9th International Symposium on Interaction of the Effects of Munitions with Structures
Berlin, 3 - 7 May 1999

Advanced Eulerian Techniques for the Numerical Simulation of Impact and Penetration using AUTODYN-3D

Gabi Luttwak[1]
Rafael, Box 2250, Haifa 31021, Israel

Malcolm S. Cowler
Century Dynamics Inc., 2333 San Ramon Valley Blvd, Suite 185, San Ramon, CA 94583.
Tel: 925-552-1600, Fax: 925-552-1609, e-mail: msc@autodyn.com

Abstract

This paper describes two new capabilities that have been implemented in the non-linear dynamics program AUTODYN-3D.

- A second order Euler processor using a Godunov solver, three dimensional interface reconstruction, and dynamic memory management.
- A Lagrange to Euler remap.

The numerical simulation of the formation and penetration of an explosively formed projectile is used to demonstrate these new capabilities.

1.0 Introduction

Impacts and penetrations typically produce large material flow and deformation. In a Lagrangian calculation the grid moves with the material velocity. This can cause severe grid distortions of the Lagrangian grid. In many cases Eulerian coordinates are a natural way to handle such problems. In the past, however, several drawbacks have prevented the efficient utilization of the multi-material Eulerian technique:

- The advection of material through an Euler grid can generate numerical diffusion, which artificially smoothes gradients in the calculation.
- It is difficult to adequately follow material interfaces, which cut through the Euler cells, especially if the material flow is in a direction diagonal to the grid.
- Most importantly, an Euler grid must cover regions not only where material initially exists, but also regions into which material is expected to flow during a calculation. For three-dimensional calculations the resulting memory requirements are usually so prohibitive, that only coarse-grid calculations can be computed.

A new second order Euler processor has been implemented in the non-linear dynamic analysis program AUTODYN-3D. This processor employs novel techniques to remedy the above deficiencies:

- A second order Godunov scheme [1] is used to decrease numerical diffusion.
- Multi-material interfaces are followed using a fast and robust algorithm similar to Young's technique [2].
- A unique dynamic memory-management scheme allows almost no memory to be used for empty (void) cells. Thus, an Eulerian calculation utilizes only slightly more memory resources than an equivalent Lagrangian calculation.

Despite problems with grid distortions, the Lagrangian approach still has several advantages over Euler, particularly in modeling complex or thin material layers and material strength and failure. Therefore, it is often preferable to perform the early stages of the calculation using a Lagrangian grid, while grid distortions are reasonable. An option

[1] Work performed at Century Dynamics, while on sabbatical leave from Rafael

has been developed for AUTODYN-3D, which allows such calculations to be remapped into a multi-material Euler grid (conserving mass, momentum and energy) when distortions become too severe.

An example illustrates these capabilities. The formation of an Explosively Formed Projectile (EFP) is computed using Lagrangian coordinates. The results of this calculation are then remapped into an Eulerian grid, and the impact of the EFP into a target is computed.

2.0 Theory

To obtain the differential equations of motion, the conservation laws of mass momentum and energy are applied to an arbitrary small closed control volume, V:

$$\frac{d}{dt}\int_V \rho\, dv = -\int_{\partial V} \rho\left(\vec{u}-\vec{u}_g\right)\bullet d\vec{s}$$

$$\frac{d}{dt}\int_V \rho u\, dv = -\int_{\partial V} \rho u\left(\vec{u}-\vec{u}_g\right)\bullet d\vec{s} + \int_{\partial V} \underline{\underline{s}}\bullet d\vec{s}$$

$$\frac{d}{dt}\int_V \rho e_T\, dv = -\int_{\partial V} \rho u e_T\left(\vec{u}-\vec{u}_g\right)\bullet d\vec{s} + \int_{\partial V} \left(\underline{\underline{s}}\bullet\vec{u}\right)\bullet d\vec{s}$$

