A New 2D/3D Hydrocode

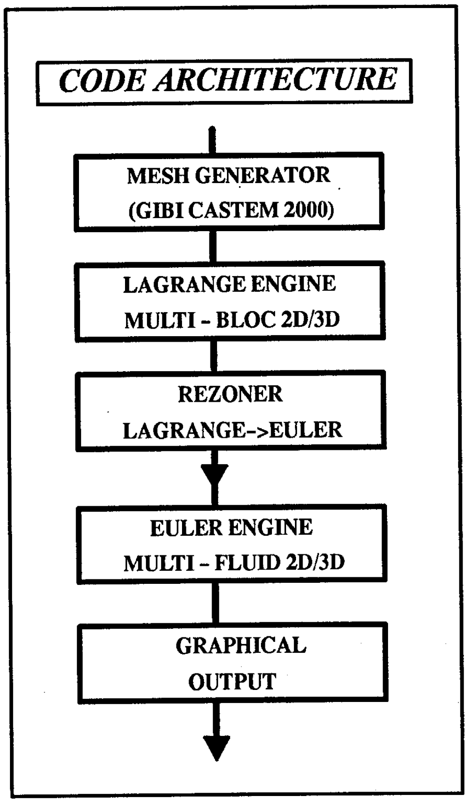
*C. Aussourd, P. Ballereau, N. Bergeret, R. David, H. Jourdren,
R. Motte,* and *F. Mougery,* CEA / DAM, Centre d' Etudes de Limeil—Valenton, Department of Applied Mathematics, 94195 Villeneuve Saint Georges, Cedex, France

ABSTRACT: *We present a 2D-3D hydrocode for compressible fluid dynamics and wave physics simulations. The software is composed of a* Finite Element *lagrangian engine, a multifluid eulerian engine and a lagrangian-eulerian rezoner used to switch from one engine to the other. Computations performed with this software often involve several million cells. The coding was first optimized for vector Cray computers. Parallel capabilities for clusters and MPP—rezoner and eulerian engine—are available for several years. The technique used is portable, based on domain decomposition method and PVM message passing. We focus here on the eulerian engine and its performance on Cray YMP and T3D.*

1 Generalities

In this paper, we won't deal with the lagrangian engine.

The software is implemented on different computers such as CRAY YMP, CRAY T3D, HP, SGl, IBM/RS6000 and SUN under UNIX. To avoid duplicated versions suitable for each computer architectures, each procedure is written only once, the slight differences being treated with the CPP (C PreProcessor) (even for FORTRAN modules). It also permits choosing generation of a

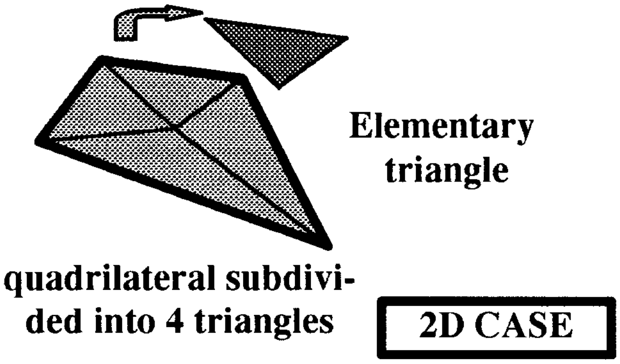


single or a double precision version only by defining a symbol and without alteration of the source code. This can be interesting on workstations using 32 bits numbers to deal with 64 bits numbers.

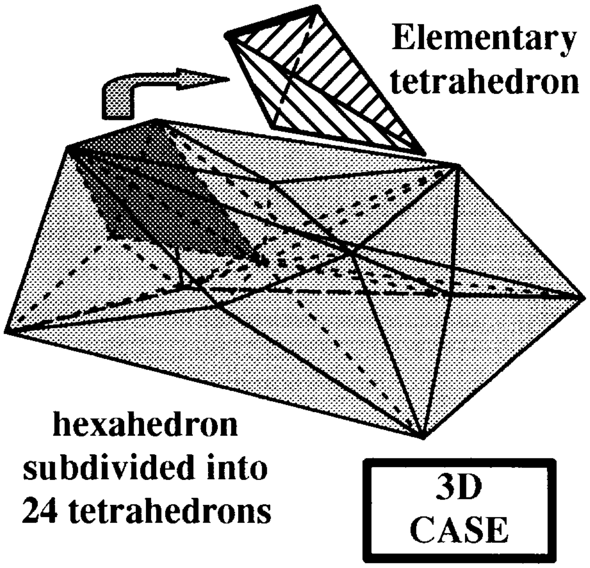
A mixed language programming strategy composed of "**ANSI-C**"to allocate and free the exact *amount of memory* needed during execution and "**FORTRAN 77**" to benefit *easy to read* and *fully vectorized* routines.

