Computer Assisted Mechanics and Engineering Sciences, 6: 295–311, 1999.

Coupled techniques for the simulation of fluid–structure and impact problems

Naury K. Birnbaum[1], Nigel J. Francis[2] and Bence I. Gerber[1]
[1] *Century Dynamics, Inc.*
2333 San Ramon Valley Blvd., Suite 185, San Ramon, CA 94583
[2] *Century Dynamics, Ltd.*
Dynamics House, Hurst Road Horsham, West Sussex, England RH12 2DT

(Received March 8, 1999)

The ability to model the different regimes of a physical event using different numerical techniques has a number of advantages. Instead of applying the same general solver to all domains of a problem, a solver optimized for a particular regime of material behavior may be used. Thus, in a single analysis, one type of solver may be used for fluid behavior while another type is used for solid/structural response. The various domains in the problem are then coupled together in space and time to provide an efficient and accurate solution.

Examples in the use of Eulerian, Lagrangian, Arbitrary Lagrange Eulerian (ALE), Structural, and Smooth Particle Hydrodynamic (SPH) techniques, in various combinations, will be applied to general problems in fluid–structure interaction and impact problems. The relative advantages and limitations of such coupled approaches will be discussed. Examples include the following:

- fluid dynamics, blast, impact/penetration (Euler)
- hypervelocity impact onto spacecraft shields (Lagrange–Lagrange)
- buckling of a thin walled structure (Structural)
- impact and penetration of a projectile onto concrete (SPH–Lagrange, Lagrange–Lagrange, Euler–Lagrange).

Numerical results are presented with comparison against experiment where available.

1. Introduction

For problems of dynamic fluid–structure interaction and impact, there typically is no single "best" numerical method which is applicable to all parts of a problem. Techniques to "couple" different types of numerical solvers in a single simulation can allow the use of the most appropriate solver for each domain of the problem.

The basic solvers for explicit integration numerical wavecodes (sometimes termed "hydrocodes") are outlined with their associated strengths and weaknesses. Techniques for coupling these solvers are further discussed. Several applications of the coupled methodologies are then presented with comparison of calculations with experimental results.

2. Numerical solvers

The numerical solvers utilized in hydrocodes generally fall into the following general classes: Lagrange, Euler, ALE, Structural, and Meshless methods. The general applicability and limitations of each type of solver is outlined. By judicious selection of the appropriate solver for the various regimes of a simulation, an optimal solution in terms of accuracy and efficiency can be achieved.

2.1. Lagrange

In a Lagrange solver processor, the numerical mesh moves and distorts with the physical material as shown in Fig. 1.

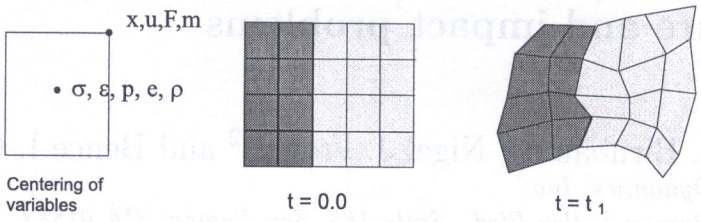


Fig. 1. Lagrange solver

A Lagrange solver is generally well suited for the description of solid materials. Free surfaces and material interfaces are located at cell boundaries and as such are well maintained throughout the calculation. History dependent material properties are also well described in Lagrange since the material and its history are contained in its original cell.

In the Lagrangian framework, coordinates (x), velocities (u), forces (F), and masses (m) are typically defined at corner nodes. Stresses (σ), strains (ε), pressures (p), energies (e), and densities (ρ) are cell centered.

While Lagrange is particularly well suited for the description of solid behavior, its main drawback is that for severe deformations the numerical mesh may become overly distorted with a resulting small timestep and possible loss of accuracy. A simple example shown in Fig. 2 illustrates this situation.

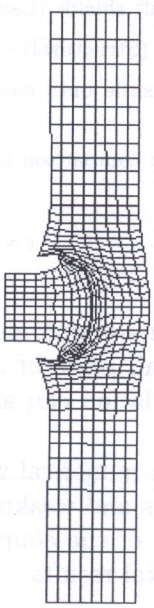


Fig. 2. Lagrange impact problem with mesh distortion

In the problem above the numerical solution can only be carried to a certain point before the Lagrangian mesh distortions cause the analysis to be stopped due to a very small timestep or mesh tangling. However, programs may include additional features that may be applied to a Lagrange solution to extend an analysis. These features include rezoning and erosion.

Rezoning involves the re-establishment of a "regular" mesh through the mapping of the distorted mesh quantities onto a newly defined mesh. An erosion technique also can overcome many of the problems associated with using a Lagrange technique for high deformation situations. Typically, if and when a limiting strain is reached within an element, the element is then "eroded", i.e. either discarded, or alternatively, transformed from a solid element to a free mass node disconnected from the original mesh. This technique thereby avoids the mesh distortion problem. Many programs simply discard the eroded node while others provide an option to retain the inertia of the eroded element as an interacting free mass node.

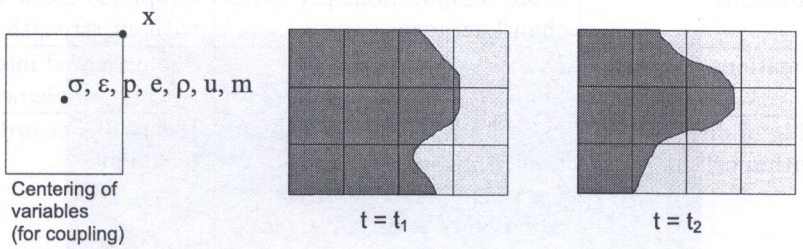
The attributes of the Lagrange solvers are summarized in Table 1.

Table 1. Lagrange solver summary

Advantages	Limitations	Enhancements to overcome limitations
Efficiency, fewer computations per timestep relative to other solvers	Element distortion can lead to small timestep	Rezoning and erosion
Clear definition of material interfaces and boundaries	Cell distortion can lead to grid tangling and inaccuracies	Rezoning and erosion
Good time history information	Thin sections need small timesteps	Use structural solvers
Good for strength modeling	Complex logic for sliding interfaces	
Simpler code		

2.2. Euler

In an Euler solver, the numerical mesh is fixed in space and the physical material flows through the mesh as shown in Fig. 3.