The control volume may be moving with an arbitrary speed, $\mathbf{u_g}$. In Eulerian coordinates, the grid is fixed, $\mathbf{u_g}$=0. In Lagrangian coordinates, the grid moves with the material velocity, $\mathbf{u_g}$= **u** . The total energy e_T is expressed as the sum of internal energy e and kinetic energy . The stress tensor is decomposed into a hydrostatic pressure p, and stress deviators $\underline{\mathbf{s}}$. The pressure is obtained from the equation of state, while the stress deviators are derived from material constitutive relations:

$$e_T = e + u^2/2 \quad ; \quad p = p(\rho, e) \quad ; \quad \underline{\underline{s}} = -p\underline{\underline{I}} + \underline{\underline{s}}$$

For compressible flow, the resulting equations of motions are hyperbolic in character. During the solution of these equations, shocks and contact discontinuities can be generated, across which the differential equations do not hold. In such situations, applying the conservation equations across the discontinuity yields the Rankine-Hugoniot relations.

The difference equations are obtained by applying the integral conservation equations directly over the computational cell. In this way, the resulting equations are locally conservative. However, like the differential equations, these difference equations are not valid at discontinuities. One solution to this problem is to explicitly follow each shock present in a calculation and solve the Rankine-Hugoniot relations across the shock front. This technique is called "shock fitting". Unfortunately, the complex formations and interactions of shocks that typically develop in multi-dimensional solutions make this technique far too complicated and impractical to implement for general-purpose calculations. An alternative, more practical approach, is to modify the difference equations to capture the shocks. The most popular of these "shock capture" techniques is Von Neumann's "Pseudoviscosity" [3]. With this technique, a term is added to the differential equations, which causes gradients at shocks to be spread over 2-3 computational cells. This method is usually used in first order solutions as results tend to be somewhat noisy near discontinuities, reducing the accuracy of calculations. In the second order Godunov scheme, an algorithm that correctly captures shocks without using artificial damping terms is used.

2.1 The second order Godunov scheme

In a first order difference scheme, variables are assumed to be constant within a computational cell. With this assumption, there will commonly be a discontinuity at the boundary between two neighboring cells. In gas-dynamics, the solution at a discontinuity between two constant states is known as the Riemann problem. Godunov [4] proposed that the Riemann problem be solved to obtain face values of variables. The resulting face values are then used to integrate the conservation equations over the control volume. In first order solutions, this technique diffuses gradients more than the Von-Neumann pseudo-viscosity method. To overcome this problem, Van Leer [1] extended Godunov's method to second order. This technique has been implemented in AUTODYN-3D. To obtain second order accuracy, each variable is assumed to vary linearly over a cell. This requires carrying not only a cell-centered value for each variable, but also its gradient. The gradient is used to extrapolate face values of variables from the cell-centered values. In general, the extrapolated face values are different on each side of a face, so an approximate Riemann problem is solved to advance the solution at each face to the middle of the time step.

To maintain stability, the advection terms must be diffusive up to the truncation error. Consequently a second order transport scheme is much less diffusive than the equivalent first order solution. However, in the neighborhood of shocks, where the gradients are infinite, a higher order scheme may perform worse. Such schemes can also artificially create new extrema, or enhance existing extrema in the distribution of the variables. To prevent such non-monotonic behavior, Van Leer proposed a slope limiting technique, whereby the density gradient (slope) in a cell is limited, so that the extrapolated values of the density at the neighboring cell centers remain bounded by the values of the densities in the neighboring cells.

The second order Godunov scheme implemented in AUTODYN-3D closely follows the work of Hancock [5]. Thus, gradients are set to zero at shocks and discontinuities, and near free surfaces and material interfaces (i.e. in these areas, the solution reverts to first order). The conservation equations are integrated over the computational cells in three-dimensional space (rather than carrying out one-dimensional direction splitting which, while technically simpler and formally second order accurate, is trouble prone). Hancock's prescriptions for slope-limiting are used. For elastic-plastic flow, the slopes of the stress deviators are not computed, so their advection is only first order accurate (for typical problems involving high-speed impact and detonation, where the stress deviators account for less than 10% of the pressure, this approach can be justified).