2 A Second-Order-Accurate Rezoner

This rezoner makes the projection of thermodynamic quantities from a non structured Lagrangian mesh (composed of 4 or 8 node cells possibly degenerated) to a *structured* Eulerian grid in a centered and in a non centered way. It operates on a 2D structure (plane and axisymmetric cases) on a 3D structure. Each 2D Lagrangian cell is *divided* into 4 triangular subcells.



In the 3D case, the Lagrangian cell is *divided* in a maximum of 24 tetrahedral subcells (for a hexahedron original cell).



This method ensures :

- an *exact* measurement of *volume* despite casual *warped* faces.

- a continuity between adjacent faces for nodal quantities, in and between cells.

Cells geometry and parameter variations obey to isoparametric finite elements rules:

$$x = \Sigma N_i . x_i \ , \quad y = \Sigma N_i . y_i \ , \quad z = \Sigma N_i . z_i$$

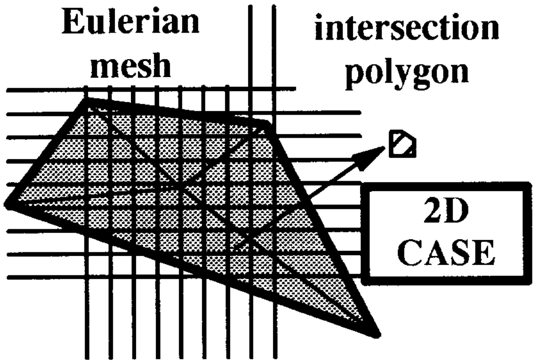
$$u = \Sigma N_i . u_i$$

$$N_i = 0.125 * (1 + \xi.\xi_i)(1 + \eta.\eta_i)(1 + \zeta.\zeta_i)$$

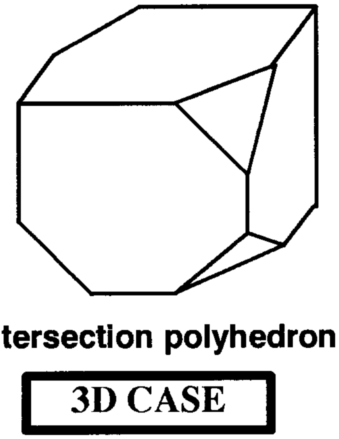
Rezoning of a conservative rezoning of centered variables thanks to an estimation of **gradients** and a **slope** limitation in 1^{st} and 2^{nd} order to limit discontinuities between cells. In addition, we assume that the quantities to be remapped have ***linear*** variations between node of a subcell.

We determine :

➔ **polygons** in 2D having from 3 to 7 vertices



➔ **polyhedrons** in 3D having from 4 to 10 subfacets (of 3 to 9 vertices) which are the intersection between a Lagrangian subcell (triangle or tetrahedron) and the Eulerian cells that it overlaps.



Then, we compute the **contribution** in each polygon (or polyhedron in 3D) by integrating the quantity to be projected along the boundary of the considered polygon (or its subfacets contours in 3D). Finally, the result of the projection is the suom of these separate contributions.

3 EULERIAN ENGINE

3.1. Generalities

Euler is a 2D/3D cartesian, explicit, multi-material, hydrodynamic code, allowing high explosive and material strength models. The numerical scheme is divided in two phases: Lagrangian + Advection.

Two methods may be used for the Lagrangian phase :

▨ The first one, based on Godunov's scheme, is restricted to the case of pure hydrodynamic and high explosive simulations.

▨ The second one is a multi-dimensional extension of Collela's and Woodward's BBC formulation and is well suited to constitutive models.

3.2. Advection Scheme

A remapping using an operator splitting technique over the directions completes the Lagrangian part of the method. The fluxes of the main quantities are calculated through the determination of the volume fluxes across each interface for all the species present in the cell. Currently, a SLIC (with multi-dimensional geometric correction) and a YOUNGS algorithms determine the advection volume and a Van Leer monotonic limiting scheme updates the variables. This procedure is, for the two Lagrangian schemes describe above, conservative with remapped data and authorizes any combination with no extra CPU, memory space or coding costs.

