ALE Adaptive Mesh Refinement in LS-DYNA®

Nicolas AQUELET
Livermore Software Technology Corp.
7374 Las Positas Rd
Livermore CA94550
aquelet@lstc.com

Abstract

*An adaptive mesh refinement capability was implemented in the 3D MMALE (Multi-Material Arbitrary Lagrange Euler) code. It automatically and locally refines (or coarsens) ALE hexahedral solid elements. The keyword *REFINE_ALE (or *ALE_REFINE) activating this feature has 3 lines of parameters that enable 3 different kinds of refinement. If the keyword has only one line, the refinement is static and only occurs during the initialization. If the second line is added, the refinement becomes dynamic. Adding the third line allows removing refined meshes.*

Introduction

The Multi-Material Arbitrary Lagrangian Eulerian (MMALE) formulation is based on three domains: the initial configuration of the material, the current configuration of the material or spatial domain, and the referential or ALE domain [1]. A map between the first and second domain defines the material motion while a map (called $\vec{\varphi}$ here) between the third and second ones determines the mesh motion. If a MMALE model is 3D, hexahedral elements (*ELEMENT_SOLID and *SECTION_SOLID with elform=11 in LS-DYNA® [2]) are usually considered to mesh the referential domain. In each of these elements, the equations for mass, momentum and energy conservation are centered integrated [1]:

$$\begin{aligned}
&\frac{\partial \rho}{\partial t} + \rho div(\vec{v}) + (\vec{v} - \vec{v}_{mesh}) grad(\rho) = 0 \\
&\rho \frac{\partial \vec{v}}{\partial t} + \rho(\vec{v} - \vec{v}_{mesh}).\overline{\overline{grad}}(\vec{v}) = \overrightarrow{div}(\overline{\overline{\sigma}}) + \vec{f} \\
&\rho \frac{\partial e}{\partial t} + \rho(\vec{v} - \vec{v}_{mesh}).\overrightarrow{grad}(e) = \overline{\overline{\sigma}} : \overline{\overline{grad}}(\vec{v})
\end{aligned} \tag{1}$$

The density, velocities and internal energy in the time derivative are functions of the referential coordinates whereas they depend on the spatial coordinates in the divergence and gradients. $\vec{v}$ and $\vec{v}_{mesh}$ are the material and mesh (or referential) velocity fields respectively and they can be different. Thus the mesh motion can be independent of the material motion. This is the main characteristic and advantage of the ALE formulation. The mesh motion can be arbitrarily defined by the user. $\vec{v}_{mesh}$ can be computed and controlled with the keyword *ALE_REFERENCE_SYSTEM_GROUP and the advection factors in *CONTROL_ALE. The advantage of a free mesh motion is that the nodes can be moved in a zone of interest to locally refine the mesh. For example, the mesh smoothing PRTYPE=8 in *ALE_REFERENCE_SYSTEM_GROUP contracts the mesh in the vicinity of a shock front [3]. However even if smoothing the mesh seems to be an interesting option to refine the mesh, the drawback is that refining an area usually involves coarsening another. Also the mesh refinement

is not immediate because the progressive mesh motion depends on the advection that along with the Lagrangian cycle are the main stages of a ALE computational time step [4], [5] . During the Lagrangian cycle a finite element code solves the physical part of (1) by integrating the following equation:

$$\frac{\partial \phi}{\partial t} = S \tag{2}$$

where $\phi = \begin{bmatrix} \rho \\ \rho\vec{v} \\ \rho e \end{bmatrix}$ and $S = \begin{bmatrix} \underline{\underline{0}} \\ div(\underline{\underline{\sigma}}) + \vec{f} \\ \underline{\underline{\sigma}} : \underline{\underline{grad}}(\vec{v}) \end{bmatrix}$ are functions of the referential coordinates. The mesh position is updated to follow the material motion as shown on Fig.1.

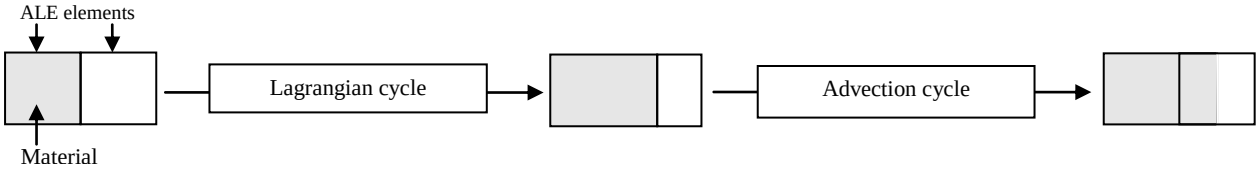


Figure 1: Lagrangian and advection cycles in a computational time step

During the advection the mesh is arbitrarily moved (or moved to its initial position for an Eulerian model like on Fig.1) and a finite volume method maps the Lagrangian solution on the new mesh positions by solving the transport equation (3). The computation of the fluxes through the faces moving to their new positions allows determining how much mass, momentum and energy an ALE cell (donor) gives to its neighbors (receptors).

$$\frac{\partial \phi}{\partial t} + div(\Phi) = 0 \qquad \text{with } \Phi = \begin{bmatrix} \rho\vec{w} \\ \rho\vec{v} \otimes \vec{w} \\ \rho e\vec{w} \end{bmatrix} \tag{3}$$

$\vec{w}$ is the material velocity in the referential coordinate system, which is related to the mesh and material velocity by: $\vec{v} = \vec{v}_{mesh} + \vec{w}.\underline{\underline{grad}}\varphi$. The fluxes computed by the finite volume method are much smaller than the volumes of the meshes. Several advection cycles are required to get a significant refinement. Another problem is that mesh distortions could occur if a refinement criterion forced the mesh to contract around complex geometries. So the ALE mesh motion is not considered in this paper to refine the mesh[1]. A solution like the usual adaptive meshing controlled by *CONTROL_ADAPTIVE is preferred.

During the preprocessing refining an ALE element requires refining its neighbor elements as well. If the sketch (a) Fig.2 represents an ALE hexahedral element (cell in grey) in a

[1] Actually most of the ALE applications are Eulerian models where $\vec{v}_{mesh} = \vec{0}$ (AFAC=-1 in *CONTROL_ALE) and the ALE codes solves the Eulerian conservative form of Eq.(1):

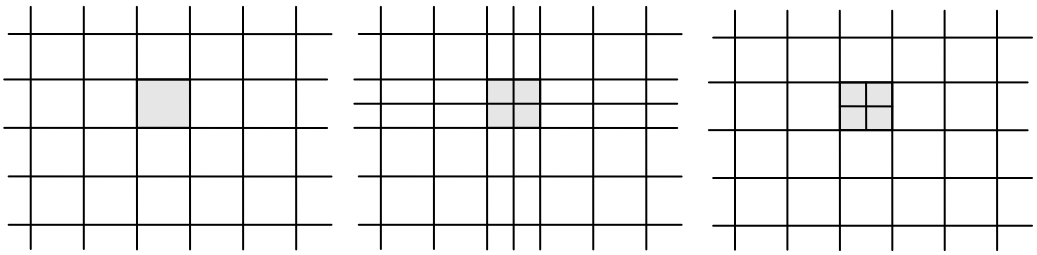
$$\frac{\partial \rho}{\partial t} + div(\rho\vec{v}) = 0$$

$$\frac{\partial \rho\vec{v}}{\partial t} + div(\rho\vec{v} \otimes \vec{v}) = div(\underline{\underline{\sigma}}) + \vec{f}$$

$$\frac{\partial \rho e}{\partial t} + div(e\vec{v}) = \underline{\underline{\sigma}} : \underline{\underline{grad}}(\vec{v})$$

If $\vec{v} = \vec{v}_{mesh}$, the equations (1) give the Lagrangian formulation (it can be obtained by setting the 1st parameter on the 2nd line of *CONTROL_ALE to a value larger than the computation end time).

computational mesh, refining this element involves refining its column and row as illustrated on the sketch (b).