The Riemann solver uses a two-shock approximation. Since the shock Hugoniot for weak waves is tangent to the isentrope, this approximation holds reasonably well everywhere, except near strong rarefactions. In compression, and locally, over the time step, a linear relationship is assumed between the shock speed, $\mathbf{U_s}$, and the material velocity, **u**:

$$U_s = c_0 + su$$

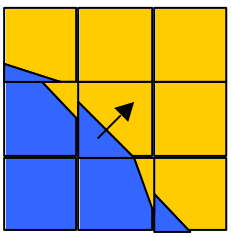
For most solids, such a relation is experimentally observed over a large range of pressures. For expansions, $\mathbf{U_s}$ is taken to be equal to the sound speed c_0. This approach does not limit the equation of state used. The role of the Riemann solver is somewhat similar to the artificial viscosity used in first order codes.

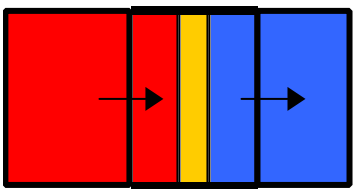
2.2 Multi-material interface tracking

In an Eulerian calculation the grid remains fixed in space, and materials flow through it. This means that material interfaces may cut through cells. One of the most difficult tasks in a multi-material Eulerian scheme is to adequately follow these material interfaces. A number of different approaches have been used to address this problem. Some solutions (e.g. Particle In Cell (PIC) [6]) use Lagrangian particles to track material interfaces. These particles move with the material and consequently suffer from distortion problems similar to those characteristic of Lagrangian grids. A recent alternative is the Level Set [7] method in which the interface is located at some constant value of a function defined at the cell vertices. Typically the function may be chosen as the distance to the interface. By advancing the values of this function in time, the interface position can be determined. Unlike Lagrange particle methods, the Level Set method can deal easily with changes in the topology of the interfaces. However, the location of interfaces obtained using either of these two approaches may not be consistent with the partial volume and mass present in a cell, as computed by the advection terms of the conservation equations. Because of these limitations, the following Volume of Fluid method (VOF) [15] has been implemented in AUTODYN.

Material interface locations are needed to compute the fraction of each material that is transported across a cell face during the advective phase. VOF methods determine the location of any material interfaces within a donor cell by analyzing the volume fractions of materials in neighboring cells, essentially performing an interface reconstruction.

Simple preferential transport [8] and donor-acceptor schemes use only the two cells adjacent to the face (i.e. a one-dimensional reconstruction) to determine the location of material interfaces. SLIC (Simple Line Interface Calculation)[9] is a further improvement of this approach in which the order of fluxing materials is chosen to be dependent on their presence in three cells (the donor cell, the acceptor cell and the cell behind the donor cell). Robust and efficient, SLIC gives good results if the flow is normal to the grid (material interfaces are always constructed normal to the grid), but can cause large errors when the flow is diagonal through the grid.





This problem is largely overcome by using a more complex multidimensional interface reconstruction [2,10-12] , where the shape of the interface cutting the cell is assumed to be planar, but not necessarily normal to the grid. If the normal direction to this plane is known, the position of the material interface can be exactly determined from the volume $\mathbf{V}_a$, of one of the materials in the cell [12]. The normal is assumed to be in the direction of the gradient of the relative partial volume, $\tilde{N}(\mathbf{V}_a/\mathbf{V})$. The gradient is computed using the values of $(\mathbf{V}_a/\mathbf{V})$ in the donor cell and all its neighbors[2].

This high-resolution algorithm is ideal for a well-behaved interface where at least one neighbor to the donor cell is full of material and one is empty. If this is not the case (e.g. if there is a single drop of material in a cell), the normal to the interface is not well defined, in which case SLIC is used. SLIC is also reverted to if there are more than two materials (including voids) in a donor cell.