The choice of the remapping dependence and the order of precision of the advection phase, can be separately specified for each quantity of each material in the input data. In order to prevent non physical internal energy undershoot, a particular treatment is done for momenta and the total energy as part of a BBC-like scheme.

First, the internal energy advected in a conservative way. Then middle cell fluxes are interpolated at first order from the interface values to allow momenta and the remapping of kinetic energy. Finally, the difference between advected kinetic energy and kinetic energy calculated from advected momenta is substracted from the internal energy, which preserves total energy.

3.3. Details about Lagrangian phase schemes

3.3.1. Godunov based method (GAD)

We refer to a cell centered scheme explicitly formulated in a conservative way which consists in the resolution of a Riemann's problem on each interface between cells. Our simplified version uses a *sonic approximation* of the waves propagation leading to a sufficient precision in most configurations.The first order accuracy of this method can be improved to the second order by using a limited anti-diffusive method, which is an extension of TVD schemes to systems of conservative laws. The main feature of this approach resides in achieving an upper order of precision without any intermediate call to EOS.

3.3.1.1. BBC based formulation Woodward-Collela ..) :

This version is based on a multi-dimensional extension of Woodward and Collela's formulation (*The Numerical Simulation of 2D Fluid Flow with Strong Shocks - JCP 54 (1984),115-173*) where strength effects reduce to source terms introduction

$$\rho\frac{\partial\vec{u}}{\partial t} = \rho\frac{\partial\vec{u}}{\partial t}\mathbf{1} + \rho\frac{\partial\vec{u}}{\partial t}\mathbf{2}$$

$$\frac{\partial\epsilon}{\partial t} = -\rho\frac{\partial V}{\partial t} + \textit{Deviatoric contribution}$$

$$\rho\frac{\partial\vec{u}}{\partial t}\mathbf{1} \quad \textit{stress diagonal impulse}$$

$$\rho\frac{\partial\vec{u}}{\partial t}\mathbf{2} \quad \textit{stress deviatoric impulse}$$

+EOS + stress advancement

Two models are available yet at the moment:

- elastic-perfectly plastic material strength model.

- SCG Plasticity is realized using Von Mises yield condition.

3.4. Available modules

Somme of the existing modules are listed below:

➔ Two predictive detonation front position methods (the first is based on Huygens's Law, the second on Halmiton-Jacobi equation)

➔ a High Explosive burning models subset (Forest Fire and Johnson-Tang-Forest kinetics).

3.5. Architecture/Memory Allocation

The upper layer of the Eulerian library, coded in ANSI-C, is devoted to memory allocation and task management (50 % of the "eulerian" sources). The ***memory structure*** is "object-oriented". According to the type of the object - empty, hydrodynamic, explosive, elasto-plastic - the exact memory is allocated as a function of the number of thermo-mechanic quantities involved, and the exact number of cells occupied by the object at the current time-step. The ***task manager*** treats each object sequentially, according to its type. Consequently, nothing has to be done for the empty material, and deviatoric stresses are not updated for a hydrodynamic material. The lower layer of the eulerian engine, coded in FORTRAN 77, is a succession of vectorized *do loops* performing the numerical calculations necessary for the object.

3.6. Calculation option

- Active Area

Usually, when the simulation begins, only a few cells are reached by a perturbation (for example: when dealing with a shockwave) and the computation of the other cells is useless. This phenomenon involves the introduction of an active area concept. At each cycle we create a "pseudo area", which contains only the active cells. This technique allows us to improve the calculation speed.

- Multicycle

This method allows the code to perform several Lagrangian steps before performing an advection one. This is interesting when the sound speed is much greater than the material speed.

3.7. Windowing

3.7.1. Generalities

The memory requirement associated with 3D simulations needs the use of disk peripherals in addition to the use of central memory.

To reduce the overhead of disk transfers during a run, some solutions fit to all the computers while others can be chosen to match **CRAY YMP** supercomputers optimization requirements.

