UCRL- 92385
PREPRINT

CIRCULATION COPY
SUBJECT TO RECALL
IN TWO WEEKS

FREE-LAGRANGE METHODS FOR COMPRESSIBLE HYDRODYNAMICS IN TWO SPACE DIMENSIONS

W. P. Crowley

This paper was prepared for submittal to
Free-Lagrangian Methods Conference
held March 4-6, 1985, at the
Sea Pines Plantation,
Hilton Head Island, South Carolina

March 1, 1985

Lawrence Livermore National Laboratory

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

FREE-LAGRANGE METHODS FOR COMPRESSIBLE HYDRODYNAMICS IN TWO SPACE DIMENSIONS

W. P. Crowley
Lawrence Livermore National Laboratory, University of California
P.O. Box 808
Livermore, CA 94550

ABSTRACT

Since 1970 a research and development program in Free-Lagrange methods has been active at Livermore. The initial steps were taken with incompressible flows for simplicity. Since then the effort has been concentrated on compressible flows with shocks in two space dimensions and time. In general, the line integral method has been used to evaluate derivatives and the artificial viscosity method has been used to deal with shocks.

Basically, two Free-Lagrange formulations for compressible flows in two space dimensions and time have been tested and both will be described. In method one, all prognostic quantities were node centered and staggered in time. The artificial viscosity was zone centered. One mesh reconnection philosophy was that the mesh should be optimized so that nearest neighbors were connected together. Another was that vertex angles should tend toward equality. In method one, all mesh elements were triangles.

In method two, both quadrilateral and triangular mesh elements are permitted. The mesh variables are staggered in space and time as suggested originally by Richtmyer and von Neumann. The mesh reconnection strategy is entirely different in method two. In contrast to the global strategy of nearest neighbors, we now have a more local strategy that reconnects in order to keep the integration time step above a user chosen threshold. An additional strategy reconnects in the vicinity of large relative fluid motions.

Mesh reconnection consists of two parts: (1) the tools that permit nodes to be merged and quads to be split into triangles etc. and; (2) the strategy that dictates how and when to use the tools. Both tools and strategies change with time in a continuing effort to expand the capabilities of the method. New ideas are continually being tried and evaluated. The successful ones stay in the code, and in some sense its intelligence increases with time.

Examples of mesh optimization tools and strategies and of sample problems will be given.

1. Introduction

The Free Lagrange effort at Livermore has been active since 1970 and has moved sequentially in three different directions. The initial effort was confined to incompressible flows [1]. During this time basic ideas were explored such as initial mesh construction and mesh reconnection during the evolution of a problem. It was soon realized that initial mesh generation and mesh reconnection were separable problems. In this paper we will assume that a mesh exists and concentrate on mesh reconnection.

Another problem that must be faced involves the centering of prognostic variables in space and time. In the initial incompressible effort and in the first compressible effort all prognostic variables were node centered. This has the potential advantage that mesh reconnection will not change the values of prognostic variables - that is, mesh reconnection does not introduce diffusion. Node centering has difficulties when it comes to multi-material problems. The notion of material interfaces, if done correctly, introduces complexities. In the second effort, material interfaces were handled in a rather crude way and this led to somewhat crude results.

The third and current effort solves the material interface problem by recentering the variables according to the traditional Lagrangian centering of Richtmyer and von Neumann [2]. Energy and mass are zone centered - position, velocity and acceleration are node centered. With this approach, material interfaces are well defined; they fall on zone edges (along lines connecting nodes). The disadvantage is that mesh reconnections now introduce a mixing of adjacent zone attributes and thus diffusion occurs. However since mesh reconnection is now more local than global, it is thought that the effect of diffusion is small.

Shock waves are an important aspect of compressible flows. In these formulations they are automatically taken care of by an artificial viscosity [2,3].

Spatial derivatives are done with the Line Integral method [4].

In the third effort the mesh reconnection philosophy changed along with the centering of variables. In the first two efforts, mesh reconnection was a global affair with the goal in mind of connecting nearest neighbors. In the third effort the strategy is threefold: (1) to accommodate shearing motions; (2) to keep the integration time step up and; (3) to prevent boomerang and bowtie zone topologies.

Since all these formulations use an explicit scheme for time differencing, the integration time step is an important consideration.

The remainder of this paper is divided into six sections. Section 2 gives the equations of motion. Section 3 describes the line integral method with particular application to the momentum equation. In section 4 the first compressible effort will be described. In section 5 the current effort will be described and in section 6 mesh reconnection and optimization will be described. An interactive

graphics capability is quite important to the development of these methods. Comments on interactive graphics are in section 7. Results from a sample test problem are described in section 8.

2. Equations of Motion

In traditional Lagrangian codes, each node has an invariant set of neighbors and the spatial difference terms usually involve the original coordinates and a Jacobian that transforms from orginal to current coordinates. In many cases the mesh elements (zones) are quadrilaterals.

In this formulation, the mesh elements are both triangles and quadrilaterals and the number of neighbors of each node may change with time. It is more convenient to express derivatives in terms of current coordinates; Pomraning [5] refers to this formulation as the Modified Eulerian and to the traditional formulation as the Lagrangian.

The equations of motion are

1. $\frac{d\vec{u}}{dt} = -\frac{1}{\rho}(\text{grad } p + \text{div } \bar{\bar{q}})$ Momentum

2. $\frac{d\rho V}{dt} = 0$ Mass

3. $\frac{d\varepsilon}{dt} = -\frac{p}{\rho}\,\text{div}(\vec{u}) - \frac{\bar{\bar{q}}}{\rho}\,\text{grad}(\vec{u})$ Specific Internal energy

To these are appended an equation of state

$$p = p(\varepsilon, \rho)$$

a kinematic equation

$$\frac{d\vec{x}}{dt} = \vec{u}$$

and an equation for the artificial viscosity

$$q = \begin{cases} \rho\Delta u(\Delta u - C_1 C) & \text{if } \Delta u < 0 \\ 0 & \text{otherwise.} \end{cases}$$

The artificial viscosity actually used is somewhat more complex in 2D - different forms are required by quadrilateral and by triangular zones. The tensor $\bar{\bar{q}}$ used here is in a state of continual development. In addition, an optional $qk\ell$ [6] may be used to inhibit nonphysical small scale motions in quads.

The equations are solved in a cylindrical geometry with

$$\vec{x} : (r,z)$$

$$\vec{u} : \left(\frac{dr}{dt}, \frac{dz}{dt}\right)$$

3. The Line Integral Method

Green's Theorem in the plane states

$$\iint \left(\frac{\partial P}{\partial x} + \frac{\partial Q}{\partial y}\right) dx\, dy = \oint (P\, dy - Q\, dx)$$

where P and Q are arbitrary functions of x and y. Taking first P to be zero and then Q we have

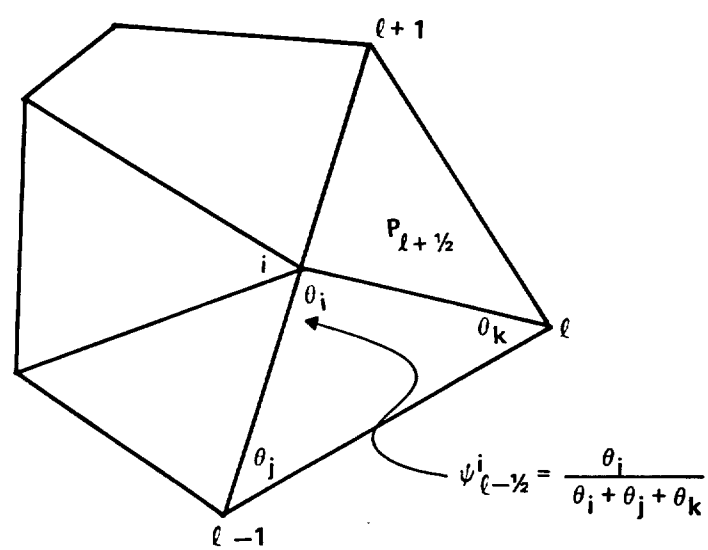
$$\frac{\partial P}{\partial x} = \frac{\oint p\, dy}{\oint x\, dy}$$

$$\frac{\partial P}{\partial y} = \frac{-\oint p\, dx}{\oint x\, dy}$$

where $\frac{\partial P}{\partial x}$ and $\frac{\partial P}{\partial y}$ are mean values over the area $\oint x\, dy$. In finite difference form we have

$$\oint p\, dy \approx \sum P_{\ell-1/2}(y_\ell - y_{\ell-1})/2 \tag{3.1}$$

The path of integration connects the mid points of the edges - it does not include points in the zones because $P_{\ell-1/2}$ is assumed constant in a zone.