2.3 Dynamic memory management

A Lagrangian grid deforms with the materials it is modeling, whereas an Eulerian grid is fixed in space and the materials flow through it. Consequently an Euler grid must span all the space reached by any of its materials during the calculation. For problems in which materials move many times their characteristic lengths, the number of Eulerian cells required will far exceed the number of Lagrangian cells necessary to describe the same problem. For three-dimensional problems this ratio may well be more than an order of magnitude. To account for the advection variables, an Eulerian cell also requires about twice the storage of a Lagrangian cell. Consequently, memory requirements for running many Eulerian calculations with adequate grid resolution exceed the available resources. To overcome these difficulties AUTODYN-3D takes advantage of modern compiler technology in using a novel dynamic memory-management scheme. AUTODYN-3D , Version 4, has been completely rewritten in Fortran 90, allowing a simpler and more transparent use of pointers. Each cell in an Euler grid has a pointer, pointing to a location in memory where grid and material variables for the cell are stored (grid coordinates are not included in these data, they are stored in separate arrays). Understanding that empty (void) cells require no grid or material data, the pointers for all empty cells point to a common location in memory containing dummy data for void cells. Each time a cell is emptied, the block of memory pointed to for that cell is released and the pointer is redirected to the common memory location for void cells. When a previously empty cell has material flow into it, a new block of memory is assigned for its grid and material variables and the pointer for the cell (previously pointing to the common memory location) is redirected to the newly assigned memory. A similar technique is used for multi-material cells. As materials flow in and out of cells, the amount of memory pointed to is adjusted according to the number of materials in each cell. For efficiency, system memory is allocated as required in large "superblocks". Released blocks are not returned to the system but reused, using stack logic. Also, a list is kept of the non-empty cells, so that most computations are not carried out for empty cells.

2.4 Lagrange to Euler Remap

The advanced multi-material Eulerian techniques described above improve the efficiency and the quality of Eulerian calculations. However, if grid distortions are small, a Lagrangian calculation still has some definite advantages over Euler. In Euler, the advection phase poses a number of problems.

- The advection terms have to be computed, and this consumes both memory and CPU resources.
- Even using a second order scheme, the advective phase of an Euler calculation adds more diffusion to solutions compared to Lagrange.
- Certain material properties cannot be advected with a high degree of accuracy in Euler.
- It is more efficient to follow a Lagrangian, body-fitted boundary, than a general multi-material interface, which cuts through Eulerian cells.
- It is difficult to track thin layers of material moving through an Euler grid with good accuracy

For these reasons, it is preferable to use a Lagrangian approach whenever practical. The Lagrange to Euler remap algorithm in AUTODYN-3D allows this to be done easily. Often one wants to use a Lagrangian solution in the early stages of a calculation before grid distortions begin to cause serious problems, then switch to an Eulerian solution for the later stages.

The remap algorithm actually serves a dual purpose, as it can also be used to set up and initialize multi-material Euler regions at the start of a calculation. Generating a Cartesian Euler grid is fairly straightforward, but filling the grid with various shaped structures can be very difficult. Simple standard fills, such as blocks, cylinders or spheres, can readily be generated, but custom fitted shapes are usually much more difficult. If such shapes are generated and filled using body-fitting Lagrange grids (a much simpler task), these grids can then be immediately remapped into an Euler grid.

The remap algorithm is based on applying the conservation laws of mass, momentum and energy. As this remap is applied only once in a calculation, a first order scheme is used that is an extension of the 2D algorithm used by Hancock [5,13] (the generalization to second order is straightforward [14]).

First AUTODYN computes the common volume δV_{ij}, between each Lagrangian cell ***i***, and every Eulerian cell ***j***. The Lagrangian cells in AUTODYN are hexahedrons, but the scheme works for any polyhedron. The density ρ_j in the Eulerian cell ***j*** is computed as:

$$r_j = \frac{1}{V_j} \sum_i r_i d V_{ij}$$

Similar relations apply for momentum and energy totals. Remapping of variables is diffusive (this is required for stability), so conservation of momentum will decrease kinetic energy, Consequently, preserving total energy results in increasing internal energy. To prevent unphysical heating in regions where the kinetic energy is large, the internal energy is preserved separately. A volume-weighted average is used for variables that do not have a simple conservation law (e.g. the stress deviators).

3.0 Example: An Explosively Formed Projectile (EFP)

To demonstrate the new capabilities described in this paper, the formation of an explosively formed penetrator (EFP) and its oblique impact on a target plate has been computed using combined Lagrange and Euler simulations.

3.1 EFP Formation

The formation of an EFP is characterized by a relatively smooth deformation process that can be easily handled in Lagrange coordinates, so a Lagrangian calculation is performed for this phase. Figure 1 shows the initial charge configuration.