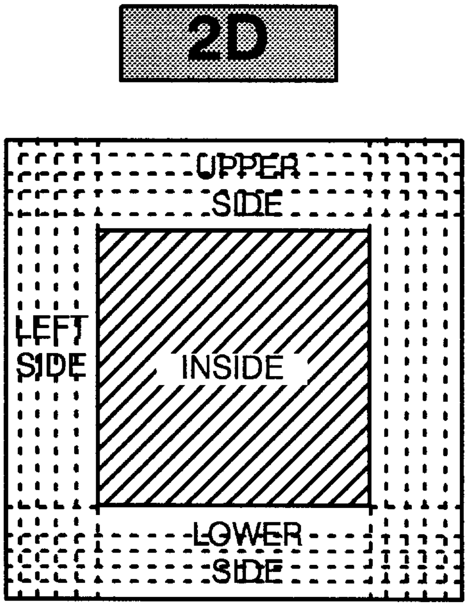
To reach these goals we use two main techniques:

- First : to isolate the necessary subset of information to be saved strongly correlated with the programming structure.

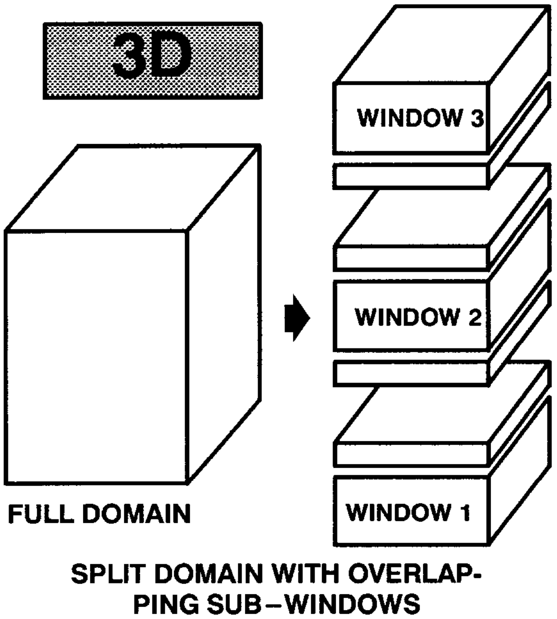
- Second : to pack information into large records tables because it is faster to write long vectors. However an asymptotic limit is rapidly reached when the lengths of the tables exceed a few thousands words.

3.7.2. Implementation

The slices consists of adjacent planes in 3D (resp. portion of plane in 2D).



A sample 2D window showing both inner and overlapping zones



3D windowing (case with 3 windows)

Although it is a classical concept, the originality of the implementation lies in two points:

- ▨ First, the input/output action is very simple.
- ▨ the treatment is slightly the same for a split or an not split domain

The generality of this approach allows the following extensions.

3.8. Concurrent Computation of Windows (Parallelism)

The computation can be split, between a group of computers (a Virtual Machine) and/or between several processors on the same host (Cray T3D , CRAY YMP ...), using PVM library Each window is associated with a processor and its memory. Overlapping information are transmitted via the network. This leads to a performance increase (no disk access and several processors working in parallel).

4 Conclusion

The nature of the hydrocodynamic code is based on very simple programming concepts. However the numerical concepts however are much more complex. They include different physical models - such as various equation of states, detonic models, strength models etc ...

It is also an industrial code and not research one. It is used by the French Defense industry. It is much more reliable than a research code and the man/machine interface has been more highly developed.

What are the limits of this code ? it performs certain simulations very well. For example the picture obtained from calculating shape charge are identical to the picture obtained from X-ray representation of an experiment performed with this shape charge. However, the results of this code are les convincing when doing the penetration simulation.

In summary the main advantage is the code's performance in terms of the number of cells computed and the calculation speed. This performance has been achieved by an important work :

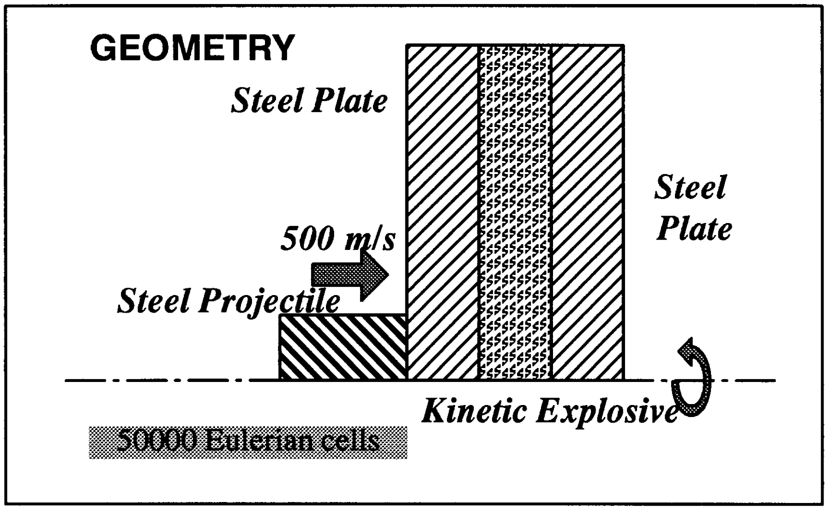
- the coding methods
- the choice of algorithms
- the suppression of useless computation