In the momentum equation we need the gradient multiplied by the reciprocal of the density. The denominator becomes

$$\rho \oint x \, dy \approx \sum_{\ell} \rho_{\ell-1/2} \, \psi^{i}_{\ell-1/2} \, A_{\ell-1/2} \tag{3.2}$$

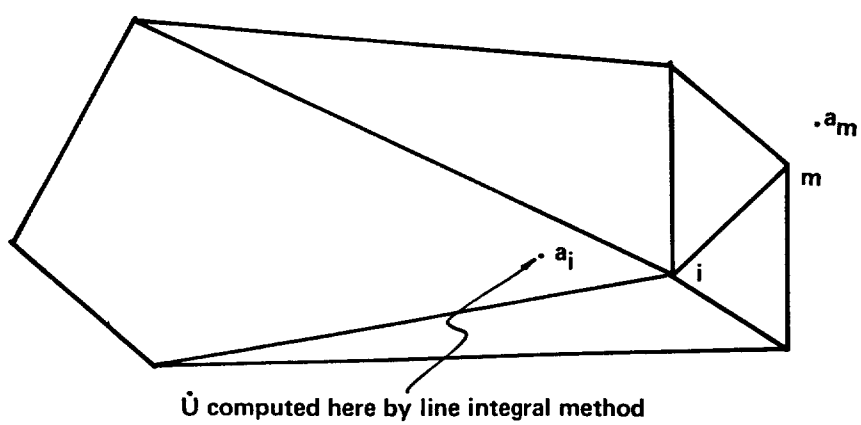
where $A_{\ell-1/2}$ is the area of zone ℓ-1/2 and where $\psi^{i}_{\ell-1/2}$ is the ratio of angle (ℓ-1,i,ℓ) to the sum of all angles in zone ℓ-1/2. The quantity $\psi^{i}_{\ell-1/2} A_{\ell-1/2}$ is thus the fractional contribution of zone ℓ-1/2 to node i.

The r-component of acceleration is the ratio of equations (3.1) and (3.2) and similarly for the z-component.

The acceleration computed above is in general not centered at node i, but rather at some other point, say a_i. The r-component of acceleration at node i is

$$\dot{u}_i = \dot{u}_{a_i} + (\text{grad } \dot{u}) \cdot (\vec{x}_i - \vec{x}_{a_i}) \, .$$

and similarly for the z-component. A similar correction for velocity rather than acceleration was suggested by Margolin and Nichols [7]. Here the term grad $\dot{u}$ is evaluated by a line integral ratio over a path including the neighbors of node i; the path follows the points a_ℓ.



Work on this correction term is in progress now. At first glance, the point a_i should be at the average centroid of the zones surrounding node i.

4. The Node-Centered Compressible Formulation

In this formulation all mesh elements are triangles and all prognostic variables are point-centered quantities. However, at the beginning of each cycle and after optimization, temporary zone-centered energies and masses are computed by a mapping process. Once these quantities are computed, the traditional Lagrangian centered difference equations [2] are used. At the end of the cycle, the point-centered

quantities have been advanced in time, and all zone-centered information is forgotten. The mapping process will be described here.

First it is necessary to introduce some notation and some conventions. There are NPTOT mesh points and each point is connected to its nearest neighbors to give triangular mesh elements. Each point has NUM neighbors where

$$\text{NUM} \geq 2 \text{ for boundary points}$$

$$\text{NUM} \geq 3 \text{ for interior points.}$$

The three vertices i, j, and k of each triangle are ordered in a counterclockwise sense in our right-handed (x,y) coordinate plane, Fig. 1a. Since j and k are two of the neighbors of i, i is a neighbor of j and of k.

The complete neighborhood of point i consists of several triangles. For example, a typical interior point is sketched in Fig. 1b: point i has five neighbors and is surrounded by five triangles. The first neighbor of this point may be any of its neighbors, but the others must follow in counterclockwise order. A typical boundary point is sketched in Fig. 1c. There, point i has four neighbors and three triangles. Interior points have NUM neighbors and NUM neighboring triangles while boundary points have NUM neighbors and NUM-1 neighboring triangles. We are more particular about the organization of boundary neighborhoods. The first and last neighbors of boundary points are also boundary points, and the counterclockwise ordering of the neighbors means that a path from the first to the last neighbor lies within the boundaries of the problem.

In referring to the neighbors of a point, we may use the local index ℓ where

$$1 \leq \ell \leq \text{NUM} .$$

Using this, we may refer to a triangle by its "proper" name (i,j,k) or by its first name and two indices $(i,\ell,\ell+1)$. It is then natural to denote point quantities