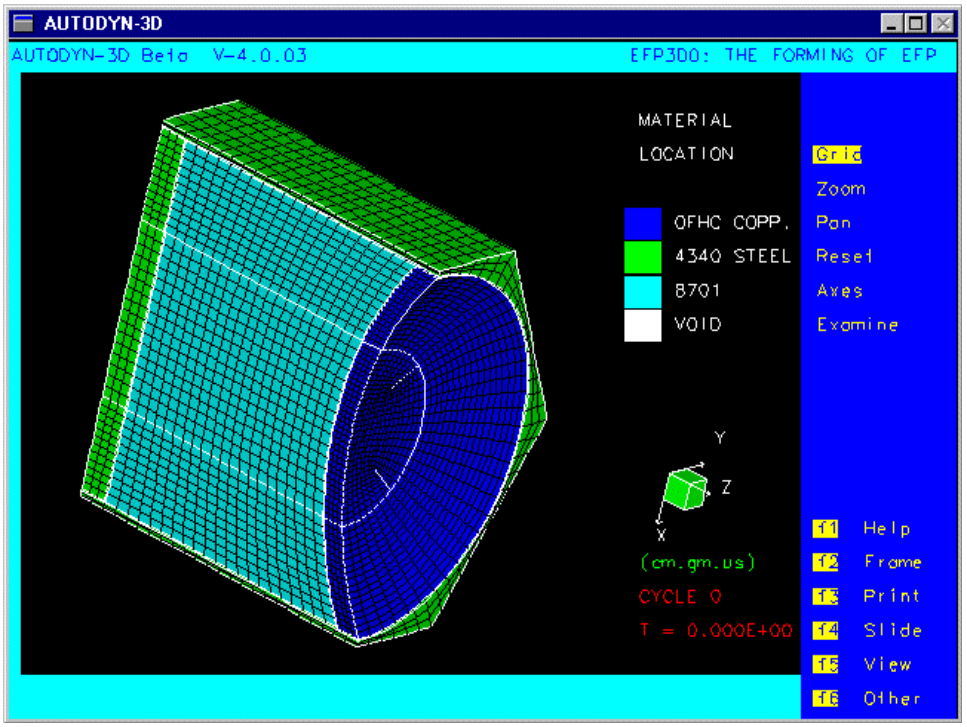


Figure 1: The initial charge configuration

The explosive loading of a thin copper liner forms the EFP. As the liner is loaded, its initial shape causes it to converge inwards towards its axis, forming a compact, high-speed projectile that can impact targets at large stand-off distances. To perform such a calculation in Euler would be much less efficient, because it would be necessary to resolve the thin liner as it moved through the Euler grid. AUTODYN's Lagrangian processor can easily handle the

explosive-liner interaction and liner deformation that occurs. The steel casing surrounding the explosive is not axially symmetric, so a three-dimensional analysis is required. However, there is a symmetry plane at y=0, so only half the geometry is modeled.

Figure 2 shows the shape of the liner after formation, when it has reached the stand-off of the target plate.