This last point is very important. We do not calculate empty or inactive cells. We begin calculations on a object when the shock wave arrives. All of these points reduce the necessary number of calculation.

Curent work adresses new numerical techniques such as Adaptive Mesh Refinement (AMR) will be the subject of another paper.

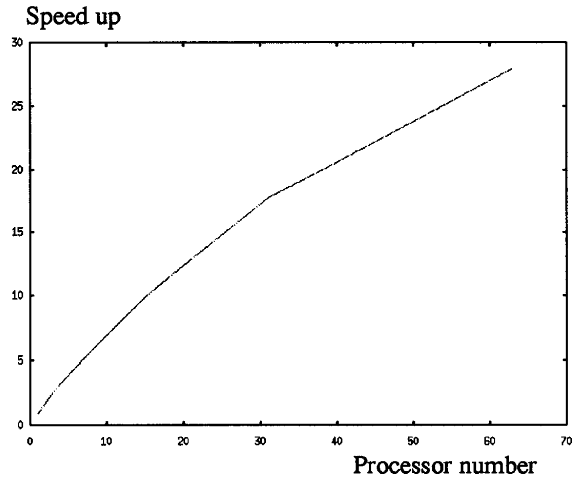
EULERIAN TEST CASES :

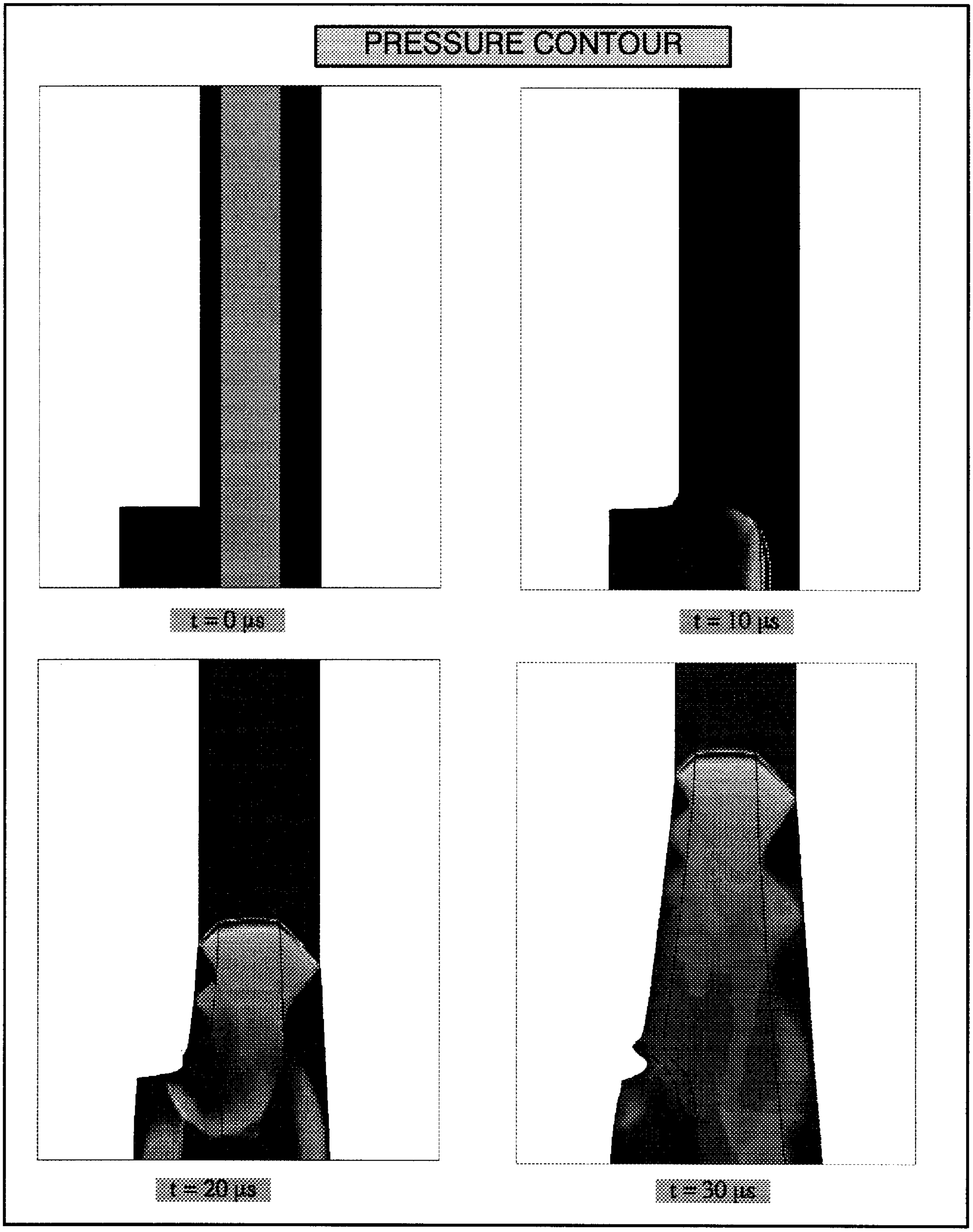