- Grid fixed in space, material flows through cell faces
- Normally used for modelling of fluids, gases and large deformations
- Multiple materials and strength included
- Cell centered varibles used facilitate coupling to Lagrange

Fig. 3. Euler solver

An Euler solver is typically well suited for the description of fluid and gas behavior. Free surfaces and material interfaces can move through the fixed Euler mesh. Sophisticated numerical techniques must be used to track the material motions. Because the mesh is fixed, large deformations or flow situations, by definition, do not result in mesh distortions. The tradeoff is the extra computational

work required to maintain interfaces and to limit numerical diffusion. Also, to describe solid behavior, the solid stress tensor and the history of the material must also be transported from cell to cell.

A number of different approaches for the centering of variables have historically been used. To facilitate coupling with other types of solvers, a scheme wherein all variables (stress, strain, pressure, energy, density, and velocities) are cell centered, as shown in the figure, is sometimes used [5].

The examples described below show the use of an Euler solver for gas dynamic and blast problems.

Figure I shows a two-dimensional explosion in a shock tube. This illustrates the use of a multiple material Euler solver applied to fluid (no strength) dynamics.

Figure 4 above shows the use of Euler for multiple materials including strength. The liquid metal jet is made of copper while the target is steel. The Eulerian multiple material formulation allows materials to separate and create free surfaces (void creation) automatically.

Figure II shows the blast loading on a set of three dimensional buildings. This example illustrates three-dimensional fluid dynamics with rigid structures.

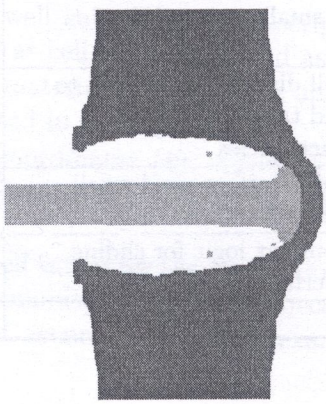


Fig. 4. Euler: high speed metal jet penetration

Table 2. Eulerian solver summary

Advantages	Limitations	Enhancements to reduce limitations
No grid distortions	More computations per cycle than Lagrange	Simplified Euler formulations without strength
Large deformations handled	Diffusion of material boundaries	Sophisticated interface tracking implementations
Allows mixing of different materials within cells	Need finer zoning for similar accuracy compared to Lagrange. Need iterative equation of state solvers for multiple material cells	Use of higher order techniques
Rezoning not required	Need void cells defined in space where material may flow can lead to very large meshes	Use of dynamic memory allocation to store information only where needed
	Shock diffusion	Higher order implementations
	Less flexible for strength modeling than Lagrange solvers	
	Thin sections need many cells	Couple to alternative solvers

Fig. I. Euler: explosion in air filled shock tube

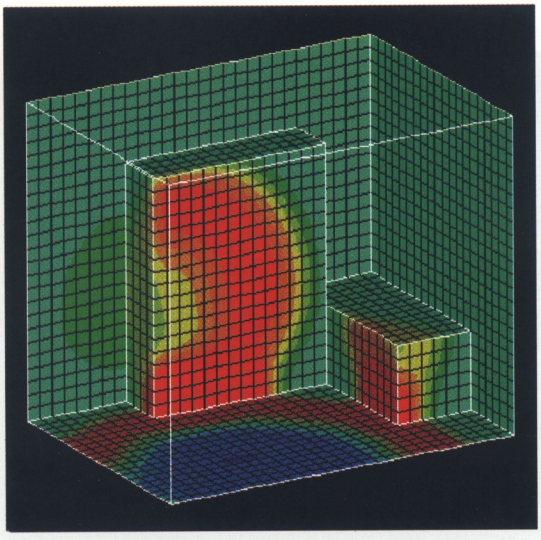


Fig. II. Euler: 3D blast on rigid buildings

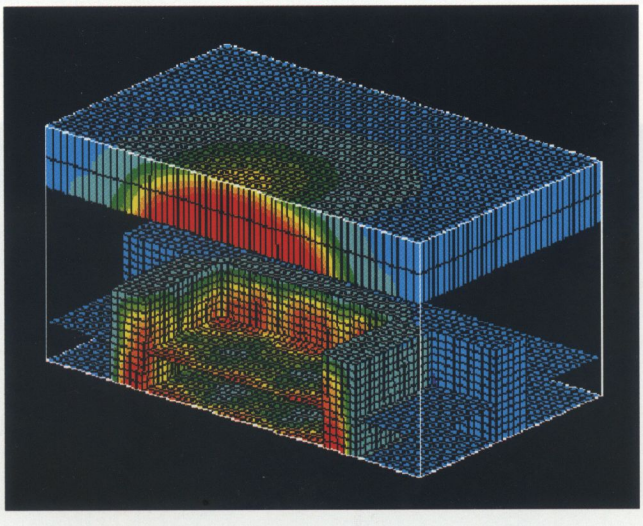


Fig. III. Arbitrary Lagrange Euler fluid–structure interaction (fluid regions not depicted except on single horizontal plane)

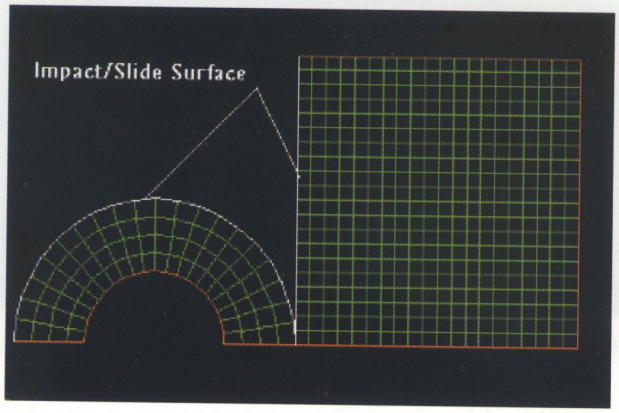


Fig. IV. Impact slide (Lagrange–Lagrange) surface (2D)