$$f^i \text{ or } f_i$$

and zone quantities

$$f_{\ell+1/2} .$$

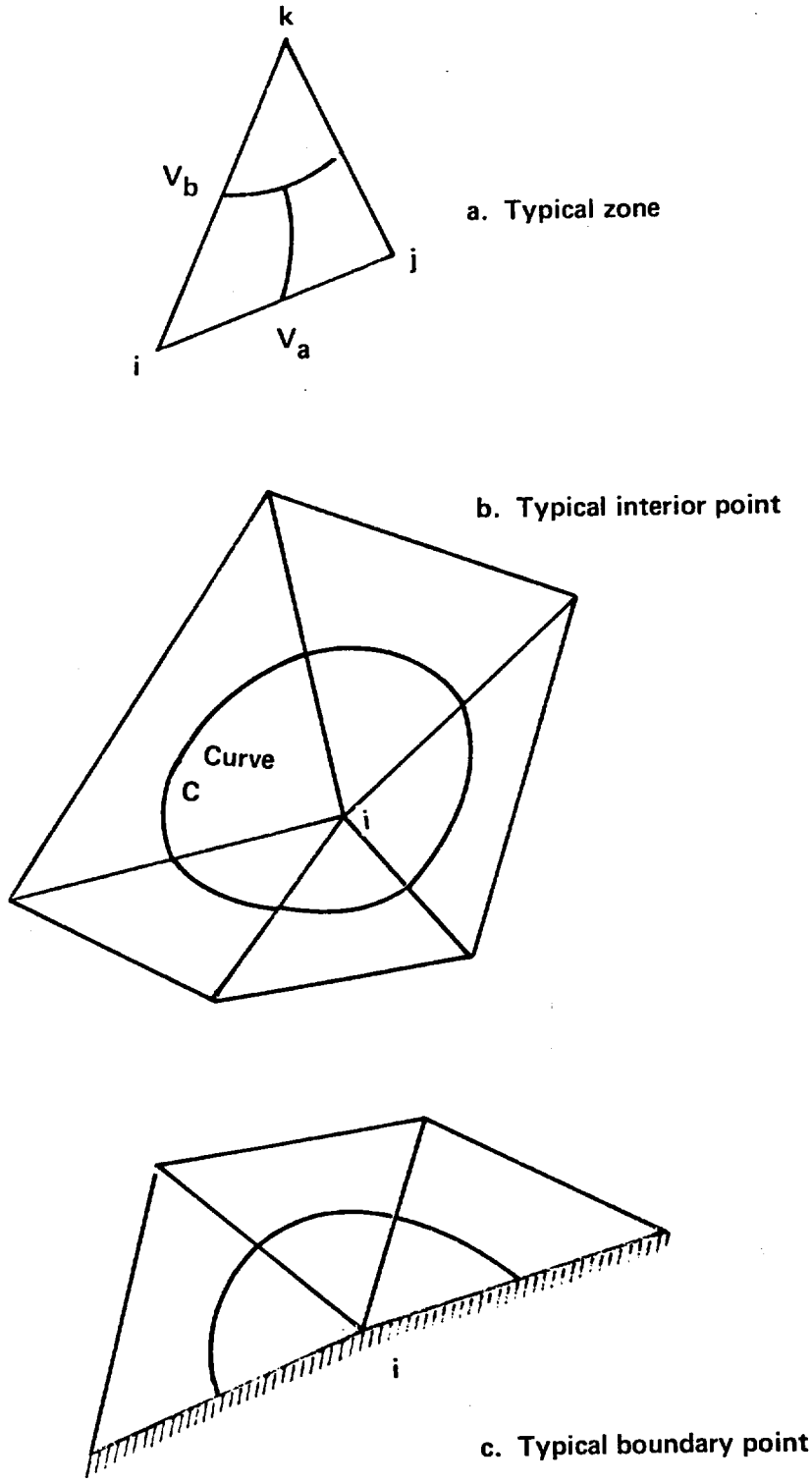
When time is introduced, we will use an additional superscript, and denote a mixed quantity at time level n

$$(f^i_{\ell+1/2})^n$$

or a point quantity at time level n

$$(f_i)^n$$

In general, the time index is suppressed implying that latest available values are used.



REPRESENTATIVE GEOMETRIC SKETCHES

Figure 1

Consider the interior mesh point i in Fig. 1b. It is surrounded by five triangles each of which has a definite area and volume. The curve C defines the area of the secondary mesh element surrounding the point i. That is, we subdivide each triangle into three parts associating one part with each vertex of the triangle. If $A_{\ell+1/2}$ is the area of triangle (i,j,k) (Fig. 1a) then

$$a^i_{\ell+1/2} = \psi^i_{\ell+1/2} A_{\ell+1/2}$$

is the area that point i and zone $\ell+1/2$ have in common. The product $\pi\psi^i_{\ell+1/2}$ is the angle in radians between side $\overline{ij}$ and $\overline{ik}$.

An indefinite summation notation is used and is to be interpreted in the following manner. The zone area is given by

$$A_{\ell+1/2} = \sum_{ijk} a^i_{\ell+1/2}$$

and the sum is over the three vertices i, j, and k associated with zone $\ell+1/2$. The point area is given by

$$A^i = \sum_{\ell} a^i_{\ell+1/2}$$

and the sum is over all zones $\ell+1/2$, (ℓ=1, L-1) surrounding a point where, if NUM is the number of neighbors for point i,

$$L = \begin{cases} \text{NUM for interior zones} \\ \\ \text{NUM-1 for boundary zones} \end{cases}$$

Similarly, there is a volume associated with each zone and with each point

$$S_{\ell+1/2} = \bar{r}_{\ell+1/2} A_{\ell+1/2} = \left(\frac{r^i + r^j + r^k}{3}\right) A_{\ell+1/2}$$

$$s^i_{\ell+1/2} = \bar{r}^i_{\ell+1/2} a^i_{\ell+1/2}$$

$$S^i = \sum_{\ell} s^i_{\ell+1/2}$$

where S is the volume of revolution divided by 2π and where

$$\bar{r}^i_{\ell+1/2} = \begin{cases} (\Omega_{\ell+1/2} r^i + \zeta)\, \bar{r}_{\ell+1/2} & \text{(cylindrical geometry)} \\ \\ 1 & \text{(plane geometry)} \end{cases}$$

and

$$\Omega_{\ell+1/2} = \frac{1 - \zeta}{\sum_{ijk} r^i \psi^i_{\ell+1/2}}$$

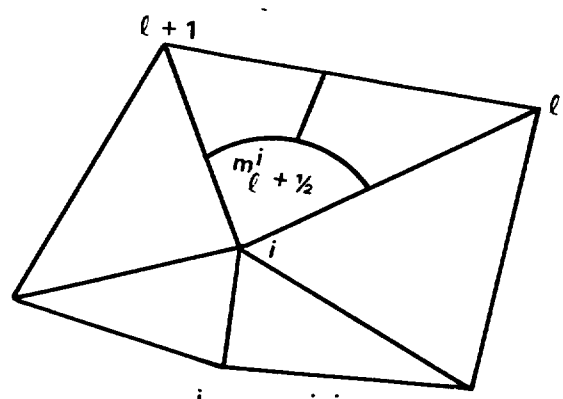
The constant ζ is taken to be 7/12 to give the correct volumes for equilateral triangles. We define the geometric factor

$$h^i_{\ell+1/2} = \bar{r}^i_{\ell+1/2}\ \psi^i_{\ell+1/2}$$

and the mapping factor

$$f^i_{\ell+1/2} = h^i_{\ell+1/2} \Big/ \sum_{\ell} h^i_{\ell+1/2}\ .$$

The mass contribution from point i to zone ℓ+1/2 is (Fig. 2)



$$m^i_{\ell+½} = m^i f^i_{\ell+½}$$
$$f^i_{i+½} = f(\vec{x}_i, \vec{x}_\ell, \vec{x}_{\ell+1})$$

Figure 2

$$m^i_{\ell+1/2} = m^i\ f^i_{\ell+1/2}$$

and the mass in zone ℓ+1/2 is the sum of the three vertex contributions

$$m_{\ell+1/2} = \sum_{ijk} m^i_{\ell+1/2}\ .$$

The specific internal energy in zone ℓ+1/2 is

$$\epsilon_{\ell+1/2} = \frac{1}{m_{\ell+1/2}} \sum_{ijk} \epsilon^i\ m^i_{\ell+1/2}\ .$$

and the density is

$$\rho_{\ell+1/2} = m_{\ell+1/2}/S_{\ell+1/2}\ .$$

Once the energy and density have been determined, the pressure is determined through the equation of state $p = p(\epsilon,\rho)$.

These mapping equations hold for boundary zones (Fig. 1c) as well as interior zones.

The mapping described here is conservative in that the sum of m^i or ε^i over all the points is identical to the sum of $m_{\ell+1/2}$ or $\varepsilon_{\ell+1/2}$ over all the zones.

Materials, material properties, and equations of state (EOS) are associated with points. When zone quantities are constructed, it is necessary to associate an EOS with the zone. If the three vertex points are all the same material then the zone is of that material. That is, the EOS of the zone is the same as the EOS of the points. If all vertex points do not have the same EOS, then the EOS of the zone is that of the material contributing the most mass to the zone.