ARMOR ANTI–ARMOR CALCULATION



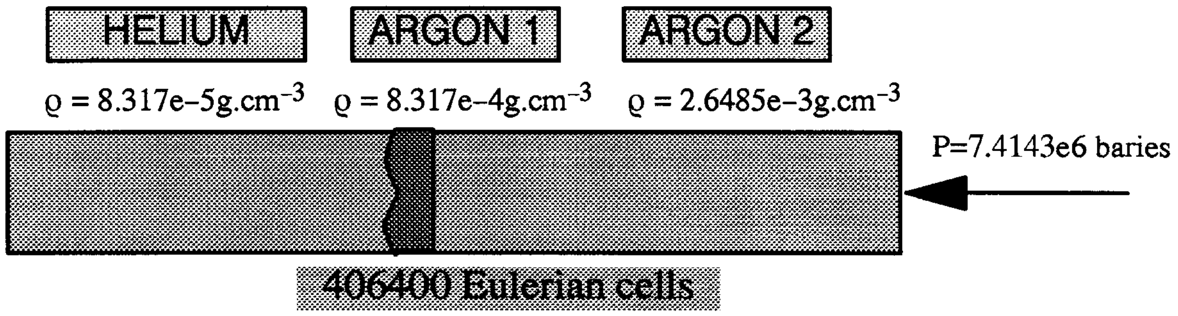
processor number	µs/cell/cycle	speed up	Performance
1	129.44	1	100 %
3+1(1)	48.62	2.66	89 %
7+1	24.7	5.24	75 %
15+1	12.97	9.98	67 %
63+1	4.62	28.01	44 %
63+1 Active Area & Multicycle	2.57	-	-

Cray YMP 1 proc.	13.91
Cray YMP 1 proc. Active Area	8.72
Cray YMP Active Area / Multicycle	6.59
Cray YMP Active Area / Multicycle2	5.21




