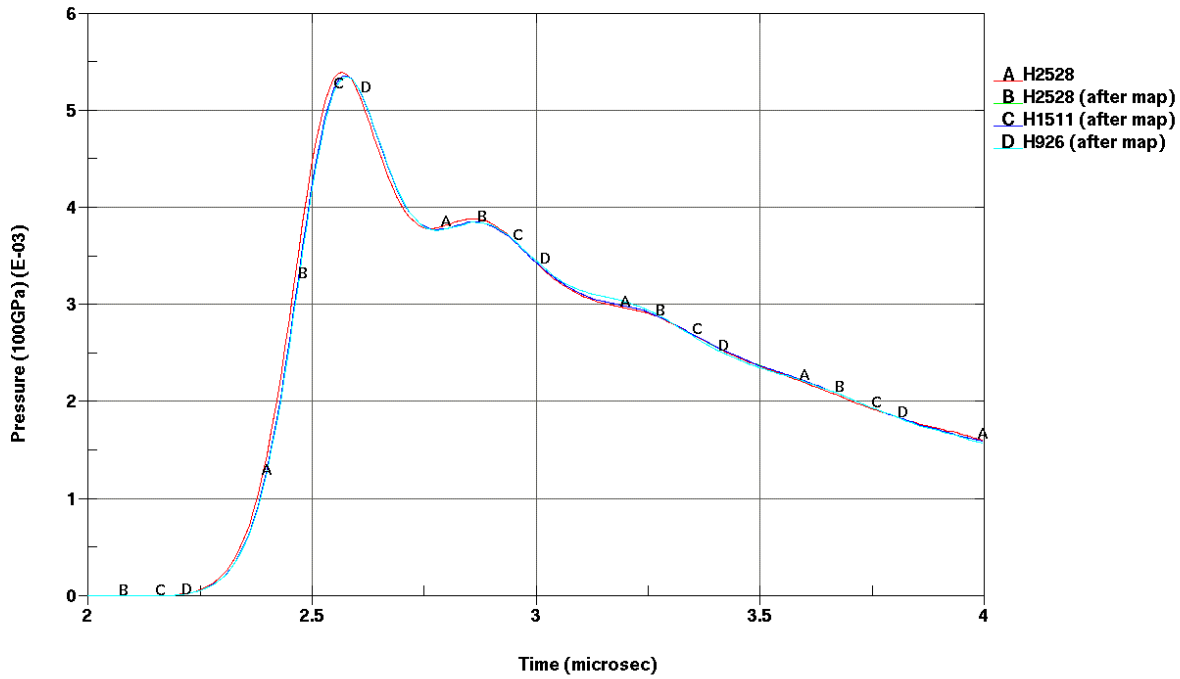


Fig.6: Pressure histories for an overall 3D (A) and 3D after the mapping (B,C and D) models. The pressure were measured at the same distance from the origin along each axis (solid 2528 for x, 1511 for y and 926 for z, see Fig.2)