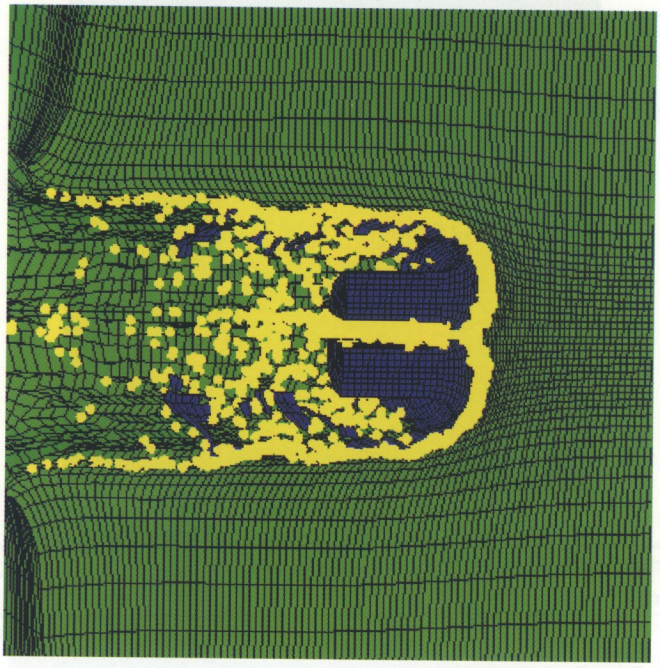


Fig. V. Lagrange–Lagrange interaction with erosion

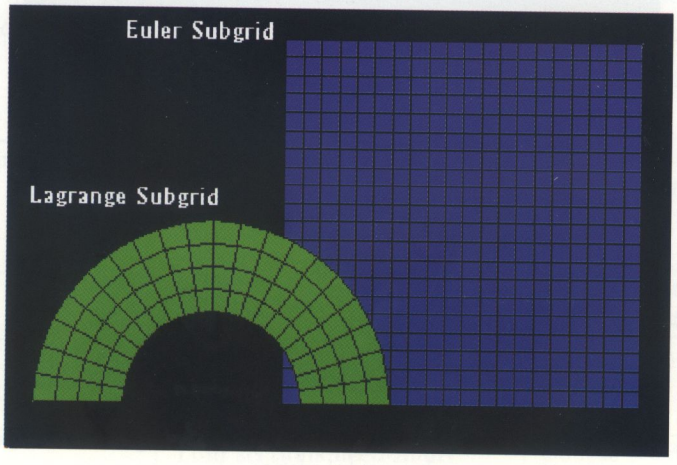


Fig. VI. Euler–Lagrange coupling

2.3. Structural solvers

Structural solvers may be used to model thin structures, beams, rods and other structural elements in situations where the use of a standard Lagrange or Euler solver would impose a very small timestep on the calculation. In the case of a shell solver, it is presupposed that the structure being modeled is "thin" such that a biaxial state of stress may be assumed. No wave propagation occurs across the shell thickness but only along its length. Thus, the timestep (t) is constrained only by the zonal dimension along the length.

An example of a 3D Shell calculation is shown in the crushing of a thin hexagonal girder as shown in Fig. 5. Modeling such phenomena in Lagrange or Euler would be virtually impossible.

A summary of a structural shell solver is provided in Table 3. Other types of structural solvers would include beams, rods, springs, etc. wherein a continuum approach is also not applicable.

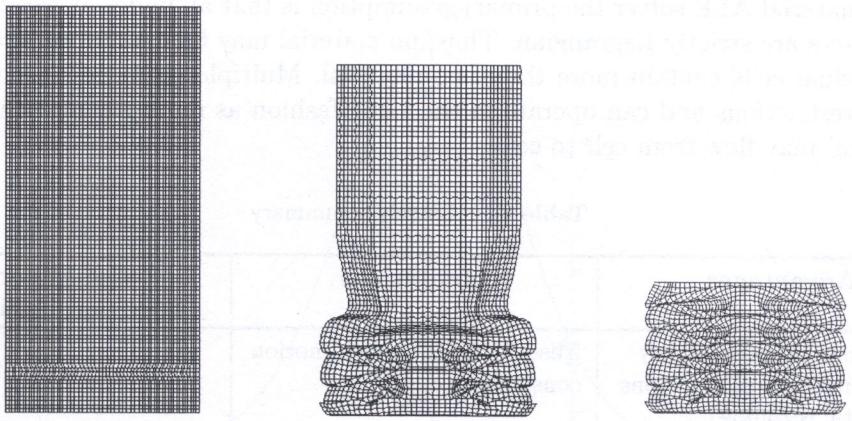


Fig. 5. Time sequence of the buckling of an hexagonal girder

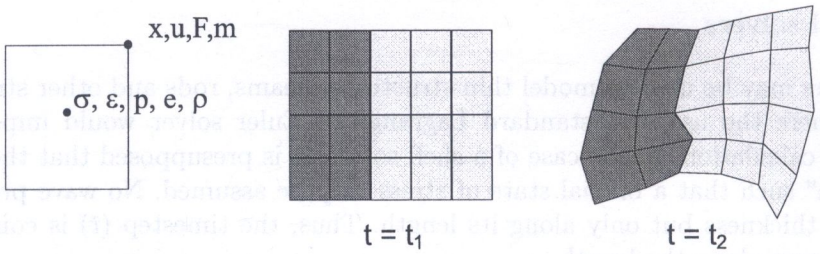
Table 3. Shell solver summary

Advantages	Limitations
Time step controlled by the segment length, not thickness	Not accurate for thick structures
	Biaxial stress state

2.4. ALE (Arbitrary Lagrange Euler)

An ALE (Arbitrary Lagrange Euler) solver allows for a type of "automatic rezoning" which can be quite useful for certain problems. Depending on the specified motions, the ALE domain can be completely Lagrangian (the nodes move with the material motion), completely Eulerian (the nodes are fixed and the material moves through the fixed mesh), or something in between, as shown in Fig. 6.

ALE may be used for the modeling of solids, fluids, and gases. It is particularly well suited for a variety of fluid–structure interaction problems. An example of ALE is shown in Fig. III. An explosion inside a building is modeled in 3D. The explosion and fluid dynamics regions of the model are simulated using Euler constraints while the solid structures are modeled with Lagrange constraints [3].



- Material interfaces are Lagrange and interior calls may be Lagrange, Euler or any other motion constraint
- Modelling of solid continua, fluids and gases including fluid–structure interaction

Fig. 6. ALE solver (single material elements)

In a single material ALE solver the primary assumption is that all nodes at free boundaries or at material interfaces are strictly Lagrangian. Thus, no material may flow in or out of an ALE domain nor may individual cells contain more than one material. Multiple material ALE formulations do not have such restrictions and can operate in the same fashion as multiple material Euler schemes wherein material may flow from cell to cell.