(a) (b) (c)

Figure 2: Sketch of refined mesh. (a) Initial ALE mesh. (b) Refinement with pre-processing. (c) Refinement with * REFINE_ALE

*CONTROL_ADAPTIVE can be applied to several shell and solid element formulations to achieve a result like the sketch (c). However, the MMALE formulation needs its own adaptive mesh refinement mainly because of the advection. A multi-material ALE element can have several material interfaces that need to be located with the Young method [6] in each mixed cell for the advection. If a material mixed element gives mass, momentum and energy to its neighbor like on Fig.1, the fluxes of these quantities need to be computed for each material according the material interface positions. If the neighbor cell is refined like on the sketch (c), the material fluxes must also take the division of the face into account to determine the material volume that each subcell needs to receive. So a specific treatment is required to handle this case. Moreover in this configuration (advection from coarse to fine meshes), the Half Index Shift code [7] that deals with the momentum advection as if the nodes are cell centered needs to be also updated when a cell has a finer neighbor. Thus new developments need to be added to the Lagrangian and advection cycles for a local refinement. A new ALE keyword, *REFINE_ALE, has been developed for this purpose. For each line in this keyword, there is a feature that will be described in the next paragraphs:

- If only the 1st line is defined, the refinement is static and the elements are only refined at the beginning of the run. This feature can be useful if the user already knows which part of the ALE mesh needs to be refined.
- If the regions to refine are not known or depends on the complex physics of the model, adding the second line allows defining a criterion (among other parameters) to dynamically refine elements meeting the required conditions.
- The 3rd line defines also a criterion to remove refining elements. With this last line the refinement can be adapted to be only localized in regions defined by the user.

The following terms will be used:

- Donor: ALE cell that gives a flux during the advection.
- Acceptor: ALE cell that receives a flux during the advection.
- Child or Children : ALE subcells that divide an ALE element into 8 volumes
- Parents: ALE elements that have children
- Cluster: a group of 2x2x2 child elements used for the refinement

Static Refinement

Description

*REFINE_ALE

Variable	ID	TYPE	NLVL					
Type	I	I	I					

VARIABLE	DESCRIPTION
ID	Set ID.
TYPE	Set type:: EQ.0: ALE Part Set, EQ.1: ALE Part, EQ.5: Solid Set
NLVL	Number of refinement levels

ID and TYPE defines which region of the mesh will be refined. If TYPE=1, ID is a part, a name given in LS-DYNA® to a region of a mesh associated with a specific element type (in this case: ALE), a specific material and equation of state. If TYPE=0, ID represents a group of parts. If TYPE=5, ID is a set of selected ALE elements. The refinement occurs during the initialization. The flowchart of Fig.3 shows the main steps.

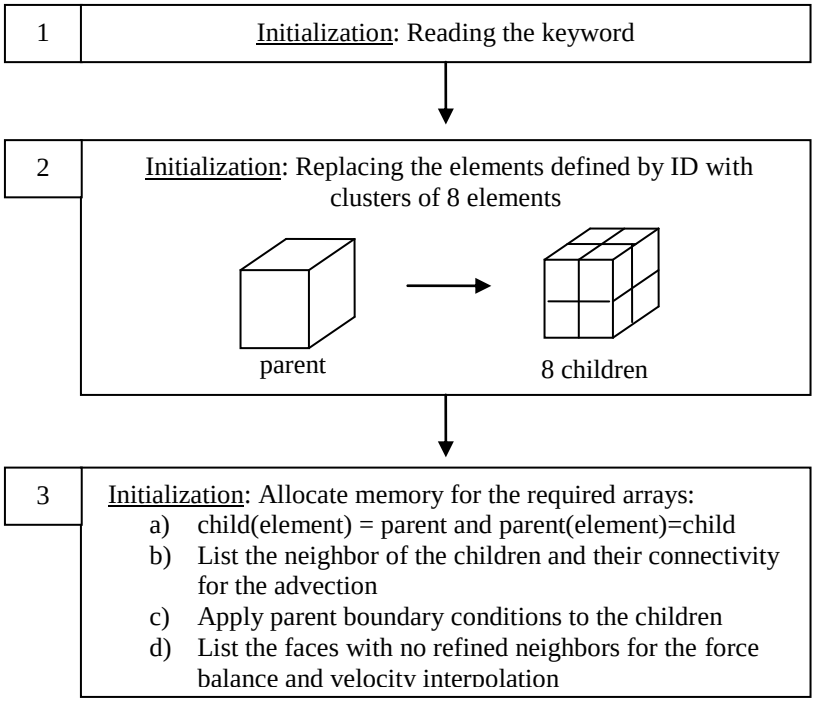


Figure 3: Flowchart of the initialization for the static refinement

In *ELEMENT_SOLID elements for the refinement are added at the end of the element list. It means that the child IDs are created by incrementally adding 1 to the largest element ID. The same logic applies to the child nodes.

If TYPE=0 or 1, a new part during the step 3 of the flowchart is created to store the parent elements while the children take the part ID of their parent. If TYPE=5, a new part is also created for the parents but the children have also a new part ID. The reason is to keep the sorting routine that groups elements of same part ID in *ELEMENT_SOLID from moving children in middle of parents and reversely. Child IDs have to remain larger than the largest element ID provided by the input deck. An extra group is also added to the multi-material group list to represent the elements that are not used, which are the parents in the case of the static refinement.
NLVL defines how refined the mesh should be. The child elements represented at the step 2 of the flowchart can be themselves refined if NLVL=2. The refinement can be reiterated as many times as the value of NLVL allows it. However NLVL for an element should not be too large comparing to the refinement level of its neighbors. A difference of 2 levels starts to be critical in terms of how the physics and expected results could be affected. A progressive refinement is advised.
After the initialization, the run begins. The dashed boxes in the flowchart of Fig.4 are the usual steps that any ALE analysis goes through whereas the plain line boxes are the steps related to *REFINE_ALE
The step 4 in the flowchart integrates the stresses over the elements to get the nodal forces. When considering refined and coarse elements side by side like on the sketch (c) of Fig.2 the nodal forces need to be balanced at the parent nodes. Each child node in faces from the list of the step3d (faces at the boundary between cells of different refinement levels) needs to add its contribution to its respective parent nodal force. The step 6 divides the forces by the nodal masses and integrates twice the resulting acceleration to get the new velocities and positions. At the end of this step, the mesh is deformed like on Fig.1. The nodes are moved to either their initial positions for an Eulerian model or user defined locations for an ALE model. The resulting fluxes are computed at the steps 8, 9 and 10. The steps 9 and 10 twist the usual Donor Cell scheme as the fluxes are not computed by the donors but the acceptors. The step 10 is the case mentioned in the introduction. A mixed cell can be neighbor of refined cells. Material fluxes to each child are computed by taking into account the interface positions and levels of refinement. Finally the step z after the advection interpolates the parent velocities at the child nodal position for the faces listed at the step 3d. After the step z, if the termination time is not reached, the next computational cycle starts again with the step 4 where the stresses and nodal forces are computed.