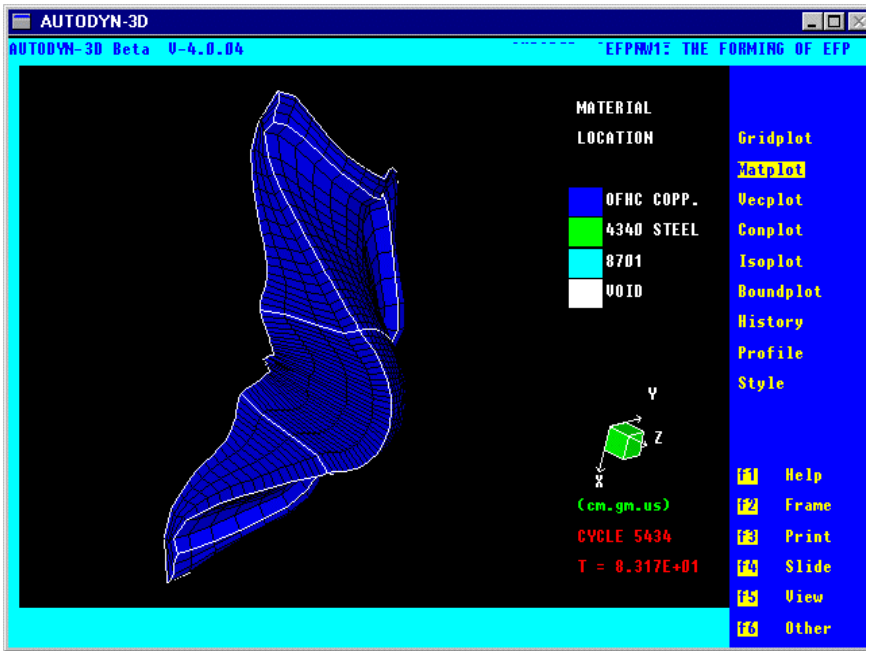


Figure 2: The liner (EFP) after formation

By this time, the loading of the liner by the detonation products is complete, so the explosive and casing are removed from the calculation, leaving only the EFP grid.

3.2 EFP Penetration

The next step is to compute the impact and the penetration of the EFP into a target where both projectile and target are expected to undergo large deformations. This type of problem is most commonly solved in Eulerian coordinates. To achieve this, a Cartesian Euler grid is generated to cover the region occupied by the projectile and the target, and regions into which they may be expected to move during the calculation. The EFP is then remapped from the Lagrange grid into the Euler grid. Figure 3 shows the Lagrange EFP grid superimposed in the Euler grid before the remap. Figure 4 shows the EFP remapped into the Euler grid.

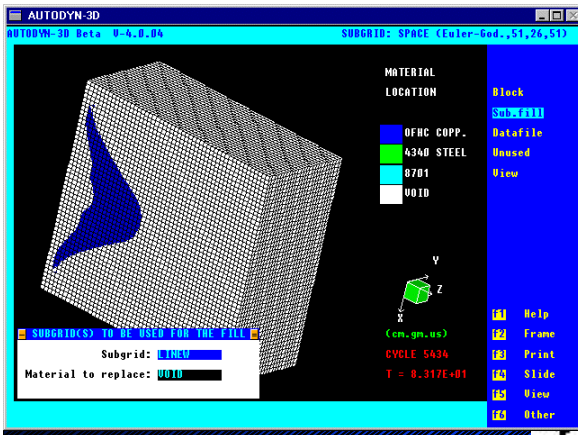


Figure 3: Lagrange EFP and Euler grid before remap

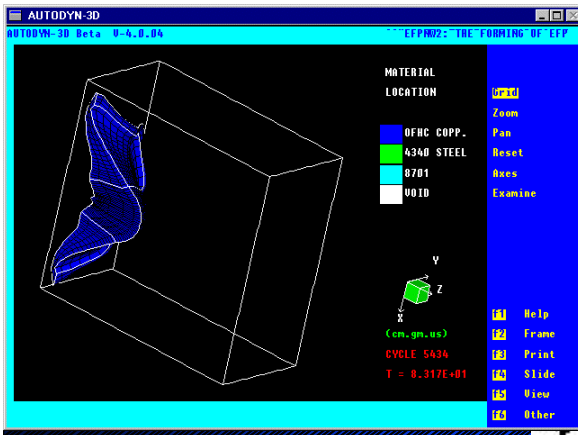


Figure 4: EFP remapped into the Euler grid

A target plate is generated in a similar way. The plate, which is at 45 degrees to the projectile, is first generated as a Lagrangian grid and then its contents remapped into the Cartesian Euler grid. Figure 5 shows the Euler grid with both the projectile and target plate remapped into it. Figure 6 shows the material surface contours in the Euler grid.