At the beginning of a cycle we have

$$x_i^n,\ y_i^n,\ u_i^{n-1/2},\ w_i^{n-1/2},\ \varepsilon_i^n,\ \rho_i^n,\ m_i\ .$$

First zone centered energies, densities and pressures are computed

$$\varepsilon_{\ell+1/2}^n,\ \rho_{\ell+1/2}^n,\ P_{\ell+1/2}^n$$

and a zone centered artificial viscosity is constructed

$$q_{\ell+1/2}^n = q\left(\rho_{\ell+1/2}^n,\ \Delta\vec{u}_{\ell+1/2}^{n-1/2}\right)\ .$$

From these quantities a pressure gradient can be calculated, and this gives the two components $(\dot{u}_i^n,\ \dot{w}_i^n)$ of the acceleration vector by the line integral method.

The acceleration provides an increment in the velocities

$$\vec{u}_i^{n+1/2} = \vec{u}_i^{n-1/2} + \dot{\vec{u}}_i^n\ \Delta t^n$$

and this results in an advanced value for the coordinates

$$\vec{x}_i^{n+1} = \vec{x}_i^n + \vec{u}_i^{n+1/2}\ \Delta t^{n+1/2}$$

and thus a new value for the volume of each zone

$$S_{\ell+1/2}^{n+1} = \bar{r}_{\ell+1/2}^{n+1}\ A_{\ell+1/2}^{n+1}\ .$$

The zonal energy change is then computed based upon the work term which includes both p and q, using a time centered pressure, $p^{n+1/2}$. Time centering this term is important to conserve energy. For general equations of state it is done with one iteration involving the EOS and the energy equation.

Energy changes are mapped back to the node quantities by defining the spatial sum

$$(md\varepsilon)_i^n = \sum_{\ell} \left(m_{\ell+1/2}\ \varepsilon_{\ell+1/2}\ h_{\ell+1/2}^i / \bar{r}_{\ell+1/2}\right)^n$$

Then

$$\varepsilon_i^{n+1} = \varepsilon_i^n + [(md\varepsilon)_i^{n+1} - (md\varepsilon)_i^n]/m_i$$

Care must be taken in updating the nodal density. If it is done with a strict volume change, or by a simple mapping from zone to node, then the density may change due to mesh optimization. This is undesirable because such changes will cause pressure changes which will cause mesh motion. We want the density to change only due to fluid motions and this is done with the notion of pseudo volumes.

Let m_i and S_i^n be the nodal mass, and pseudo-volume of node i at the beginning of a cycle. At problem generation time $J_i^n = S_i^n$.

First we reconnect the mesh and then we compute J_i^n, the actual volume, after mesh reconnection but before hydro. After hydro, the actual volume is J_i^{n+1} and the pseudo volume is

$$S_i^{n+1} = S_i^n + J_i^{n+1} - J_i^n$$

The new density is $\rho_i^{n+1} = \rho_i^n \, S_i^n/S_i^{n+1}$. With this approach the mass of a node remains constant and the density changes only due to fluid motions.

5. Staggered Mesh Compressible Formulation

In this formulation mesh elements are a mixture of quads and triangles and the prognostic variables are staggered in space and time as suggested by von Neumann and Richtmyer [2]. The variables are

$$\vec{x}_j^n,\ \dot{\vec{u}}_j^n,\ \vec{u}_j^{n+1/2},\ \rho_z^n,\ \varepsilon_z^n,\ EOS_z^n,\ Q_z^{n+1/2}$$

where a subscript z designates a zonal quantity.

The calculation is advanced in time as follows:

1. Mesh optimization

2. $Q^{n-1/2}\ (\rho^n,\ \Delta u^{n-1/2})$

3. $\dot{\vec{u}}_j^n$

3a. Apply boundary conditions

4. $\vec{u}_j^{n+1/2}$

5. $x_j^{n+1} \rightarrow S_z^{n+1}$

6. $\rho^{n+1} = \rho^n \, S^n/S^{n+1}$

7a. $\delta\tilde{\varepsilon}^{n+1} = \varepsilon(p^n, Q^{n-1/2}, u^{n+1/2})$

7b. $\tilde{p}^{n+1} = p(\tilde{\varepsilon}^{n+1}, \rho^{n+1})$

7c. $\delta\varepsilon^{n+1} = \varepsilon(p^n, \tilde{p}^{n+1}, Q^{n-1/2}, u^{n+1/2})$

8. $\varepsilon^{n+1} = \varepsilon^n + \delta\varepsilon^{n+1}$

Δt

We start with a mesh that consists primarily of quads. This has the positive aspect of reducing the number of zones by a factor of two and thus of reducing memory requirements and of speeding up the calculation by a factor near two.

Experimentally we have found that in calculating the growth of Meshkov-Richtmyer instabilities, triangular zones tend to stiffen the mesh and to reduce the growth rate significantly. On the other hand, quads can tangle with resulting bowties and boomerangs which may result in negative volumes.

As a problem runs and the fluid distorts, mesh optimization tends to introduce triangular zones as will be discussed in section 6.

6. Mesh Reconnection and Optimization

With this Free-Lagrange algorithm, the nodes are reconnected subject only to a few constraints. Mainly we require that the area of each zone be positive and that the connectivity be symmetric. The first concern is that any small area of the physical domain of the problem be covered uniquely by zones. The symmetry concern helps to conserve momentum.

The mesh may be reconnected in two modes: (1) by an interactive graphics code that operates on a restart dump and attempts to reconnect as directed by a human; (2) automatically by the main code at execution time. Comments on the first mode of operation will be found in section 7. The second mode requires a predetermined strategy that is determined by a set of user controlled parameters. We develop this strategy and the options it employs through experimenting with both modes 1 and 2. Each new problem brings with it new challenges for the optimization strategy.

We learn how to run complex problems by a trial and error procedure - by trying out new and different mesh optimization options for example. Older options are rarely discarded in favor of new options - they are reprioritized relative to the new options in terms of when and if they are used. The prioritized collection of mesh reconnection options is called an optimization strategy. Thus the number of mesh optimization options increases with age as does the complexity of the mesh optimization strategy. This growth permits more and more complex fluid motions to be treated automatically by the code as time passes. The analogy with a growing child is impossible to avoid.

Different mesh situations may call for different options and the user can influence the optimization priorities with a set of parameters.

Mesh optimization involves reconnecting nodes together in order to enhance the calculation in some sense. In the first and second versions of the code the primary strategy had to do with connecting nearest neighbors together. In the third version we are more concerned with keeping Δt up. In some sense we've moved from a global to a local strategy.

In this algorithm, mesh optimization or reconnection takes place for three reasons: (1) we try to prevent the mesh from getting into unsuitable topologies such as bowties and boomerangs (a non-problem with triangular mesh elements); (2) when Δt falls below a user defined threshold, we change the mesh locally to improve Δt; (3) mesh reconnection takes place along material interfaces when one material wants to slide along another.

In terms of optimizing Δt and accommodating shearing motions this manifestation of the Free-Lagrange method may be classified as an adaptive mesh scheme.

A simple example of mesh optimization is based on the notion of nearest neighbors. We want each node to be coupled only to those nodes close to it because in some sense this improves the accuracy. Assume all mesh elements are triangles and that a mesh exists. Figure 3a shows the relation between mesh elements $ik\ell$ and ijk - they have a common side ik which is a diagonal of quadrilateral $ijk\ell$. Their connections to the rest of the mesh are not shown. Since every quad has two diagonals it is possible to do a local reconnection by breaking connection ik and making connection $j\ell$ as shown in Fig. 3b. That is, we disconnect one diagonal and connect the other. Nodes i and k each lose a neighbor while nodes j and ℓ each gain a neighbor. The number of nodes and zones are conserved. Both new zones are allowed since their areas are positive. This reconnection is called a "FLIP".