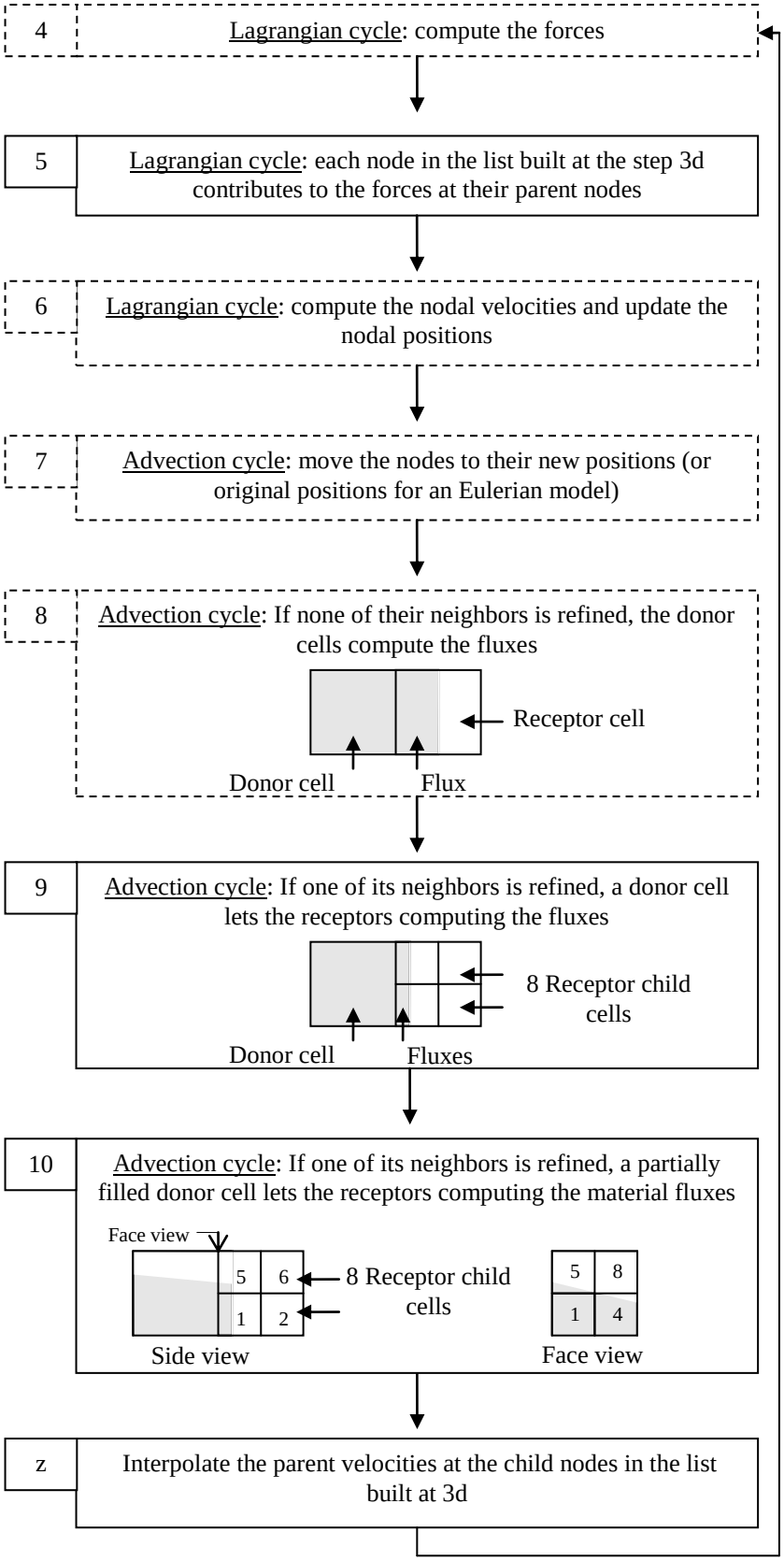


Figure 4: Flowchart of the Lagrangian and advection steps (in dashed boxes) with the *REFINE_ALE static steps (in plain line boxes)

Application

The application is a spherical pentolite charge with a radius of 2mm exploding in water. Figure 5 shows a one-level refinement region defined by a solid set. A tracer measuring the pressure history is located in this region at (x=0.745cm, y=0, z=0). The pressure is compared to a case where the mesh is fully refined (see Fig.6). Since the number of elements for a globally refined mesh (28000 solids) is greater than a model with a refined region (7700 solids), a speedup is expected. It takes 53min to run the globally refined model while it requires 16min to finish the locally refined one on a Quad-Core AMD Opteron(tm) Processor 2380.

*REFINE_ALE

Variable	ID	TYPE	NLVL					
Value	1	5	1					

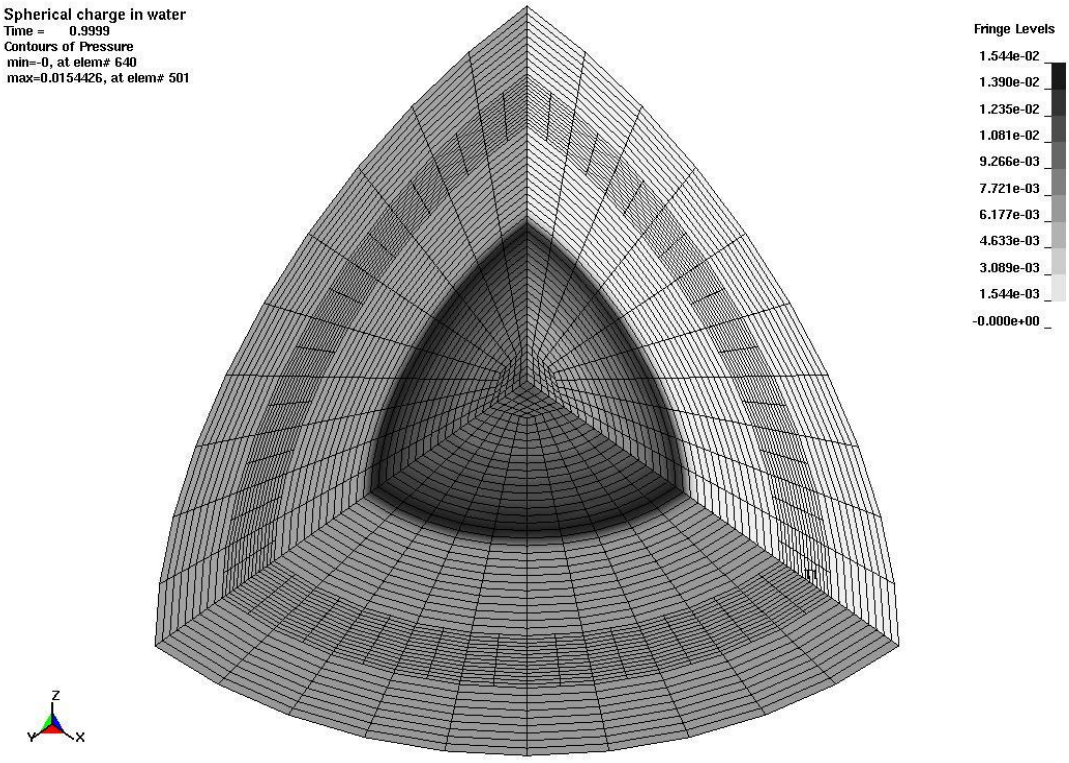


Figure 5: Pressure field 1μs after the detonation. Tracer location

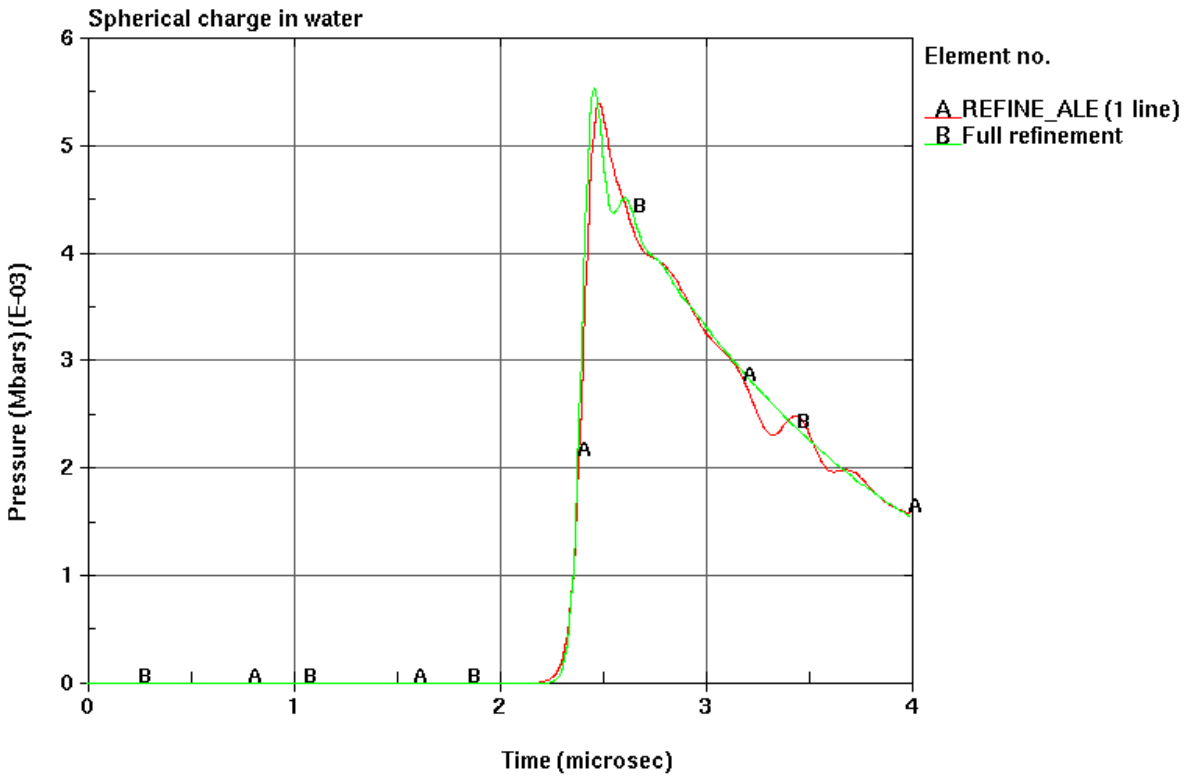


Figure 6: Pressure history from the tracer

DYNAMIC REFINEMENT

*REFINE_ALE

Variable	ID	TYPE	NLVL	MMSID			
Type	I	I	I	I			
Variable	NTOTRF	NCYCRF	CRITRF	VALRF	BEGRF	ENDRF	LAYRF
Type	I	I	I	F	F	F	I

VARIABLE	DESCRIPTION
ID	Set ID.