Table 4. ALE solver summary

Advantages	Limitations	Enhancements to overcome limitations
Wide range of applicability with arbitrary mesh motions (automatic rezoning)	Need to specify mesh motion constraints	
Clear definition of material interfaces and boundaries	Cell distortion can lead to grid tangling and inaccuracies	Rezoning and erosion
Good time history information	Thin sections need small timesteps	Use shell solver
Good for strength modeling	Complex logic for sliding interfaces	

2.5. Meshless techniques

There are a number of meshless techniques such as Smooth Particle Hydrodynamics and Element Free Galerkin. In this discussion we will address only one generic meshless technique, SPH, but the comments are generally applicable to most meshless techniques.

SPH is a Lagrangian technique having the potential to be both efficient and accurate at modeling material deformation as well as flexible in terms of the inclusion of specific material models. In addition, SPH is a meshless or gridless technique such that it does not suffer from the normal problems of grid tangling in large deformation problems. Presently many large deformation problems are calculated using Eulerian techniques that do not suffer from grid tangling but have some limitations in terms of modeling material interfaces, the inclusion of specific material models, and associated high computational expense.

The main potential advantages of the SPH technique are that in not requiring a numerical grid, no grid tangling, by definition, can occur. SPH is Lagrangian in nature thus allowing efficient tracking of material deformations and history dependent behavior. Compared with Euler, it is more efficient

in that one only need model regions where materials exist, not all regions where material might exist. Being meshless, phenomenon such as fracture and fragmentation may be modeled. Fracture can occur arbitrarily without the a priori strictures of a numerical mesh.

With all of its promise, however, SPH technology is relatively immature compared with standard grid based Lagrangian and Eulerian techniques. Several problems need to be solved before the technique becomes a fully developed computational continuum dynamics technique. There are remaining known problem in the areas of stability, consistency, and conservation. Despite these problems very encouraging results have been obtained for a variety of applications using the SPH technique [6, 7, 8, 9]

The name SPH includes the term "particle". Although this is appropriate for describing the Lagrangian motion of mass points in SPH, it is misleading because the "Particles" are really interpolation points. This is best demonstrated through a simple example. Consider a rod of steel which is represented by a series of SPH particles, as illustrated in Fig. 7.

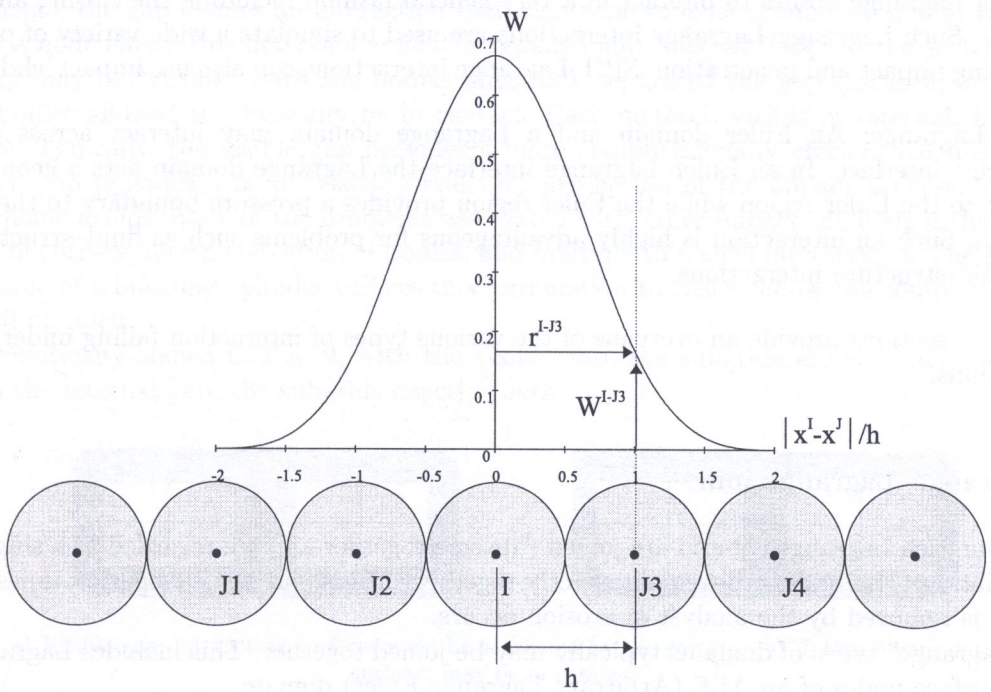


Fig. 7. Kernel density example

The density at a particle I can be calculated using an expression such as

$$\rho = \sum_{J=1}^{N} m^J W^{IJ}(x^I - x^J, h)$$

where m^J is the mass of particle J, W^{IJ} is a weighting function (Kernel B-spline), x is the position of the center of a particle and h is known as the smoothing length or particle size.

Therefore, to calculate the value of a function at particle I, in this case density, the value of the function is summed at all neighboring particles (interpolation points $J1$, $J2$, I, $J3$, $J4$) multiplied by a weighting function (the Kernel function).

Hence, the SPH particles are not simply interacting mass points but rather they are interpolation points from which values of functions, and their derivatives, can be estimated at discrete points in the continuum. In SPH, the discrete points at which all quantities are evaluated are placed at the center of the SPH particles.

3. Interaction of solvers, coupling algorithms

In order to optimize the solution of "coupled" problems (e.g. impacts, explosions, fluid–structure interactions, etc.) techniques have been devised to allow interaction between solvers. Thus different numerical solvers can be applied selectively according to the phenomena being modeled and then linked together automatically in space and time.

3.1. Types of interaction

The typical types of interaction available depend on the solver types that are to be coupled:

- Lagrange–Lagrange: Two Lagrange type domains (including Shell and ALE) may interact through either a joined ("glued") or an impact–slide surface. Impact–slide surfaces allow the surfaces of Lagrange bodies to interact in a very general fashion including the closing and opening of gaps. Such Lagrange–Lagrange interactions are used to simulate a wide variety of phenomena including impact and penetration. SPH–Lagrange interactions can also use impact–slide surfaces.

- Euler–Lagrange: An Euler domain and a Lagrange domain may interact across an Euler–Lagrange interface. In an Euler–Lagrange interface the Lagrange domain acts a geometric constraint to the Euler region while the Euler region provides a pressure boundary to the Lagrange domain. Such an interaction is highly advantageous for problems such as fluid–structure or gas dynamic–structure interactions.