PRESSURE CONTOUR
t = 0 μs
t = 10 μs
t = 20 μs
t = 30 μs

RAYLEIGH – TAYLOR INSTABILITIES GROWTH



HELIUM

ARGON 1

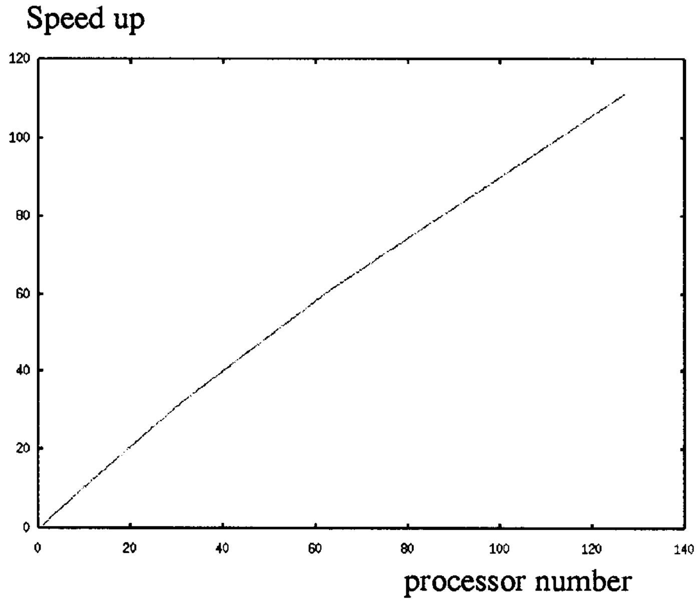
ARGON 2

$\varrho = 8.317e\text{-}5 g.cm^{-3}$ $\varrho = 8.317e\text{-}4 g.cm^{-3}$ $\varrho = 2.6485e\text{-}3 g.cm^{-3}$

P=7.4143e6 baries

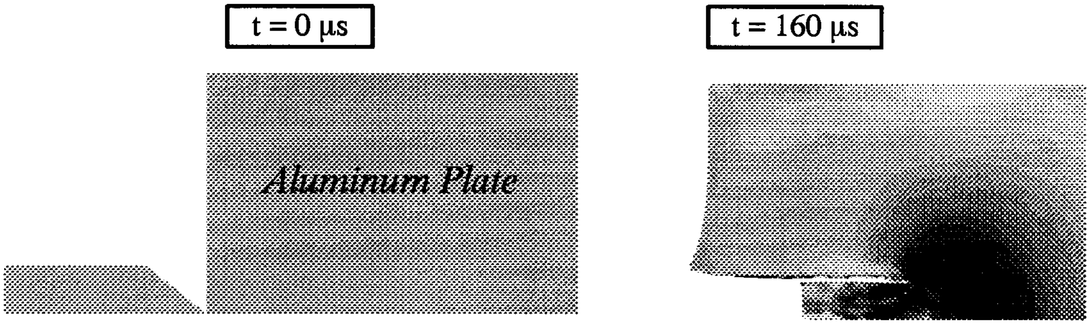
406400 Eulerian cells

Processor number	µs/cell/cycle	speed up	Performance
1	135.88	1	100 %
3+1	40.94	3.31	110 %
7+1	18.09	7.51	107 %
15+1	8.53	15.93	106 %
31+1	4.25	31.97	103 %
63+1	2.22	61.2	97 %
127+1	1.22	111.4	88 %
crayymp 1 proc	10.8		-