TYPE	Set type:: EQ.0: ALE Part Set, EQ.1: ALE Part, EQ.2: Lagrangian Part Set coupled to ALE, EQ.3: Lagrangian Part coupled to ALE, EQ.4: Shell Set coupled to ALE, EQ.5: Solid Set.
NLVL	Number of refinement levels.
MMSID	Multi-Material Set ID: LT.0: only ALE elements with all the multi-material groups listed in the set can be refined (or removed otherwise) GT.0: ALE elements with at least one of the multi-material groups can be refined (or removed)
NTOTRF	Total number of ALE elements to refine.
NCYCRF	Number of cycles between each refinement.
CRITRF	Refinement criterion: EQ.0: static refinement (as if only the 1st card is defined), EQ.1: Pressure (if pressure > VALRF), EQ.2: Relative Volume (if V/Vo < VALRF) , EQ.3: Volume Fraction (if Volume fraction > VALRF).
VALRF	Criterion value to reach for the refinement.
BEGRF	Time to begin the refinement.
ENDRF	Time to end the refinement.
LAYRF	Number of element layers to refine around an element reaching the refinement criterion.

As previously ID and TYPE defines which region of the mesh will be refined. Three more TYPEs have been added: TYPE=2, 3 and 4 define ID as being a structure of shells instead of ALE solids. These TYPEs are Fluid-Structure Interaction related. Since an ALE mesh usually deals with fluids in motion [8], it can be coupled to a structure to model a Fluid-Structure Interaction problem. If one of these TYPEs is used, only the ALE cells that enclose the structure are refined:

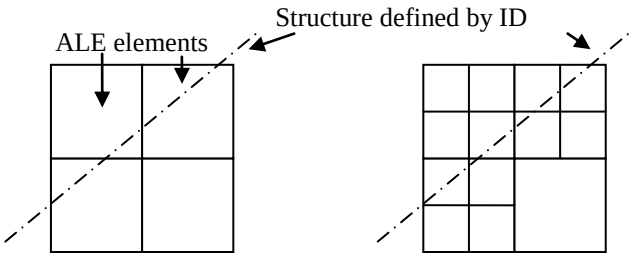


Figure 7: Sketch of a structure cutting ALE elements. Only the cut elements are refined.

The region of the mesh to refine can be also controlled by the materials in the ALE elements. If only ALE elements with specific materials needs to be refined. These materials are listed in a set defined by MMSID. The interfaces between these materials can be of special interest and only the ALE cells with the interfaces are refined. It is possible also to refine an ALE element as soon as it has one of the listed materials.

The number of ALE elements to refine and the frequency of this refinement are controlled by NTOTRF and NCYCRF. Three refinement criteria can be used:

- For CRITRF=1, if the pressure in the ALE element is larger than VALRF, the element is refined.
- For CRITRF=2, if the materials in the ALE element are compressed below a volume ratio equal to VALRF, the element is refined.
- For CRITRF=3, if the fraction of the element volume occupied by the materials defined by MMSID is larger than VALRF, the element is refined.

BEGRF and ENDRF give a time frame during which the refinement should occur.

If an element is refined, it is possible to refine the neighbor elements as well. LAYRF defines how many cells around a selected element should be refined:

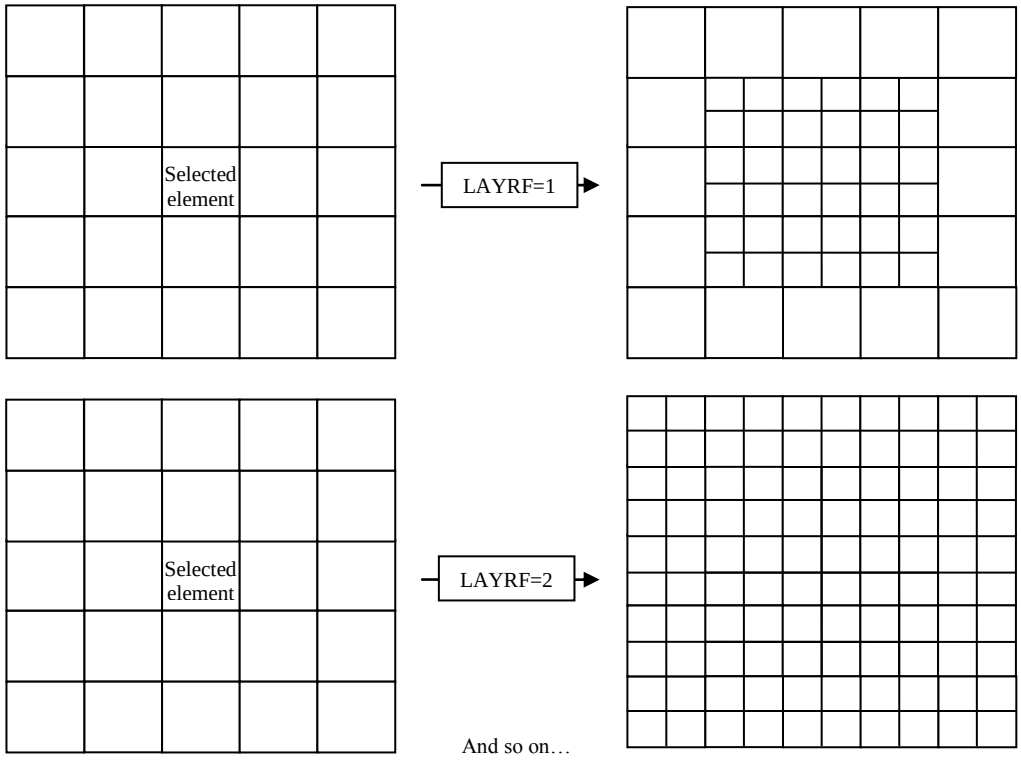


Figure 8: Refinement of a neighborhood around a selected cell with LAYRF

This feature allows refining the selected elements and their neighborhood. The number of layers of elements around the selected element is controlled by LAYRF. The purpose of this parameter is to have a larger refined region where the results are of interest and push away the refined/no refined boundary that may affect these results otherwise.
The initialization in the previous flowchart is modified. The step 2 (see Fig.9) is skipped as the refinement occurs during the run. A stock of elements that will be used for the refinement is created at the step 3: 8*NTOTRF elements are added after the largest element ID in *ELEMENT_SOLID and 27*NTOTRF nodes are added after the largest node ID in *NODE.