The next sections provide an overview of the various types of interaction falling under the above classifications:

3.2. Lagrange–Lagrange join

Nodes from each Lagrange subgrid are joined ("fused") together and are regarded as a single node in the calculation. The nodes will remain exactly together throughout the calculation unless the Join condition is removed by the analyst or erosion occurs.

All "Lagrange" types of domains typically may be joined together. This includes Lagrange, Shell, and the surface nodes of an ALE (Arbitrary Lagrange Euler) domain.

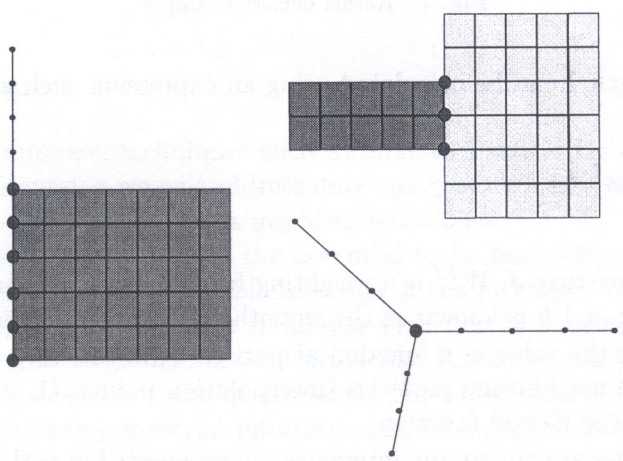


Fig. 8. Joined Lagrange grids (including shells)

3.3. Lagrange–Lagrange interaction (dynamic impact/slide surface)

A surface on one Lagrange domain can interact with another surface of a different domain allowing for impact and sliding, gap opening and closing, between the bodies. The impact/slide surface typically may include friction or not. Gaps may open and close along a slide surface. Figure IV illustrates a generic Lagrange–Lagrange interface.

In many programs that include Lagrange–Lagrange interactions, each time step an array is constructed containing all the current surface faces of the first body termed the "master". All surface nodes of the second body, the "slave", are then tested to see if penetration of the master faces occurs during the current time step. If any of the slave nodes penetrate, momentum conserving interactions are computed to prohibit penetration. When this is completed the master and slave bodies are reversed and the procedure repeated, providing symmetry to the process.

A number of different interaction algorithms have been devised by various developers. One that is particularly robust is based upon the use of a small "gap" to determine if bodies are interacting [3]. In this scheme, the gap defines an interaction detection zone to exist around each interacting face. Whenever a node enters this detection zone it is repelled and "pushed back" to the surface.

The gap may be "visible", with the bodies physically separated the gap distance, or "internal" with the bodies allowed to physically be in contact. Each method, visible or internal, has certain advantages. Typically, the visible gap provides a more computationally efficient calculation while the internal gap provides a more "visually realistic" perception of the impact process. The use of the gap scheme avoids many of the problems associated with other impact–slide algorithms such as treatment of corners, interpenetration of bodies, and multiple thin shell interactions. The previously cited example of a buckling cylinder utilizes this formulation to easily follow the complex folding of a thin shell on itself.

As schematically shown in Fig. 9, with the visible gap, the subgrids are separated by a small gap. With the internal gap, the subgrids exactly touch.

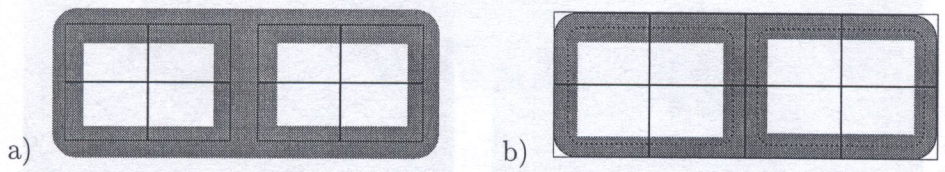


Fig. 9. a) Visible gap: interacting surfaces must be separated by the gap size; b) Internal gap:: interacting surfaces may be in contact

As a Lagrangian body moves and distorts, and potentially erodes, the interacting Lagrange surface must be automatically redefined. When a Lagrangian element cell is eroded, the mass of the cell can either be discarded or retained at the four corner nodes of the element. If the mass is retained, conservation of inertia and spatial continuity of inertia are maintained during the erosion process.

If the four cells surrounding a particular node are eroded, the node becomes a free node. Free nodes can be automatically added to the arrays of slave nodes used in the impact–slide logic and thus continue to interact with all defined impact–slide boundaries. Figure 10 illustrates the creation of free nodes and the automatic redefinition of the impact/slide surface whenever erosion occurs

Erosion does not necessarily simulate a physical phenomenon. It is fundamentally a numerical technique introduced to overcome the problems associated with the mesh distortions caused by gross motions of a Lagrange grid. When a node is eroded the compressive strength of the material within the zone is lost and the mass of the eroded node may be retained or discarded. Erosion strains should therefore be chosen such that zones are not eroded until they are quite severely deformed and their compressive strength and/or mass are not likely to effect results. Typical erosion strains are usually taken to be in excess of 150%. This feature extends the capabilities of Lagrange type calculations for extreme deformation problems.

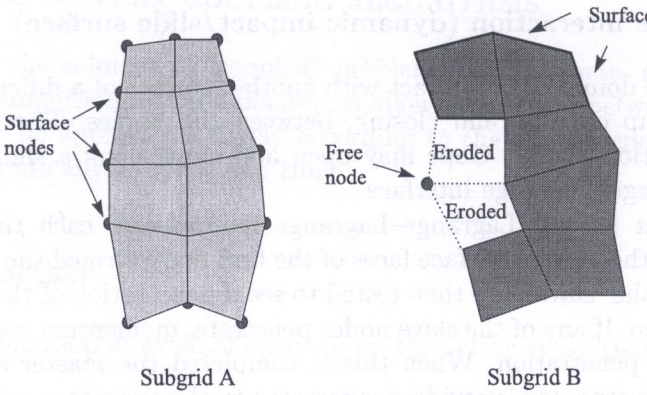


Fig. 10. Eroded nodes: dynamic redefinition of impact/slide surface

Figure V shows the three dimensional Lagrange–Lagrange interaction of two tungsten rods impacting steel at 4 km/sec. Erosion has been specified with retained inertia. The eroded free nodes are depicted as small spheres

A second example (Figs. 11 and 12) demonstrates the excellent agreement with experiment that can be obtained using an erosion technique. The oblique impact of an aluminum sphere at 6.56 km/sec on an aluminum plate is simulated using Lagrange–Lagrange interaction with erosion (retained inertia option). The problem is of interest in protecting spacecraft against orbital debris.