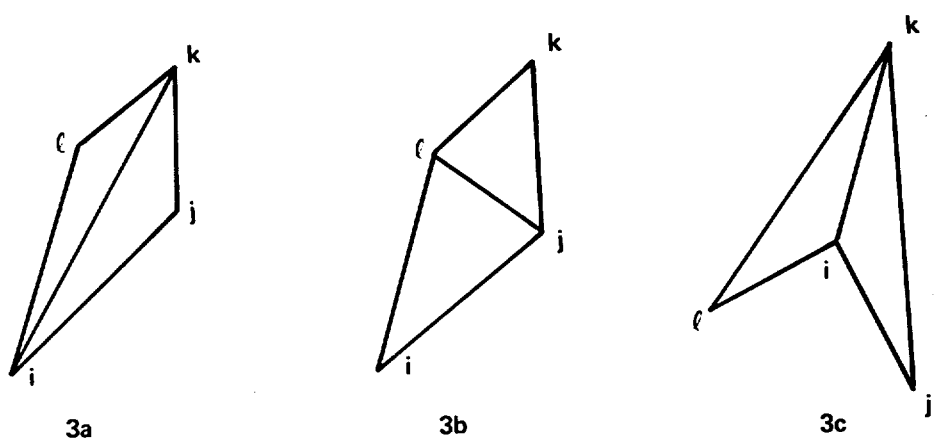


Figure 3

Considering Fig. 3c, we see that a constraint on the FLIP option is necessary. Distance $j\ell$ is less than distance ik but a FLIP would result in zone $ij\ell$ having a negative area. The FLIP could possibly still take place if the next layer of

neighbors were considered, but the loss of simplicity for a small gain makes this extension unattractive. Other constraints can also be applied - for example, we want the area of the smallest new zone to be greater than some fraction of the sum of the two areas (typically 20%).

The FLIP operation is local but the notion of a nearest neighbor mesh is global. We construct a nearest neighbor mesh by sweeping over all mesh connections and considering a FLIP for each one. This is an iterative process and we continue making mesh passes until we find that no flips have occurred. We then have a nearest neighbor mesh.

The FLIP option can be used to optimize the mesh to states other than nearest neighbor. For example we might want to generate a mesh in which the variations in zone area is minimized and so flips would make/break connections so that the difference in the two new areas was less than the difference in the two old areas. Equilateral triangles minimize truncation errors and so FLIP could be made to minimize the angle variation in the new zones.

With a triangular mesh, the FLIP option was quite simple as illustrated at the top of Fig. 4. The introduction of quads means more flexibility in maximizing Δt with FLIP as is seen in the remainder of Fig. 4.

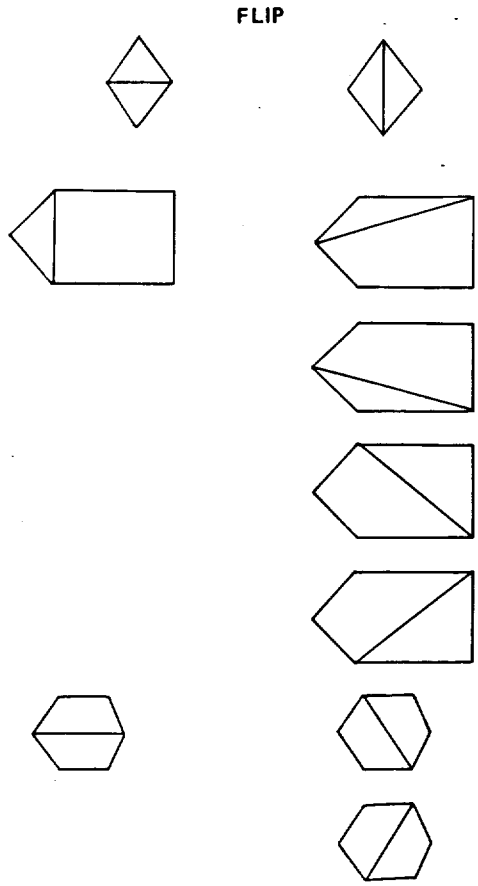


Figure 4

The FLIP option may be considered a tool and the mesh loop inside an iteration may be considered a strategy for accomplishing some goal (creating almost equilateral

triangles). Another strategy for using FLIP comes into play when we have large relative fluid motions (shearing motions for example).

When one fluid moves relative to another along a material interface, the mesh is reconnected by flips along the interface. This SLIP-FLIP operation is triggered when an angle (with its vertex node on the interface) exceeds 95°. Figure 5A shows a slip motion with SLIP-FLIP inhibited and Fig. 5B shows the same motion with SLIP-FLIP operational. In Fig. 5 the left material has moved upwards since the initial time when both fluids were at the same height.

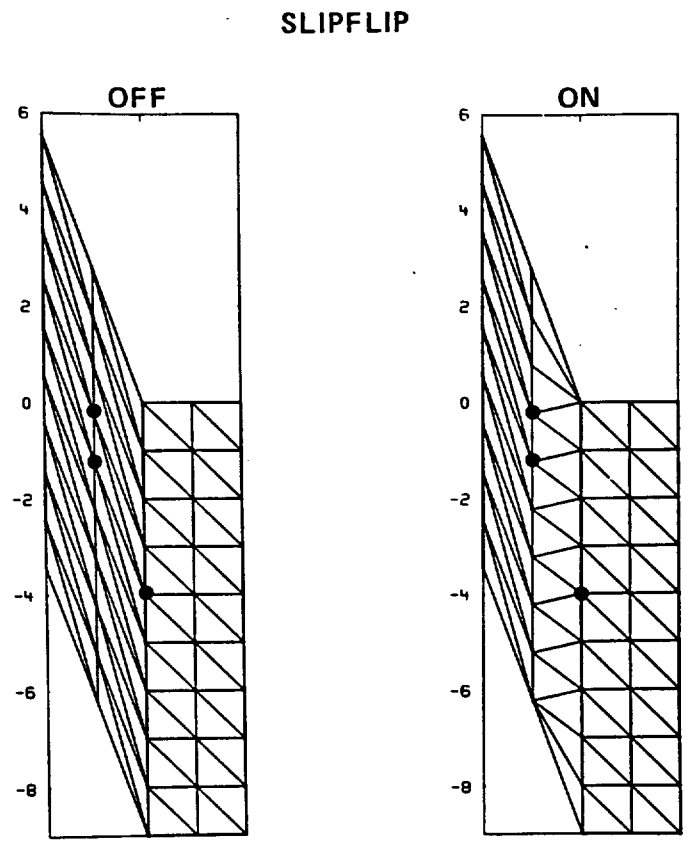


Figure 5

Compressible hydro codes using an explicit time centering scheme have a Δt limitation based on the Courant condition,

$$\Delta t < \min (L/C)$$

where C is the sound speed and L is the altitude of the triangle. As zones compress and distort L may decrease and C may increase to the point where economics dictates that the problem cannot continue due to a small Δt. At this point the problem dies or we invent ways to increase Δt by increasing L (C is not at our disposal). The primary tools for increasing Δt are FLIP, MERGE, NYM and ZAZ.