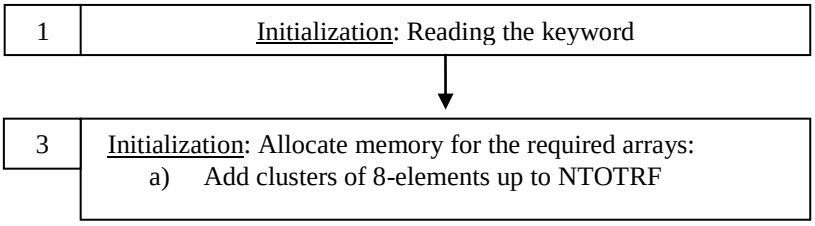


Figure 9: Flowchart of the initialization for the dynamic refinement

After the initialization, the flowchart Fig.10 is the same: it has the steps 4 to 10. Between the steps 10 and z, the refinement routines are called. The step 11 first checks if it's the right time to refine. The refinement time should be between BEGRF and ENDRF and the computational cycle should be modulo NCYCRF. Then the routine checks if some ALE elements are candidate to a refinement. The cell should be in the region defined by ID and TYPE and have the materials listed by MMSID. Finally it should meet the refinement criteria defined by CRITRF and VALRF. Two exceptions though : first if the maximum of refinement level NLVL is reached, the cell is not refined and second if the element has been selected through LAYRF, the refinement occurs even if the criteria is not satisfied. The refinement involves an update of the connectivity for momentum advection using the Half Index Shift code [7]. The neighbor list is also updated for the cell centered advection. The list of the faces at the refined/no refined boundary is modified for the velocity interpolation at the step z and force computation at the step 5. At the step 13, if the parent was a mixed element, the volume fractions of each material in each child element are computed by determining the intersection of the material interfaces and element faces. Once the volume fractions are known, the other element centered data can be calculated: material volumes, element masses, internal energies … The step 15 deals with the velocity interpolation at the child nodes. After the step 15 comes the step z from the flowchart Fig.4. The step 5 for the force contribution and step z for the velocity interpolation use a list that is built at the step 12c.

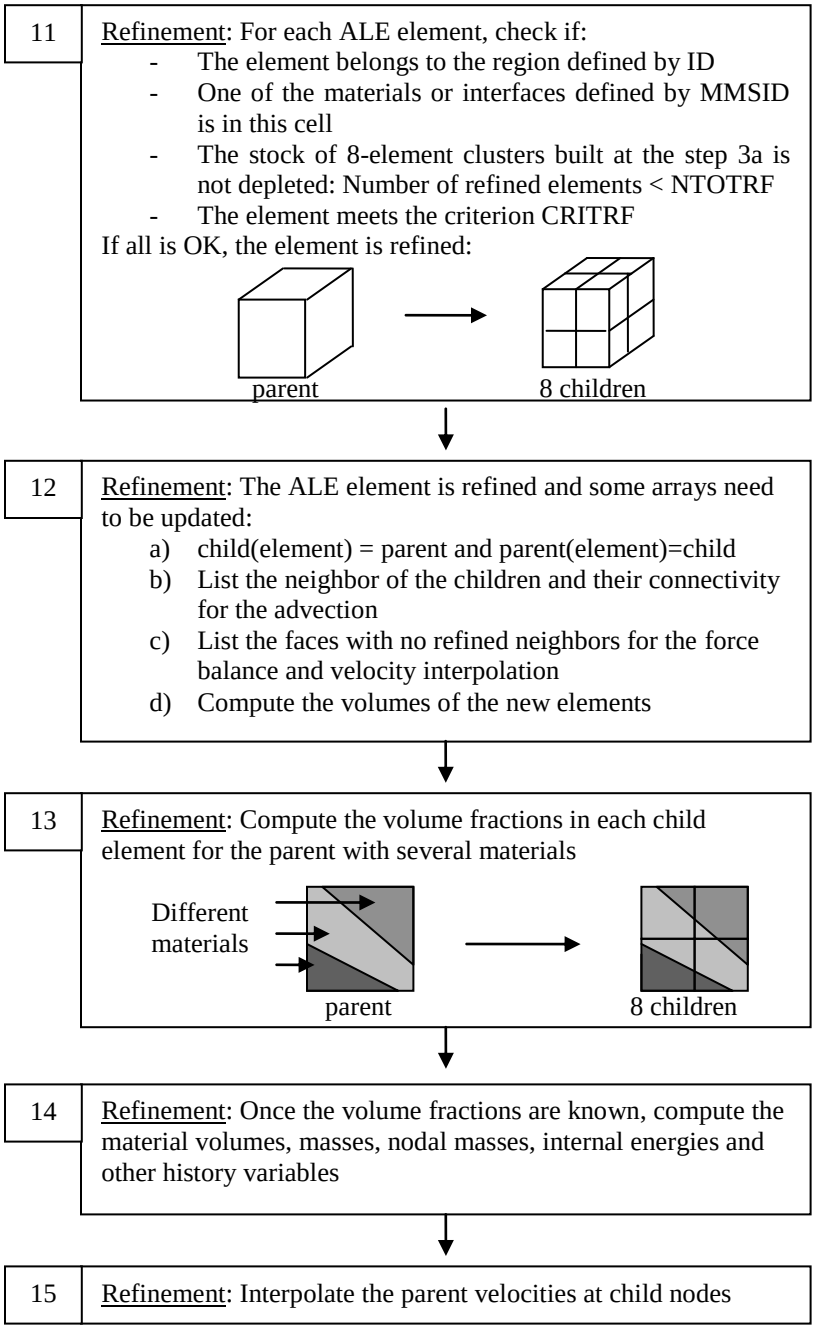


Figure 10: Flowchart of the *REFINE_ALE dynamic steps

Application

The application is the same as previously. Figures 11 and 12 show that the refinement of the part 2 follows the shock wave propagating in water (material group defined in MMSID). Every 100 cycles from 0.28microsec the part 2 is refined by picking clusters of 8 twice smaller elements in a stock of 3200 clusters. The refinement occurs if the water compression is more than 10% (current volume / initial volume < 0.9). A cubic region of 27 elements is refined as well if the center element satisfies the criteria CRITRF (the side of the cube is LAYRF elements + selected element + LAYRF elements). On Figs.11 and 12, 4 elements are refined ahead of the shock. The

code does not wait for the shock to pass these 4 layers of refined elements and reach a layer of coarse elements to refine 4 layers again. Instead an extra layer is refined each time the shock advances one element. Figure 13 shows a good agreement in the pressure history of the tracer T1 between a fully and progressive refined mesh. Contrary to the static run, the adaptive case has little more elements (29100 solids) than a fully refined model (28000 solids). It takes 56min to run the dynamically refined model and 53min for the globally refined one.

*REFINE_ALE

Variable	ID	TYPE	NLVL	MMSID			
Value	2	1	1	2			
Variable	NTOTRF	NCYCRF	CRITRF	VALRF	BEGRF	ENDRF	LAYRF
Value	3200	100	2	0.9	0.28	0.0	4

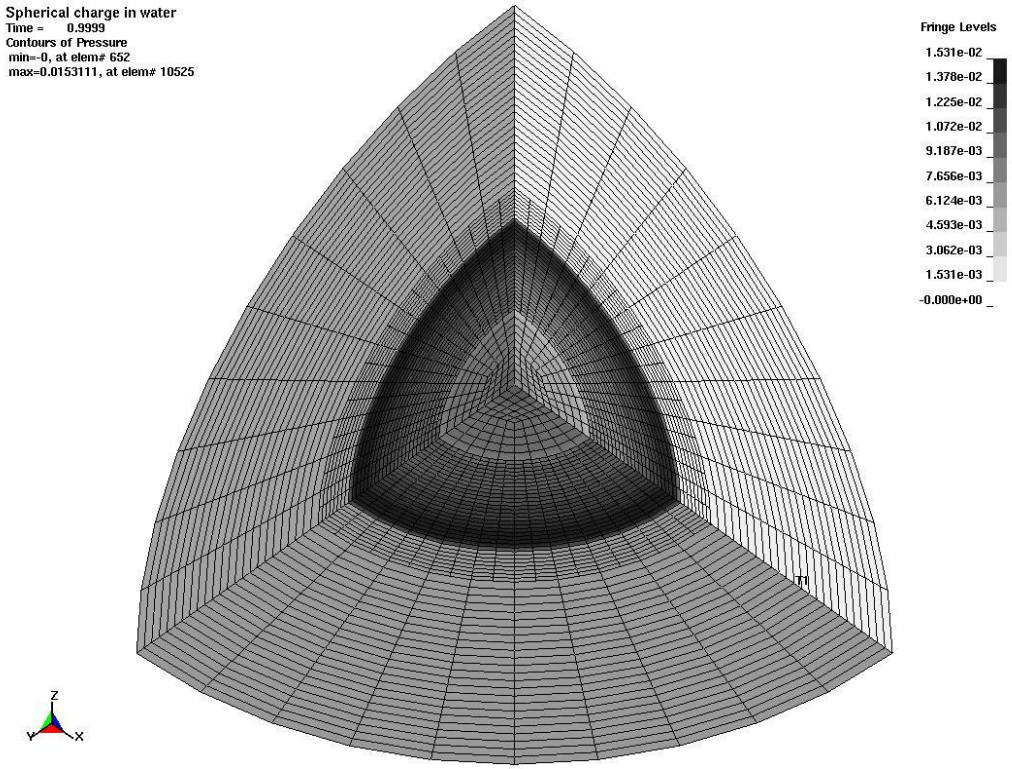


Figure 11: Pressure field and refinement 1μs after the detonation. Tracer location.