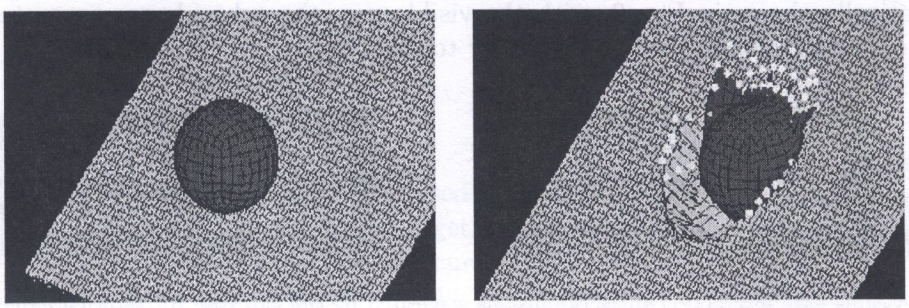


Fig. 11. Lagrange–Lagrange interaction: oblique impact at 6.56 km/sec

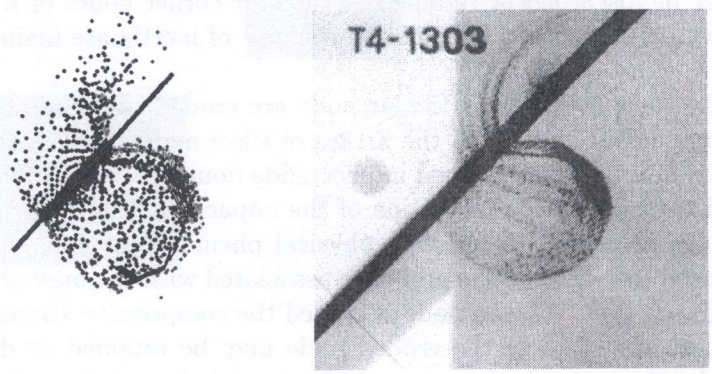


Fig. 12. Debris cloud: calcultion vs. experiment

3.4. Euler–Lagrange interaction

An Euler region and a Lagrange body may interact across an Euler–Lagrange interface. In an Euler–Lagrange interface the Lagrange body acts a geometric constraint to the Euler region while the Euler region provides a pressure boundary to the Lagrange body. This is depicted schematically in Fig. VI.

As the Lagrange subgrid moves and distorts, it "covers" and "uncovers" the fixed Eulerian cells. Logic must be implemented to avoid overly small Eulerian cells that may arise during this processS-mall Eulerian cells that could severely limit the calculation timestep typically are automatically blended with larger neighbors. The Euler–Lagrange interface allows complex fluid–structure interaction problems to be solved as well as large deformation structural problems.

Usual Euler–Lagrange implementations use a polygon concept to define the physical surfaces of the various Lagrange bodies that are used in Euler–Lagrange interaction. This approach presupposes that there is a regular, bounded surface along with the interaction occurs. The interacting polygons will move according to the internal stresses in the Lagrange body coupled with the boundary pressures from the adjacent Euler subgrid. Figure 13 shows the polygon interacting with Euler.

The example in Figs. 14 and 15 simulates a steel projectile impacting and penetrating an aluminum target. The steel projectile is modeled in Lagrange while the target is modeled in Euler. The initial setup showing the Lagrange and Euler domains and the interacting polygon are shown as well as the results of the calculation after penetration.

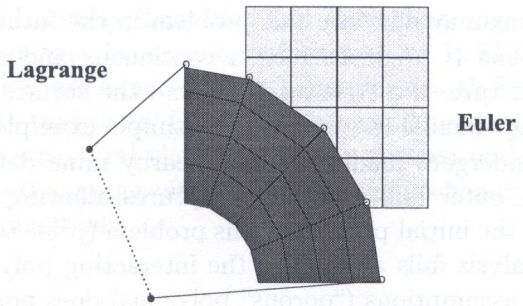


Fig. 13. Lagrange polygon interacting with Euler region

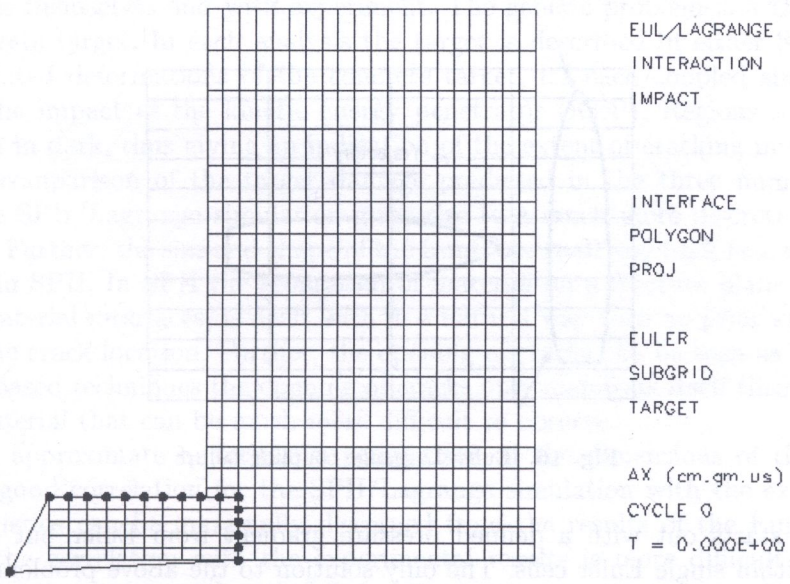


Fig. 14. Euler–Lagrange initial setup

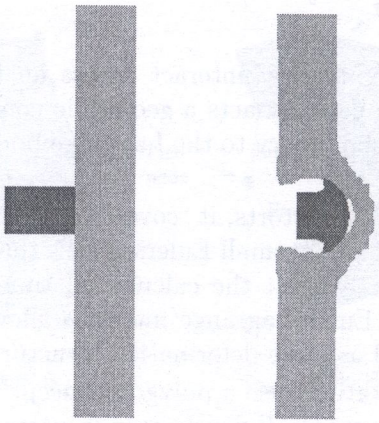


Fig. 15. Euler–Lagrange penetration simulation

The Euler–Lagrange coupling can be very powerful for fluid–structure interaction problems. But it does have a number of limitations that restrict its use.