The MERGE option is used to merge two nodes of a triangle together when Δt in that zone falls below a user defined threshold as seen in Fig. 6. If nodes A and B are both interior nodes, the MERGE action destroys two zones and one node. A new

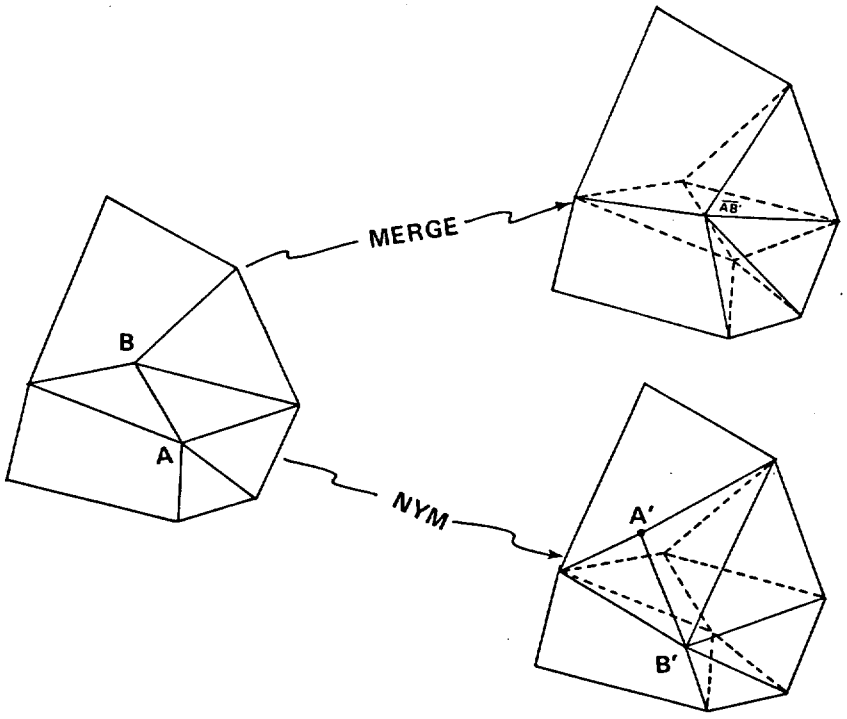


Figure 6

option, NYM (after Peter Roberts) tries to improve Δt by moving the nodes away from each other. The idea here is to increase Δt by increasing the critical length in the zone controlling Δt. Zones are not destroyed and it appears that NYM is an improvement over MERGE. As with other mesh optimization actions the price to be paid for NYM is a small amount of local diffusion because of the necessary overlay calculation that conserves mass and energy. The current strategy is to try NYM first and if it fails (e.g., it can't improve Δt enough) then to resort to MERGE.

The ZAZ option (Fig. 7) destroys a zone by moving the node opposite the longest edge over to the longest edge. This destroys one zone, but creates a new one by subdividing the zone on the opposite side of the longest edge. Since ZAZ subdivides a zone it may in fact cause Δt to decrease. In the overall Δt optimization strategy, it is therefore used only when all other tools have failed. In practice it may lower Δt, but it also loosens up the mesh structure so that MERGE or another option may fix the problem the following cycle.

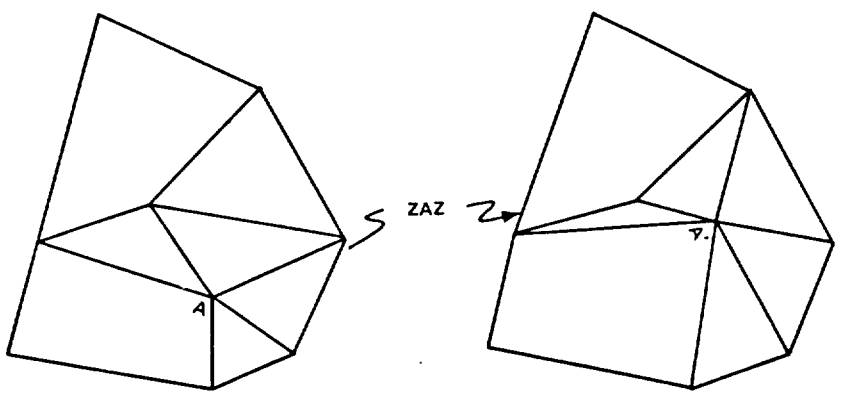


Figure 7

If a quad turns into a boomerang it is split into two triangles (Fig. 8). Similarly, if we project that a quad will bowtie in 3 cycles the quad is split into two triangles. We try to avoid SPLIT by doing a NYM operation first. If the NYM fails to fix the problem (possible because of geometric constraints on node motion) then we resort to SPLIT.

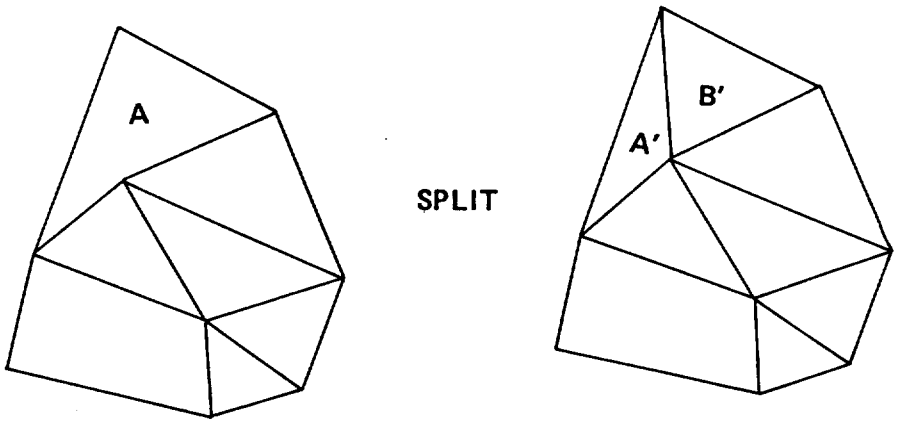


Figure 8

As a problem evolves, part of the mesh may become poorly resolved because nodes move away from each other - consider for example the growth of a perturbation on an unstable interface. The CREAP option (Fig. 9) subdivides zone edges (CREATE POINTS) when they become longer than a given value. Currently CREAP operates only on interfaces and boundaries.

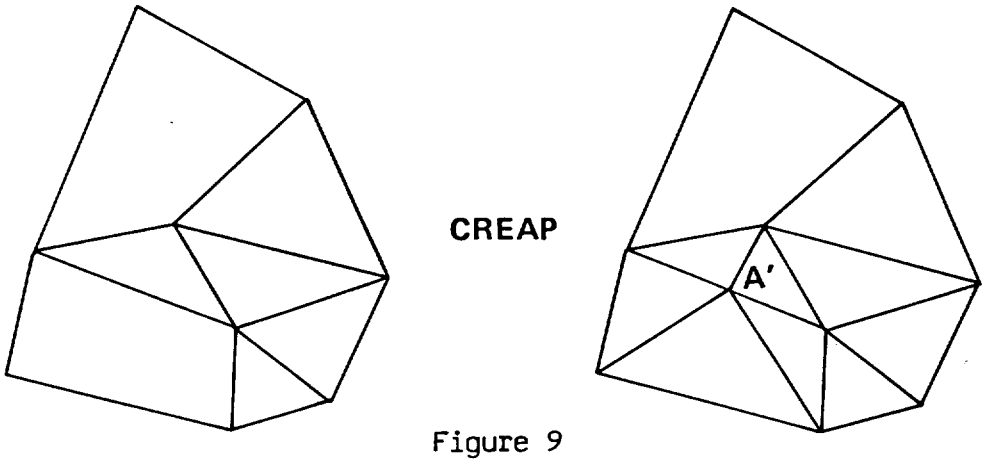


Figure 9

Two simplified examples of mesh strategy are shown in Figs. 10 and 11. In Fig. 10 we see a "prevent" strategy in which we are attempting to anticipate the occurrence of an undesirable situation (a bowtie here) and to prevent this situation from developing. In Fig. 11 we attempt to keep the integration time step up by first trying to NYM and then, if that fails, to merge zones whose current Δt falls below user defined thresholds. The complete mesh optimization strategy including SLIP-FLIP checks, checks on geometry and interfaces and so on is much more complex than these simple examples.