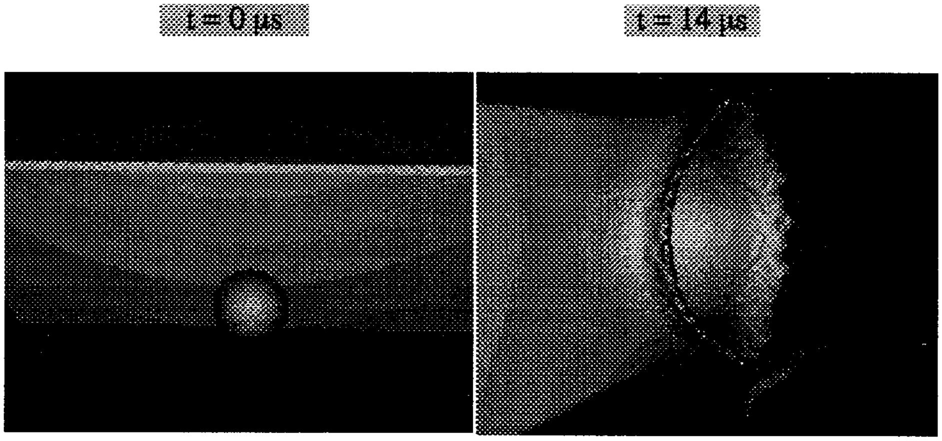
TUNGSTEN PENETRATOR

PRESSURE CONTOUR



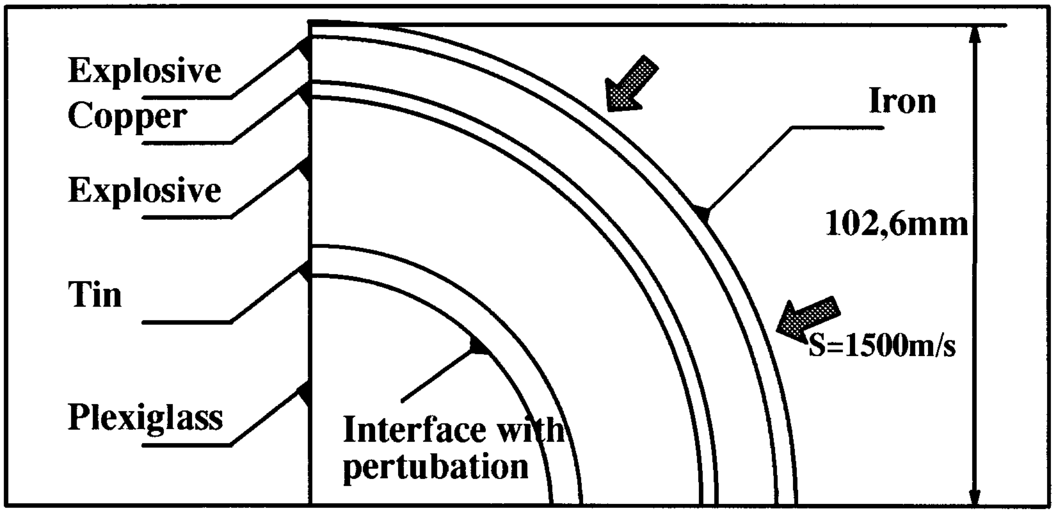
ELASTIC-PLASTIC BBC
$V_{penetrator}$ = 600 m/s – 28, 350 cells

IMPACT OF A STEEL PELLET AGAINST A STEEL WALL.



HYDRODYNAMIC
V_{pellet} = 7000 m/s
1, 500, 000 cells

RICHTMYER–MESHKOV
INSTABILITIES GROWTH : 2D CYLINDRICAL IMPLOSION



PLEXI–TIN INTERFACE : MULTI–MODE SINUSOIDAL DEFAULT SURFACE :
R(Θ) = 0.005cos48Θ + 0,006cos96Θ + 0,005cos288Θ + 0,008cos480Θ + 4,6

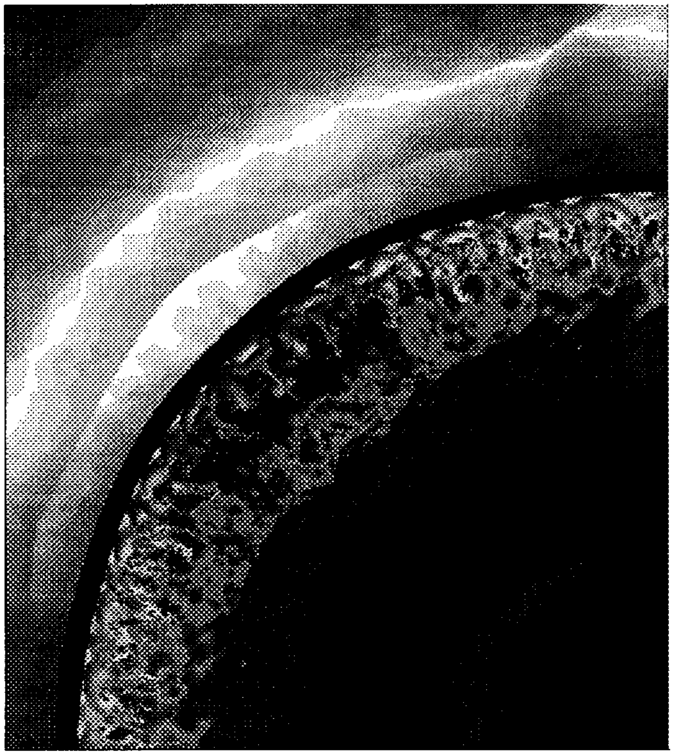
7 800 000 CELLS, 35 000 CYCLES, 18 MICRON CELLS

SOFTWARE	Grind times μs/ cell/cycle	Mflops	Computational time
Previous code 1 proc. Cray YMP	250 (*)	15 (*)	18 months (*)
1 proc. Cray YMP	12,9	77	650 h (*)
1 proc. Cray YMP (Multicycle)	8,70	76	440 h (*)
128 proc. Cray T3D (Multicycle)	1,20	550 (*)	60 h

(*) Estimate

Pressure Field

t = 15 us Explosive/Tin/Plexiglass



Focus