The Euler–Lagrange schemes have only proved themselves efficient and useful in 2D. In 3D, the computational requirements are excessive even on supercomputers. This stems from the substantial and complex intersection calculations that must be performed each timestep. Increasingly powerful computers and new algorithms may alleviate this problem in the future.

The Euler Lagrange scheme is predicated on a continuous and closed interacting surface as defined by the Lagrange structure. If a structure fails and the surface is no longer continuous, the Euler Lagrange scheme cannot handle the situation. A simple example illustrates this limitation.

A double hulled vessel undergoes loading from a nearby mine detonation. The initial loading deforms the outer hull. The outer hull eventually ruptures allowing fluid and loadings onto the secondary hull. Even though the initial portion of this problem (prior to rupture) can be solved with Euler/Lagrange, such an analysis fails as soon as the interacting polygon is no longer continuous. Even the use of very ad hoc assumptions ("porous" polygons) does not alleviate such issues.

An additional limitation, depending on the Euler mesh resolution, is that a Lagrange body can get "lost" within the Euler mesh, with no pressure gradient acting upon it. The Lagrange body will artificially be "trapped" in a given location, as shown in Fig. 16.

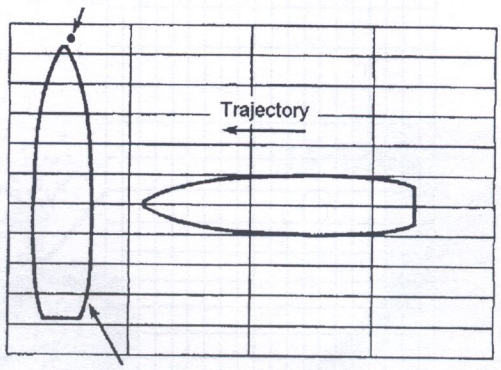


Fig. 16. Euler–Lagrange coupling failure

The projectile starts out with a defined pressure gradient from Euler but after tumbling is later contained within single Euler cells. The only solution to the above problem would be a very finely zoned Euler mesh, which in 3D usually makes the calculation impractical even on the fastest supercomputers.

Thin shell structures with ends or structural details cannot readily be used in Euler–Lagrange coupling because of the inability of creating a continuous and closed polygon [12]. The example shown in Fig. 17 cannot be solved with current Euler Lagrange coupling methodologies.

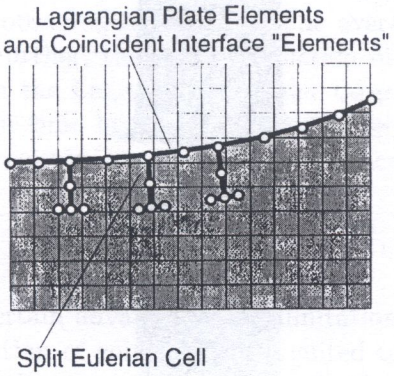


Fig. 17. Difficult problem for Euler–Lagrange coupling

3.5. SPH–Lagrange coupling

Because of the limitations of Lagrange/Lagrange and Euler/Lagrange coupling current work using SPH/Lagrange coupling is underway by a number of researchers. In SPH/Lagrange coupling, regions where large flow and distortions are expected may be modeled with SPH while solid and structural response may be modeled with traditional Lagrange/structural approaches. The interaction between SPH nodes and Lagrange or structural elements may be treated with algorithms similar to those used for continuum Lagrange/Lagrange interactions. In such schemes, the SPH nodes are treated as "slave" nodes to the Lagrange body.

3.6. Coupled calculation comparisons

A final set of examples, illustrated in Fig. 18, shows three different types of coupled analyses compared between themselves and with experiment. The generic problem is a Lagrange projectile impacting a concrete target. In each analysis the target is described in either SPH, Lagrange, or Euler. The calculated deformations of the concrete target, for each coupled simulation is shown 3.5 msecs after the impact of the kinetic energy penetrator (KEP). Regions which are failed in tension are shown in dark, thus giving an indication of the extent of cracking in the concrete.

A qualitative comparison of the target damage predicted in the three numerical simulations indicates that the SPH/Lagrange simulation gives rise to a much more discrete representation of the failed regions. Further, the size and shape of the front face spall and back face scab are visualized particularly well in SPH. In SPH the separation of material on a fracture plane, and the creation of new internal material interfaces, is dealt with in a natural way with no prior knowledge required with respect to the crack location. Further, the opening of cracks can be seen as the SPH particles separate. In grid based techniques the opening of cracks only manifests itself through the growth of failed zones of material that can be much more difficult to observe.

In Table 5 an approximate quantitative comparison of the dimensions of the failed concrete regions indicates good correlation for the SPH/Lagrange simulation with the experiment. Similar damage characteristics can be marginally discerned from the results of the Euler/Lagrange simulation however the correlation with the experimental results is more difficult to define. In the Lagrange/Lagrange simulation, although failed regions are concentrated near the KEP entry and exit points, no discrete crack patterns can be observed or correlated with the experiment.

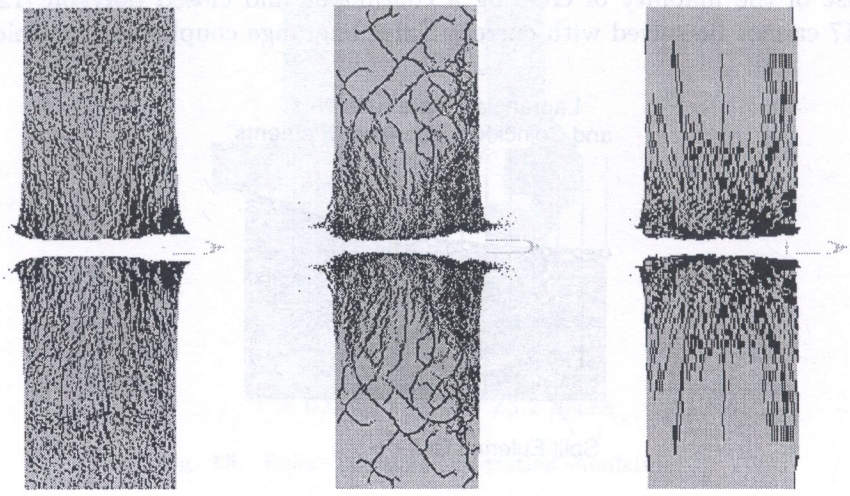


Fig. 18. Final deformation, SPH–Euler–Lagrange simulations

Table 5. Comparison of experimental/calculated target damage

	Damage: mm		
Diameters	**Euler/Lag**	**Euler/Lag**	**SPH/Lag**
Bore Hole	90 (±10)	90 (±10)	90 (±10)
Primary Front Face Spall	1500 (±150)	2000 (+1000)	1200 (±100)
Back Face Spall	2200 (±200)	2380 (+620)	1900 (±200)

