JOURNAL OF COMPUTATIONAL PHYSICS 82, 29–63 (1989)

A General Topology Godunov Method*

JOHN K. DUKOWICZ, MICHAEL C. CLINE, AND FRANK L. ADDESSIO

Theoretical Division, Group T-3, Los Alamos National Laboratory, Los Alamos, New Mexico 87545

Received December 21, 1987; revised May 23, 1988

We describe a numerical technique for solving 2-dimensional compressible multimaterial problems using a general topology mesh. Multimaterial problems are characterized by the presence of interfaces whose shapes may become arbitrarily complex in the course of dynamic evolution. Computational methods based on more conventional fixed-connectivity quadrilateral meshes do not have adequate flexibility to follow convoluted interface shapes and frequently fail due to excessive mesh distortion. The present method is based on a mesh of arbitrary polygonal cells. Because this mesh is dual to a triangulation, its topology is unrestricted and it is able to accommodate arbitrary boundary shapes. Additionally, this mesh is able to quickly and smoothly change local mesh resolution, thus economizing on the number of mesh cells, and it is able to improve mesh isotropy because in a region of uniform mesh the cells tend to become regular hexagons. The underlying algorithms are based on those of the CAVEAT code. These consist of an explicit, finite-volume, cell-centered, arbitrary Lagrangian–Eulerian (ALE) technique, coupled with the Godunov method, which together are readily adaptable to a general topology mesh. Several special techniques have been developed for this extension to a more general mesh. They include an interface propagation scheme based on Huygens' construction, a "near-Lagrangian" mesh rezoning algorithm that minimizes advection while enhancing mesh regularity, an efficient global remapping algorithm that is capable of conservatively transferring quantities from one general mesh to another and various mesh restructering algorithms, such as mesh reconnection, smoothing, and point addition and deletion.

1. INTRODUCTION

Multimaterial problems, which inherently contain interfaces, are best suited to Lagrangian methods of computation. This is because Lagrangian methods resolve interfaces crisply, in contrast to Eulerian methods, which diffuse or smear them out. Unfortunately, Lagrangian methods in multidimensions suffer from "mesh tangling," an extreme form of mesh distortion which limits the time to which a purely Lagrangian computation may be taken. This difficulty may be alleviated by the use of the arbitrary Lagrangian–Eulerian (ALE) technique [2], which permits arbitrary mesh motion. In particular, by requiring only the interface to be Lagrangian (in fact, only its normal velocity needs to be Lagrangian), it is possible

0021-9991/89 $3.00

to retain the sharp definition of the interface, together with a robust, well-defined mesh in the interior. This is the principle of the CAVEAT code [1], and it has generally proven to be successful.

However, such a method still may experience difficulty and may even fail because of mesh difficulties. This is because most codes (including CAVEAT) utilize a fixed topology mesh, defined at the outset, which in general will not be able to adapt to the dynamically evolving interface shape, in spite of efforts at regularization. The most general solution to this difficulty, while preserving a Lagrangian interface, is to relax the constraints on mesh topology. This is the principal idea behind the CAVEAT-GT code, the subject of the present paper.

The CAVEAT-GT code is therefore a general topology extension of the 2-dimensional CAVEAT code [1]. This code solves the transient, multimaterial, compressible equations of fluid dynamics. CAVEAT-GT is currently restricted to solving the multimaterial compressible Euler equations, neglecting the effects of material strength, viscosity, and heat conduction. In most cases it uses the same algorithms, including the cell-centered Godunov method, most of which are easily adapted to a general topology mesh. In this paper, therefore, we will restrict ourselves to a description of those aspects, primarily associated with the general topology mesh, in which CAVEAT-GT differs from CAVEAT.

The CAVEAT-GT computational mesh consists of arbitrary polygonal cells, which are the control volumes of a finite-volume method. Closely associated with this mesh is a dual triangulation. It is the arbitrary topology of this triangulation which permits the great flexibility of the computational mesh. This flexibility allows the mesh to adapt itself to arbitrary interface contours and to resolve regions of even the highest curvature, if desired. It permits a rapid and smooth change in local mesh resolution, which economizes in the total number of cells required for a given degree of resolution. Another advantage is the enhancement of mesh isotropy because in regions of uniform mesh the computational cells tend to become regular hexagons, whose isotropy is higher than that of square or rectangular cells.

The desirable properties of such a general topology mesh are not without a price. In general, the data structure is more complicated and the algorithms are more complex. There is very little prior experience to draw from and so the development of CAVEAT-GT has been largely experimental. More work is required. Nevertheless, the experience has been largely successful and even at its current stage of development, as described herein, CAVEAT-GT is capable of solving nontrivial problems.