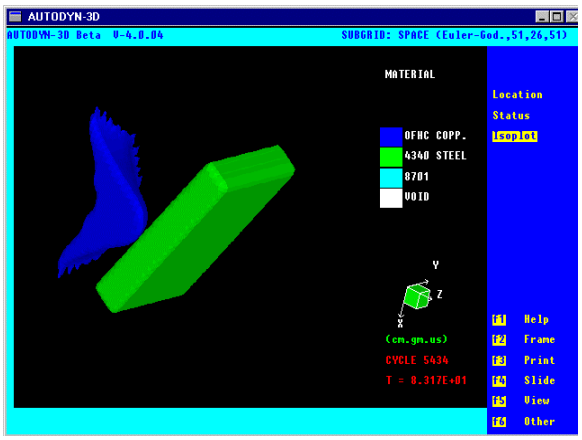


Figure 5: Projectile and target remapped into Euler

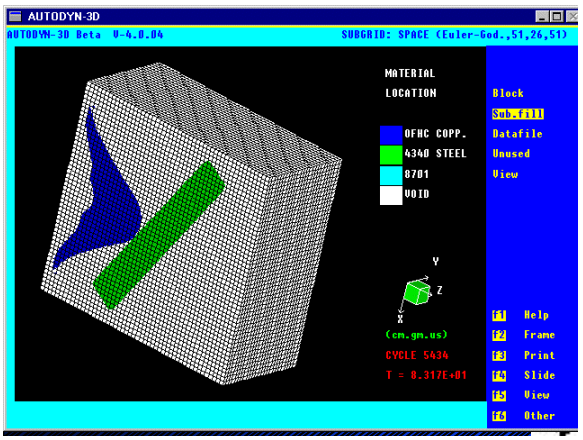


Figure 6: Material contour plot after the remap

Notice that in figure 5, the majority of the cells in the Euler grid are empty. The dynamic memory-management scheme implemented in AUTODYN-3D greatly reduces memory requirements in this situation, allowing the calculation to be carried out on a PC with only 64 Mb of memory

Figure 8 shows the Euler grid at two different times during the penetration calculation. Figure 9 shows material contour plots at the same two times.

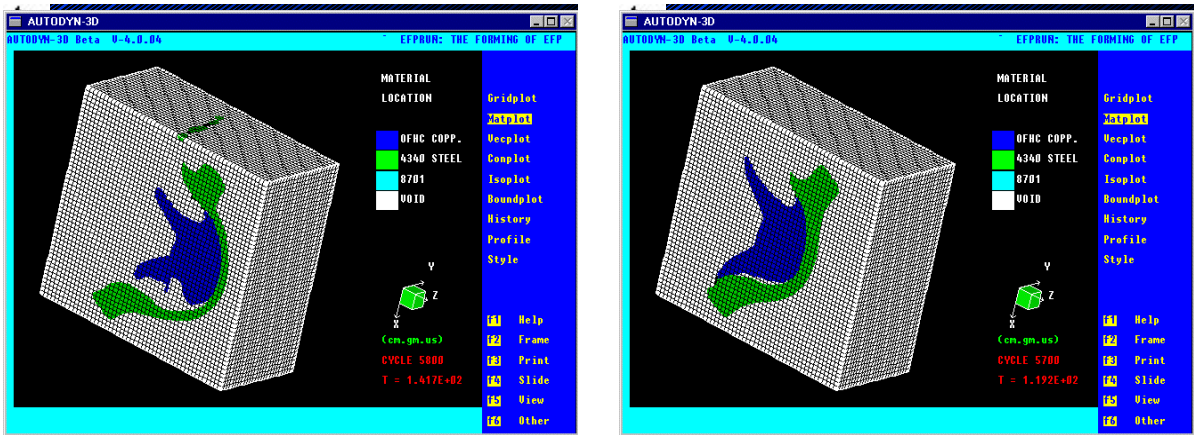


Figure 8: The Euler grid at two different times during the penetration calculation