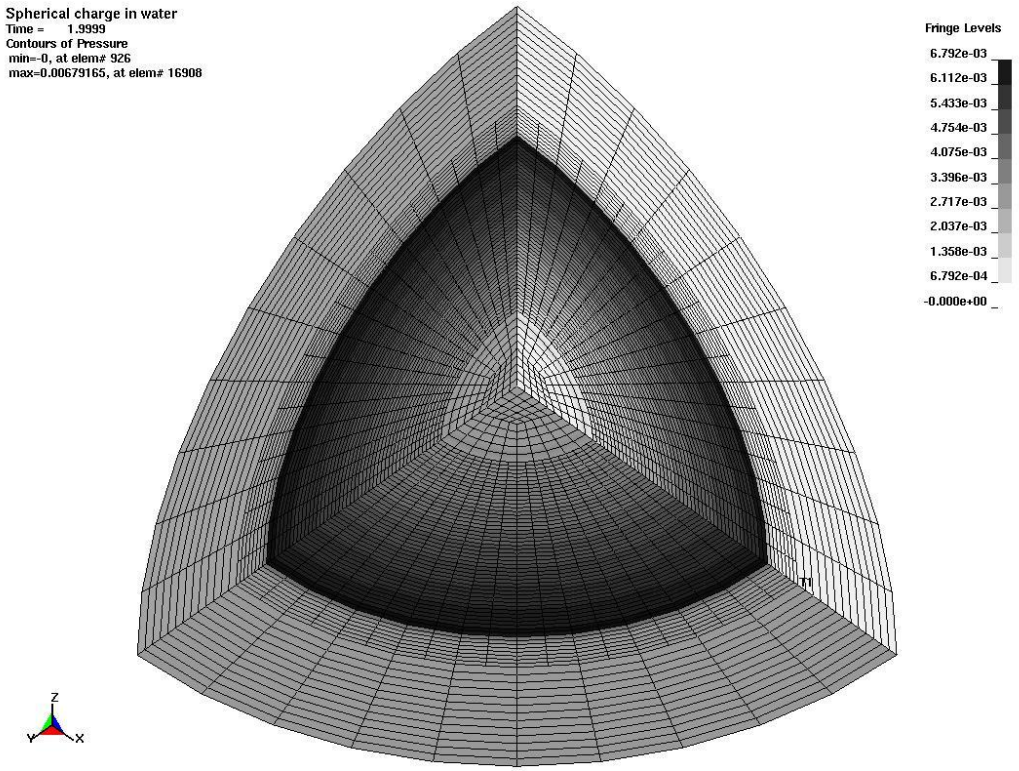


Figure 12: Pressure field and refinement 2μs after the detonation. Tracer location.

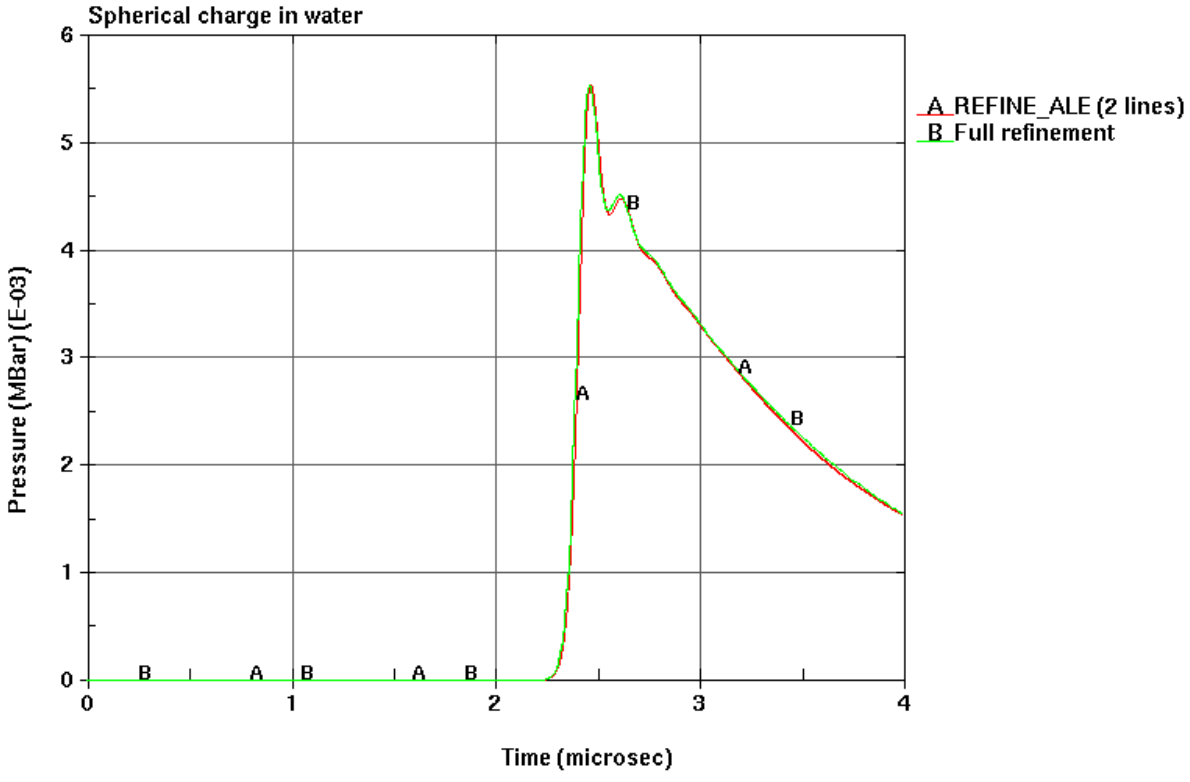


Figure 13: Pressure history from the tracer

DYNAMIC REFINEMENT WITH DELETION

*REFINE_ALE

Variable	ID	TYPE	NLVL	MMSID			
Type	I	I	I	I			
Variable	NTOTRF	NCYCRF	CRITRF	VALRF	BEGRF	ENDRF	LAYRF
Type	I	I	I	F	F	F	I
Variable	MAXRM	NCYCRM	CRITRM	VALRM	BEGRM	ENDRM	MMSRM
Type	I	I	I	F	F	F	F

VARIABLE	DESCRIPTION
MAXRM	Maximum number of child clusters to remove : LT.0: for the whole run, GT.0: every NCYCRM cycles
NCYCRM	Number of cycles between each deletion
CRITRM	Deletion criterion: EQ.0: no deletion (as if only the 1st and 2nd card are defined), EQ.1: Pressure (if pressure < VALRM), EQ.2: Relative Volume (if V/Vo > VALRM) , EQ.3: Volume Fraction (if Volume fraction < VALRM).
VALRM	Criterion value to reach in each child elements of a cluster for its deletion.
BEGRM	Time to begin the deletion.
ENDRM	Time to end the deletion.
MMSRM	Multi-Material Set ID for the deletion.

The 3rd line looks alike the 2nd one. So a copy of the 2nd line should work in most of the cases (except for the 7th column). However the deletion should be controlled with care to avoid undesired results like an element being refined and coarsened every 2 cycles. If NTOTRF defines the total number of ALE elements to be refined, a negative MAXRM is the exact

opposite and it defines a total number of child clusters to remove for the whole run. If positive, MAXRM defines an upper limit for the number of child clusters to remove every NCYCRM cycles. There are three removing criteria, opposite of the 3 matching refinement criteria. BEGRM and ENDRM give a time frame during which the deletion should occur. One purpose of the deletion is to fill up the stock of child clusters with unused refined elements to further refine the mesh if needed. It's not always easy to guess when the stock will be low. So another feature can be activated if BEGRM<0. |BEGRM| represents then a percent of NTOTRF below which the deletion should begin. If |BEGRM| = 0.1, it means that the deletion starts when 90% of the stock of clusters is used for the refinement. MMSRM defines a set of materials where the refinement can be removed. It can help to control where the cluster removal should occur. If MMSRM is not given, the material list defined by MMSID on the first line is checked instead.