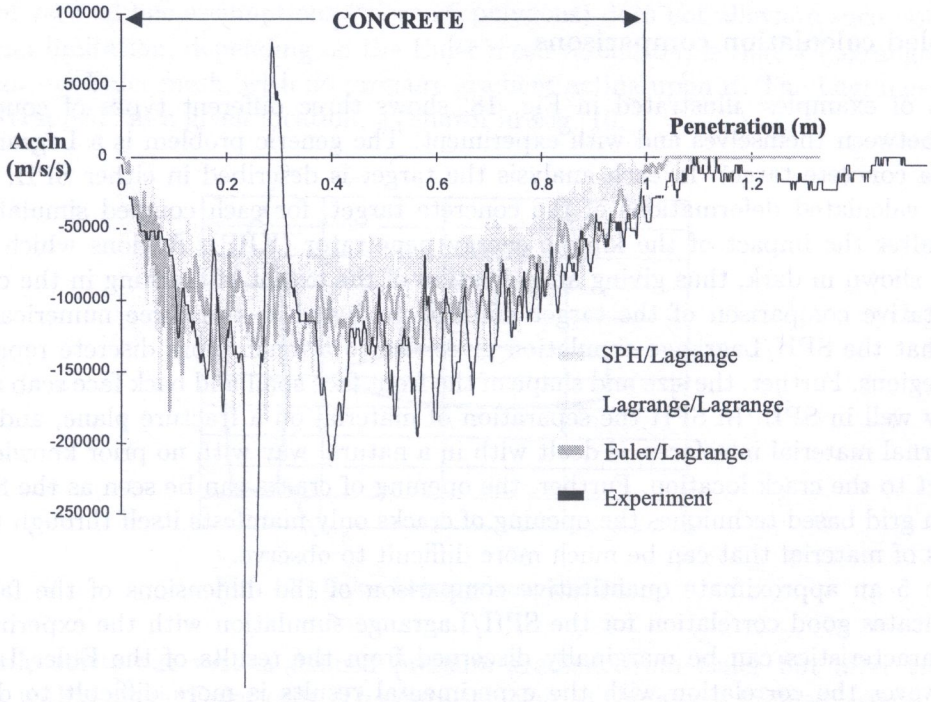


Fig. 19. Projectile deceleration histories

In the experiment, the deceleration of the kinetic energy penetrator was measured via an onboard accelerometer. For comparison, the average deceleration of the penetrator was recorded in each of the numerical simulations. These are compared in Fig. 19.

The general characteristic features of the KEP deceleration for the three coupled techniques show good correlation with those observed in the trial. The average peak deceleration is however under-predicted by 20% to 30%. Further, the under prediction of the peak deceleration is greater for the SPH target compared with the Lagrange and Euler targets. (It should be noted that the spike in the trial KEP deceleration time history at approximately 0.25 m penetration is thought due to be a characteristic of the accelerometer rather than physical phenomena.)

4. Conclusion

Different numerical solvers have certain advantages and limitations. It is critical that analysts understand which solver, or combination of solvers, is best suited to modeling their problems of interest. Utilization of software that allows general coupling of different solvers provides a framework within which to apply solvers optimal for the various domains of a problem. Nevertheless, coupling techniques, while extremely powerful, also have inherent limitations that must be recognized. The choice of using Lagrange–Lagrange, Euler–Lagrange, or SPH–Lagrange coupled techniques still, many times, is only resolved by numerical experimentation and validation against experiment.

References

[1] A.A. Amsden, H.M. Ruppel, C.W. Hirt. *SALE: A Simplified ALE Program for Fluid Flow at all Speeds*, Los Alamos Report, LA-8095, 1980.

[2] T. Belytschko *et al.*. Meshless Methods: An Overview and Recent Developments. *Computer Methods in Applied Mechanics and Engineering*, **139**: 3–47, 1996.

[3] N.K. Birnbaum, J. Tancreto, K. Hager. Calculation of blast loading in the high performance magazine with AUTODYN-3D. *26th DoD Explosives Safety Seminar*, Miami, FL, 1994.

[4] N.K. Birnbaum, *et al.* Analysis of blast loads on commercial buildings. *14th International Symposium, Military Aspects of Blast and Shock*, Las Cruces, New Mexico, 1995.

[5] N.K. Birnbaum. *AUTODYN Users Manual.* Century Dynamics, 1997.

[6] R.A. Clegg, A.J. Sheridan, C.J. Hayhurst, N.J. Francis. The application of SPH techniques in AUTODYN-2D to kinetic energy penetrator impacts on multi-layered soil and concrete targets. *8th International Symposium on Interaction and Effects of Munitions with Structures*, Virginia, USA, 1997.

[7] C.J. Hayhurst, R.A. Clegg. Cylindrically symmetric SPH simulations of hypervelocity impacts on thin plates. *Int, J. Impact Engng., Hypervelocity Impact Symposium*, Freiburg, Germany, 1996.

[8] C.J. Hayhurst, R.A. Clegg. The application of SPH techniques in AUTODYN-2D to ballistic impact problems. *16th International Symposium on Ballistics*, San Francisco, CA, 1996.

[9] C.J. Hayhurst, R.A. Clegg. *AUTODYN SPH Tutorial and User Manual.* Century Dynamics, 1997.

[10] N.E. Hoskin. *AUTODYN Theoretical Manual.* Century Dynamics, 1997.

[11] G.R. Johnson. Computational approaches for penetration problems. *8th International Symposium on Interaction of the Effects of Munitions with Structures*, McLean, Virginia, 1997.

[12] H.U. Mair. *Shock and Vibration Computer Programs, Hydrocodes for Structure/Medium Interaction*, SAVIAC SVM-13, 1995.

[13] A.J. Piekutowski. *Formation and Description of Debris Clouds Produced by Hypervelocity Impact.* NASA Contractor Report 4707, 1996.

[14] J.W. Swegle, S.W. Attaway. On the feasibility of using smooth particle hydrodynamics for underwater explosion calculations. *Computational Mechanics*, **17**: 151–168, 1995.