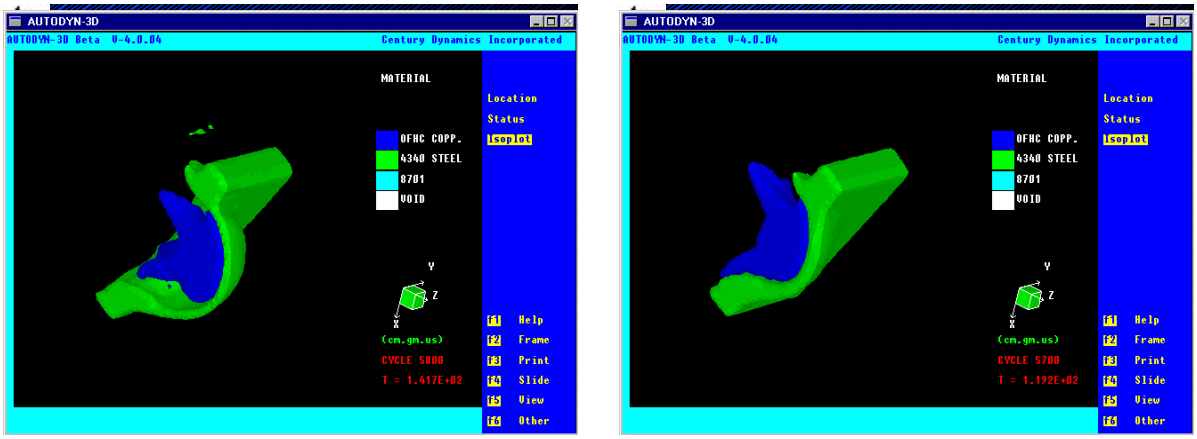


Figure 9: Material contour plots at two different times during the penetration calculation

4. Conclusions

This paper has described and illustrated some novel Eulerian techniques implemented in the three-dimensional code AUTODYN-3D. They include a second order Godunov processor with a high-resolution multi-material interface tracking algorithm and efficient dynamic memory-management scheme, and a Lagrange to Euler remap capability. An example has demonstrated how, together, these new features can alleviate traditional drawbacks associated with Eulerian techniques. In particular, dynamic memory management permits virtual Eulerian grids to be set up that require much less memory than conventional Euler grids, and the remap capability allows the combined advantages of Lagrange and Euler techniques to be brought to bear in solving complex interaction problems.

REFERENCES

[1] B. van Leer, J.Comp.Phys, 32, p101, (1979)

[2] D.L.Youngs, "An Interface Tracking Method for a 3D Hydrodynamics", AWRE-44-92-35, (1987)

[3] J. Von Neumann and Richtmyer, J. Appl. Phys., 21, p232, (1950)

[4] S. K. Godunov, Mat.Sb., 47, p271, (1959)

[5] S. L. Hancock, "PISCES 2DELK Theoretical Manual", Physics Int. Co., (1985)

[6] F.H. Harlow, Proc. Symp. Appl. Math., 15, p269, (1963)

[7] S. Osher, J. A. Sethian, J.Comp.Phys., 79, p12, (1988)

[8] W. E. Johnson "Development and Application of Programs Related to Hypervelocity Impacts", Systems Science, Software rep. 3SR-353, (1970)

[9] W. F. Noh and P.R.Woodward,"SLIC,Simple line Interface Calculation", in the Proc. of the 5th Int. Symp. on the Num. Meth. in Fluid Dynamics, held at Enschede, The Netherlands,(1976), A. I. Van Der Vooren et al. ed., Springer Verlag, (1976)

[10] R. De Bar "The KRAKEN code…", in L.L.N.L rep. UCID_17366,(1974) or "A Method in Two-D Eulerian Hydrodynamics", UCID-19683, LLNL(1974)

[11] D. L. Youngs , "Time dependent multi-material flow with large fluid distortion", in Num. Meth. in Fluid Dynamics, ed.K. W. Morton, B. W. Baines, Academic Press, (1982)

[12] G. Luttwak, "Numerical Simulation of water jet penetration", p191, Shock Waves in Condensed Matter-1983, ed.J.R.Asay et al., Elsevier Science Publishers B.V., (1984)

[13] B. J. Thorne, D. B. Holdridge, "The TOOREZ Lagrangian rezoning code", Sandia N.L. rep. SLA-73-1057, (1974), see appendix by D.S.Mason.

[14] G.Luttwak, "Second Order Discrete Rezoning", p1777 in High Pressure Science and Technology-1993, Schmidt et al. ed., AIP-Press (1994)

[15] C.W.Hirt,B.D.Nichols, J.Comp.Phys., 39, p201, (1981)