The initialization steps are described on Fig.9. After the initialization, the flowchart follows the steps 4 to 10 of Fig.10. Between the steps 10 and z, the deletion routines of Fig.14 are called along with the refinement routines. The step 16 first checks if it's the right time to coarsen. The refinement removal time should be between BEGRM and ENDRM and the computational cycle should be modulo NCYCRM. Then the routine checks if some 8-child clusters can be removed. The cells should be in the region defined by ID and TYPE and have the materials listed by MMSRM or if the latter is not defined, MMSID. Finally it should meet the removal criteria defined by CRITRM and VALRM. As for the refinement there is an update of the connectivity and neighbor list for the momentum and cell centered advections. The list of the faces at the refined/no refined boundary needs also to be updated for the velocity interpolation at the step z and force computation at the step 5. At the step 18 and 19, the parent data are computed. The extensive element centered data like the material volumes and masses are summed while the intensive variables like the stresses are volume averaged. After the step 19 comes the step z from the first flowchart. The step 5 for the force contribution and step z for the velocity interpolation use a list that is updated at the step 12c and 17c.

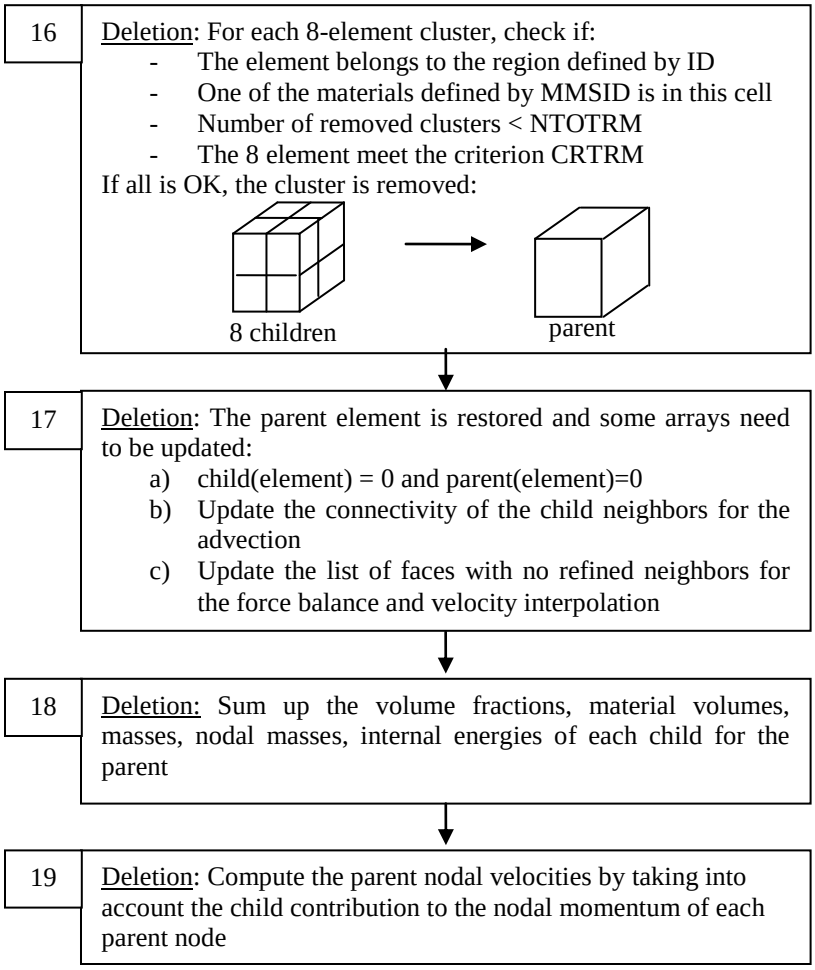


Figure 14: Flowchart of the *REFINE_ALE dynamic steps

Application

The application is the same as previously. The 2nd line is the same too except that less clusters are involved in the refinement because some of them are deleted behind the shock front and reused ahead. The 3rd line is almost a copy of the 2nd one. VALRM concerns the explosive compression (the explosive group is listed in MMSRM). After t = 1microsec (=BEGRM), the expansion of the explosive easily reaches a compression ratio above 5% and the 8 children are replaced by their parents. The newly available clusters are then used ahead of the shock front to build new refined layers. The removal can be less frequent than the refinement (NCYCRM=1000 instead of 100 for NCYCRF). Its role is to insure an available stock of clusters for the refinement. MAXRM=NTOTRF but this value is meaningless in this case. All the clusters can not be removed in one cycle. On Figs. 15 and 16, the dynamic refinement without the explosive mesh highlights the cluster removal between the explosive and the moving refined mesh. The inner limit between fine and coarse meshes shows where the material interface between explosive and water is (because only the explosive is listed in MMSRM). Figure 17 compares the tracer pressure histories between fully and dynamically refined mesh. Comparing to the previous dynamic model, there are 8*1200 elements less: 19500 solids. The time to run the job is 44min compared to 53 min for a globally refined mesh. The refining and coarsening processes take 7min as 37min should be expected to terminate the adaptive case

(53min*19500solids/28000solids approximately gives 37min). However if the shock wave needed to be propagated further, a larger mesh would be required and so the time to dynamically adapt the mesh would be much smaller than the overall computational time.

*REFINE_ALE

Variable	ID	TYPE	NLVL	MMSID			
Value	2	1	1	2			
Variable	NTOTRF	NCYCRF	CRITRF	VALRF	BEGRF	ENDRF	LAYRF
Value	2000	100	2	0.9	0.28	0.0	4
Variable	MAXRM	NCYCRM	CRITRM	VALRM	BEGRM	ENDRM	MMSRM
Value	2000	1000	2	0.95	1.0	0.0	1

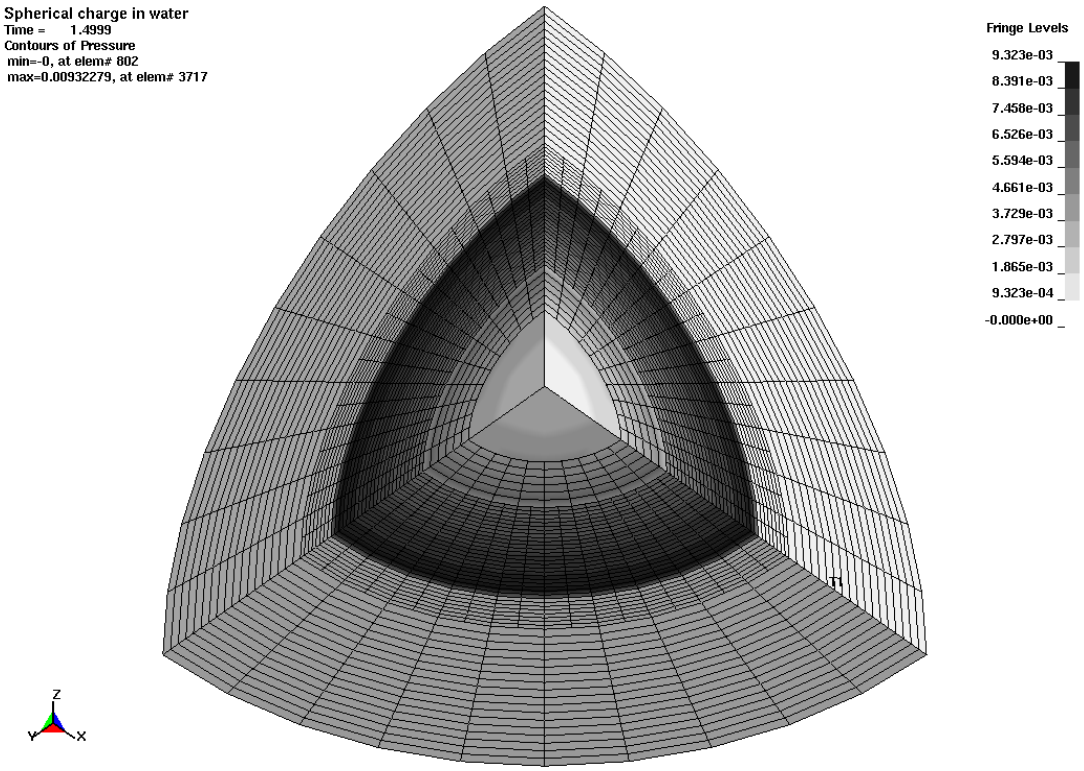


Figure 15: Pressure field and refinement 1.5μs after the detonation. Tracer location.