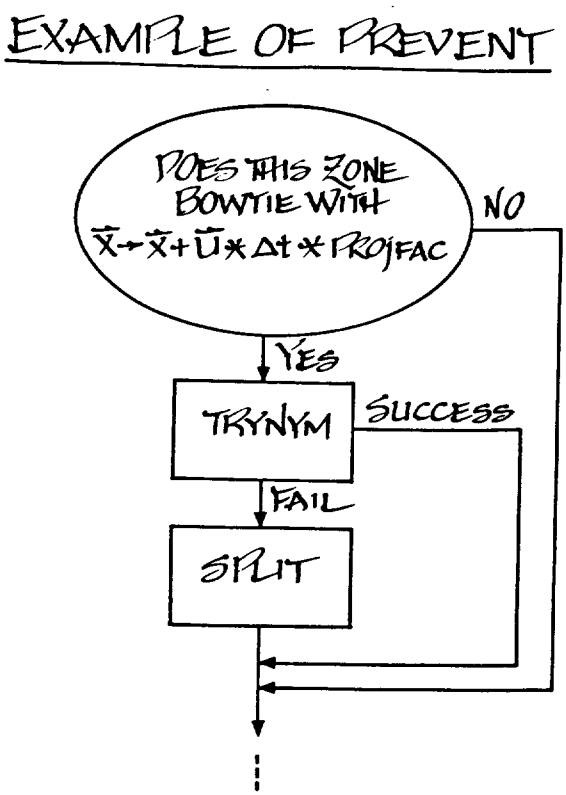


Figure 10

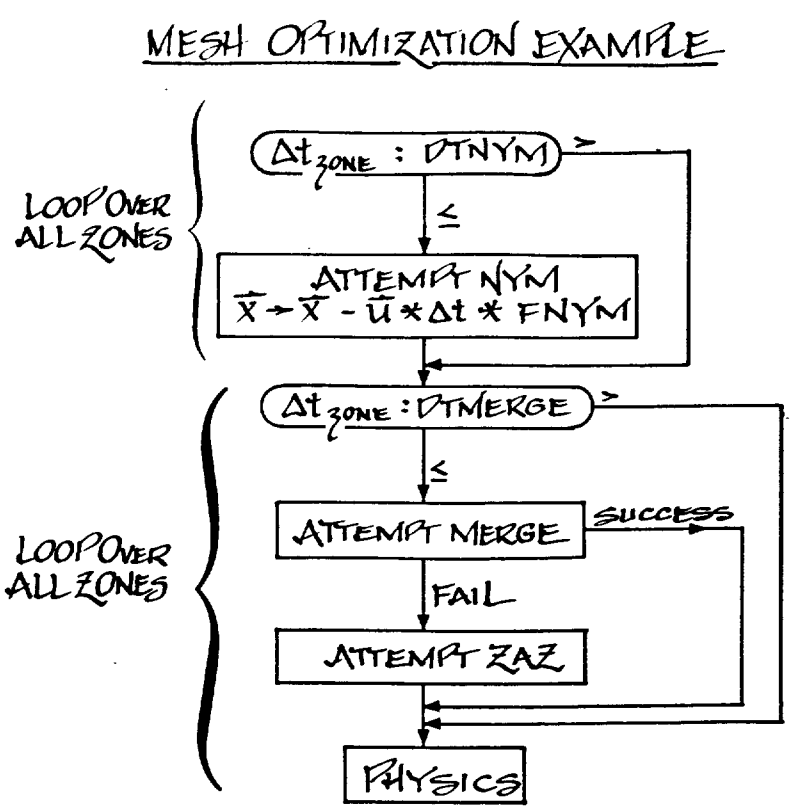


Figure 11

7. Interactive Graphics (IG)

As a problem runs it changes its shape and it is of some interest to the user to study this shape change. We can specify ahead of time that the hydro code produce graphics output as it runs but many times we do not know ahead of time what we want to see. As the problem runs we therefore leave a trail of restart dumps and we can use IG on these and get snapshots of any part of the problem.

Another important aspect of IG is that it allows humans to reconnect the mesh. Sometimes during the evolution of a problem, the mesh gets into a configuration that cannot be improved by automatic reconnections. This happens because the reconnection algorithm is not smart enough or because the wrong optimization strategy has been chosen by the user. In these situations, the user can fix the problem with IG because the human has a more global view of the mesh than does the reconnection algorithm. With more global information the human can cause mesh reconnections to happen that the hydro code has not been taught to consider. Sometimes this leads to an improvement in the hydro code; usually it results in a broken problem being fixed and being able to continue.

In the early development stages of the hydro code, the reconnection algorithm had very little intelligence. By using IG on a number of different problems, humans were able over a period of time to figure out how best to reconnect the mesh and were able to teach much of this to the hydro code in terms of expanding the mesh reconnection algorithm.

8. Results from Meshkov-Richtmyer Instability Growth

An example of instability growth will illustrate the staggered mesh scheme. In Fig. 12 we see the evolution of a perturbation on an interface that is accelerated by several shocks. In this Meshkov-Richtmyer situation the interface is unstable to shocks in both directions. The jump in perturbation velocity is proportional to the jump in fluid velocity due to the shock.