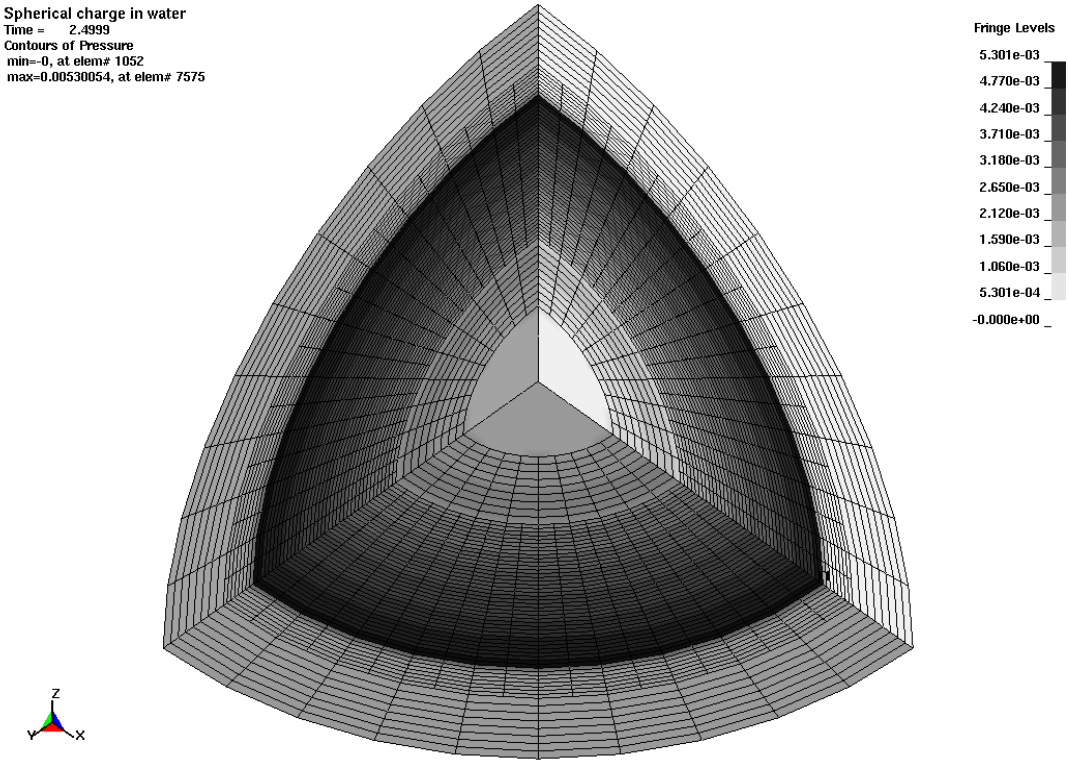


Figure 16: Pressure field and refinement 2.5μs after the detonation. Tracer location.

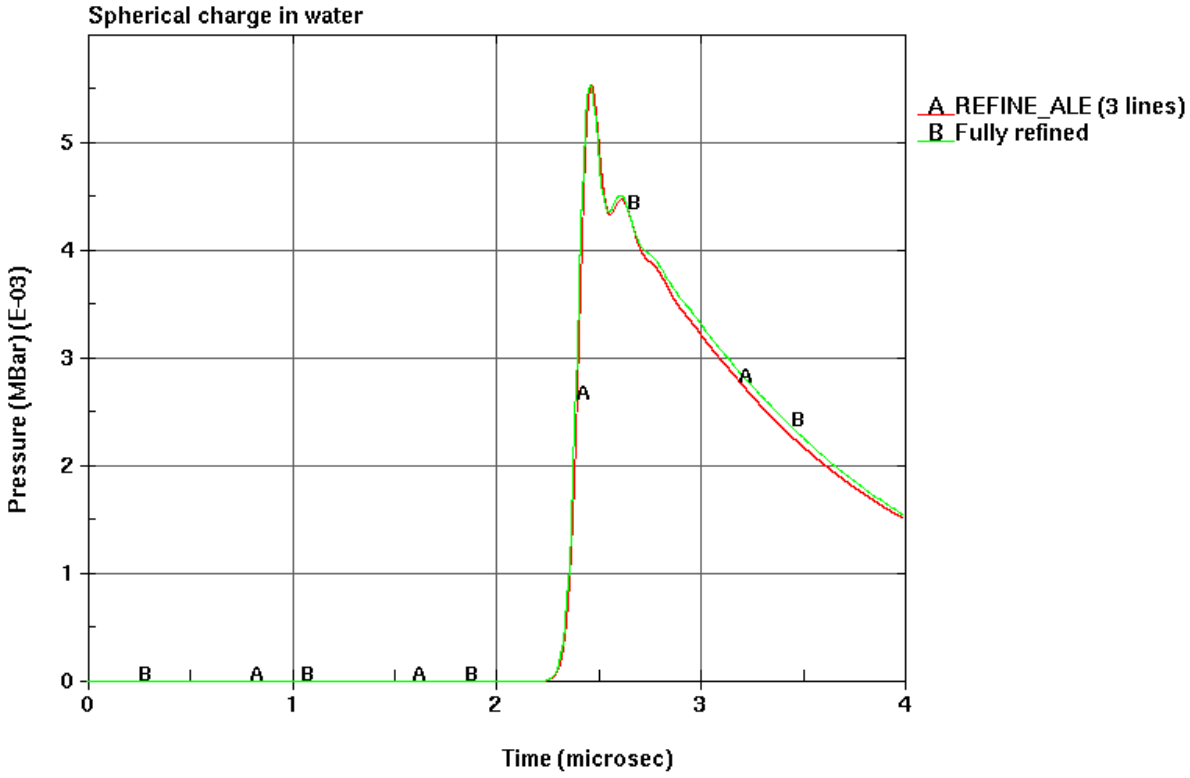


Figure 17: Pressure history from the tracer

Conclusion

A code to dynamically refine ALE meshes has been implemented in LS-DYNA®. The code is activated by a new keyword *REFINE_ALE. If only the first line of this keyword has inputs, the ALE mesh is statically refined, which means that the mesh is refined once during the initialization. If a second line is added, the mesh can be refined dynamically according parameters and criteria defined on this extra line. A third line allows removing the refinement in regions satisfying a coarsening criterion. The list of criteria to refine or coarsen the mesh can be extended. A user defined criteria could be envisaged. A MPP version will be developed along with a generalization of the approach to other solid and shell elements to create the following keywords:

- *REFINE_SOLID
- *REFINE_SHELL
- *REFINE_ALE2D

References

[1] T. Belytschko, W.K. Liu, B. Moran, "Nonlinear Finite Elements for Continua and Structures", John Wiley & Sons, LTD,2000

[2] "LS-DYNA, Keyword User's Manual," Livermore Software Technology Corporation, Livermore, 2012.

[3] A. Alia, N. Aquelet, M. Souli, L. Olovsson, "A delayed remap technique in multi-material ALE methods", European Journal of Computational Mechanics, **15**(5), pp. 465-480, 2006.

[4] D.J. Benson, "Computational methods in Lagrangian and Eulerian hydrocodes," Computer Methods in Applied Mechanics and Engineering, **99**(2), pp.235-394, 1992.

[5] T.J.R. Hughes, W.K. Liu, T.K. Zimmerman, "Lagrangian Eulerian finite element formulation for viscous flows," Computer Methods in Applied Mechanics and Engineering., **21**, pp.329-349, 1981.

[6] D. L. Youngs, "An Interface Tracking Method for a 3D Eulerian Hydrodynamics Code", Technical Report 44/92/35, AWRE, 1984.

[7] D.J. Benson, "Momentum Advection on a Staggered Mesh", Journal of Computational Physics, **100**(1), pp.143-162, 1992.

[8] M. Souli, D.J. Benson, "Arbitrary Lagrangian Eulerian and Fluid-Structure Interaction: Numerical Simulation", John Wiley & Sons, 2010.