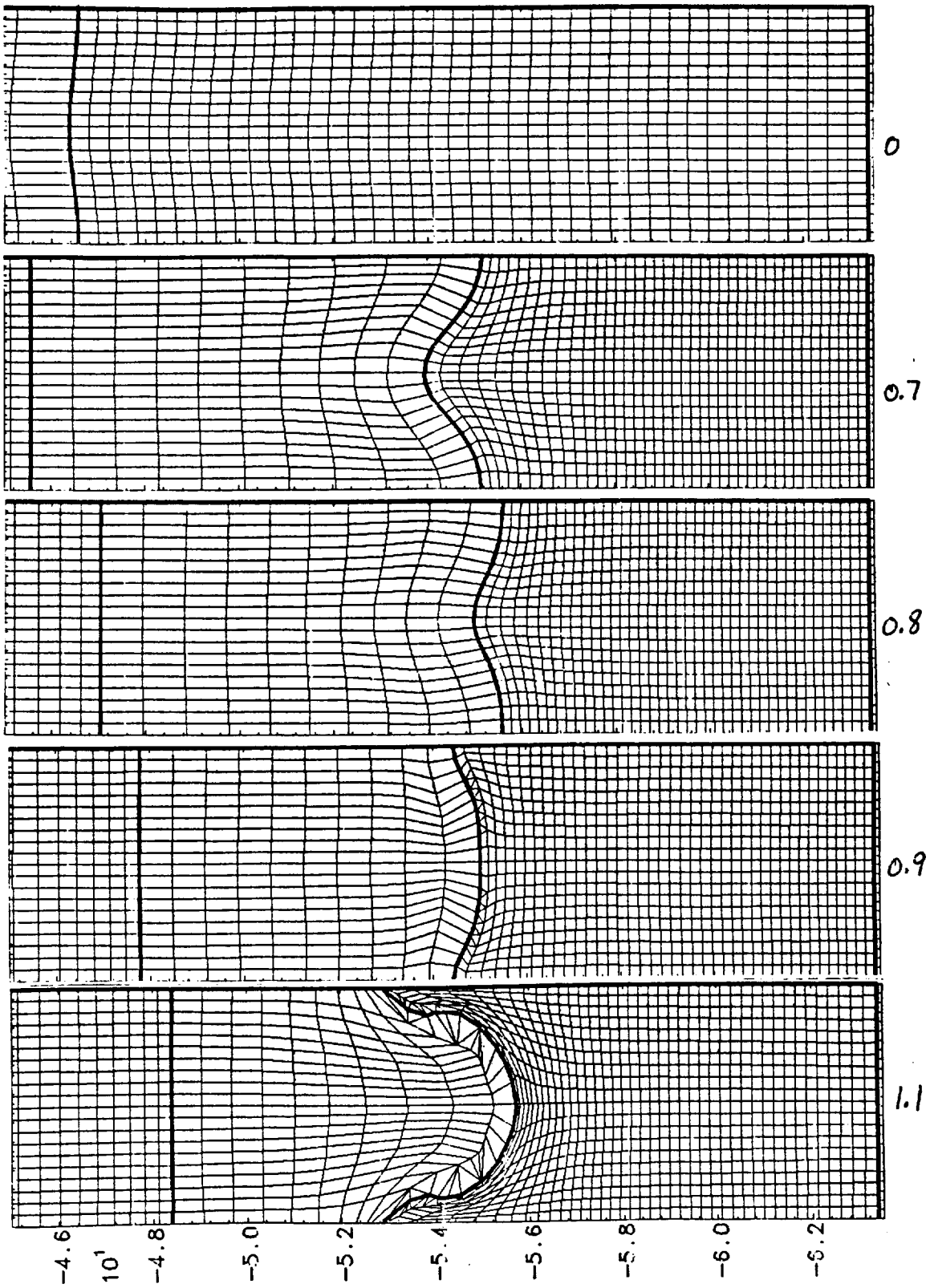


Figure 12

The initial shock travels from left to right and the interface separates light (left) from heavy (right) fluids. Only part of the total mesh is shown. The initial shock reflects off the right edge of the problem and reaches the interface about t = 0.7. At this time the reflected shock passes through the interface. Since this shock is from heavy to light, the perturbation goes through a phase change while continuing to grow.

Figure 13 shows the mesh around the perturbation at the final time. As the perturbation grows, resolution is lost along the interface. The CREAP option is being used to extend the calculation to later times.

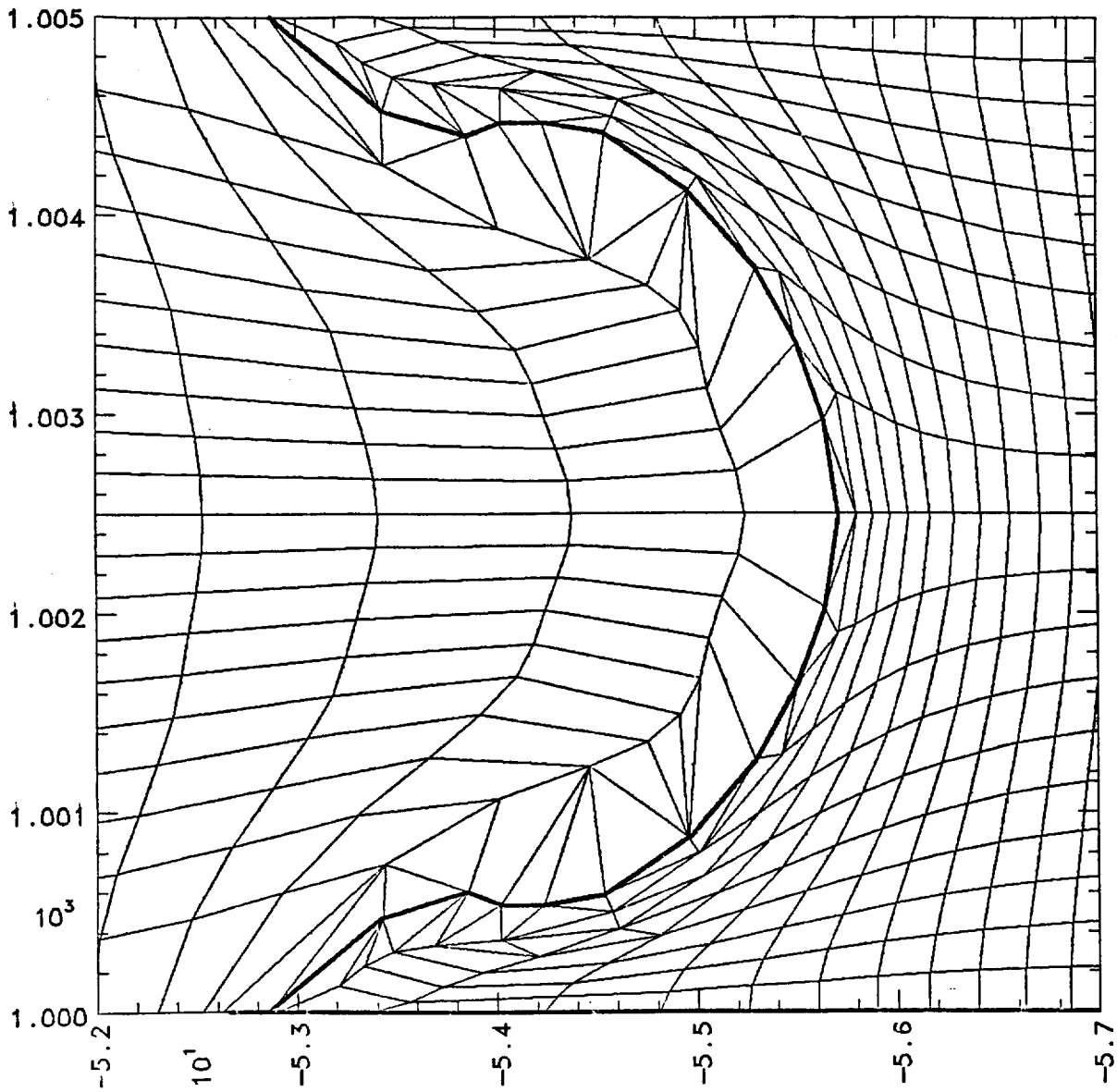


Figure 13

References

1. Crowley, W. P. "Flag: A Free-Lagrange Method for Numerically Similating Hydrodynamic Flows in Two Dimensions", Proceedings of the Second International Conference on Numberical Methods in Fluid Dynamics, Springer-Verlag, pp. 37-43, (1970).

2. von Neumann, J. and Richtmyer, R. D., "A Method for the Numerical Calculation of Hydrodynamic Shocks:, J. App. Phys., vol. 21, pp. 232-257 (1950).

3. White, J. W. "A New Form of Artificial Viscosity for Elastic Solids", J. Comp. Phys., vol. 16, pp. 119-126 (1974).

4. Noh, W. F., "CEL: A Time Dependent, Two-Space-Dimensional, Coupled Eulerian-Lagrange Code", Methods in Computational Physics, vol. 3, pp. 117-179 (1963).

5. Pomraning, J., Radiation Hydrodynamics, Pergamon Press, p. 232 (1973).

6. Maenchen, G. and Sack, S. "The Tensor Code", Methods in Computational Physics, vol. 3, pp. 181-210 (1963).

7. Margolin, L. G. and Nichols, B. D. "Momentum Control Volumes for Finite Difference Codes", Proceedings: Third International Conference on Numerical methods in Laminar and Turbulent Flow, 1983. Also, LANL manuscript LA-UR-83-524.

Acknowledgment

This project has profited over the years by the efforts of a number of people. I would especially like to thank T. Rudy, M. Uyemura, R. Williams, R. Cooper, D. Braddy, and M. Johnson for their invaluable aid and assistance.

This work was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.