2. Mesh Geometry

The computational domain modeled by the CAVEAT-GT algorithm is divided into nonoverlapping, closed regions. Each region typically is associated with a specific material. It may be convenient, however, to divide a single material into a number of regions. This gives the ability to follow Lagrangian surfaces within a

material. Boundaries along which regions interact are referred to as interfaces. Because interfaces are associated with more than one region, mesh vertices along interfaces are doubly defined.

Associated with each region is an underlying triangulation. This triangulation is used to define the computational cells and it is useful for describing the mesh topology (connectivity) and the associated data structure. The computational cells are the control volumes used by the finite-volume formulation of the governing equations. Interior computational cells are defined by vertices that are the centroids of the associated triangles (Fig. 1). Therefore, there is a cell associated with each triangle vertex. It is from these nonoverlapping, closed, arbitrary polygonal cells that CAVEAT-GT derives its geometrical flexibility. Cells lying adjacent to boundaries also are defined by cell vertices interior to a region. Along the boundary, however, a point lying halfway between the cell points is used to define the extent of the boundary cell. Such points are referred to as boundary points (Fig. 1). Computational cells are not necessarily convex. Boundaries and interfaces are constructed with linear segments defined by the cell points lying on the boundaries. Boundary segments and points that lie on the interfaces separating regions are doubly defined.

In general, computational quantities are associated with one of three locations on a computational cell. Extensive properties (i.e., mass, momentum, and energy), as well as the intense properties (i.e., density, velocity, pressure) derived from them, are associated with the cell centroids. The computational procedure also requires intensive quantities on cell sides. Information on cell sides is necessary to obtain left and right states for the Riemann problem required by the Godunov method and for remapping. Finally, coordinate positions are associated with cell vertices. The subscripts k, m, and n are used to denote cell centered, side, and vertex (or triangle) quantities, respectively.

Certain special boundary cell-points are treated differently. They are called "fixed points" and they include triple points, special symmetry points, and user-specified boundary cell-points. Boundary cell-points located where three regions adjoin are referred to as triple points. The intersection of a line of symmetry and a region boundary or two lines of symmetry are fixed symmetry points. Finally, kinks or corners in the boundary contour which are not to be smoothed may be defined as

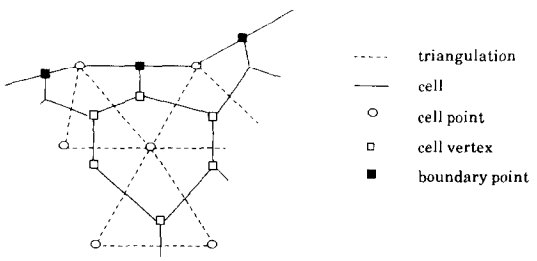


FIG. 1. Mesh geometry.

"fixed points" by the user. The locations of "fixed points" are determined prior to and remain fixed during the interface construction. That is, they act as Dirichlet boundary conditions for the interface construction algorithms.

3. METHODOLOGY

CAVEAT-GT is a 2-dimensional computer program defined for either Cartesian (planar) or cylindrical geometry. Time differencing is explicit; consequently, a stability limit is imposed on the time-step size (Δt), based on a local sound speed and a minimum characteristic cell dimension. Additional time-step restrictions based on accuracy considerations are also used to limit volume changes within a cycle or to limit the interface motion.

CAVEAT-GT uses the arbitrary Lagrangian–Eulerian (ALE) method: a 3-phase algorithm to advance the material state one full cycle (Fig. 2). During the Lagrangian phase, the material state is advanced by solving the conservation equations applied to volumes following the material motion. Boundary and interface positions also are updated during this phase. Following the Lagrangian phase, a new mesh is generated. This step is the rezoning phase. Finally, in the remapping phase, the variables calculated in the Lagrangian step are transferred from the

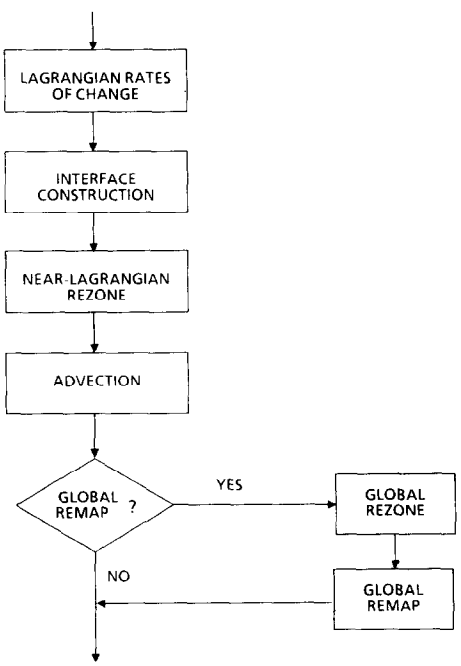


FIG. 2. The CAVEAT-GT computational cycle.

Lagrangian mesh to the new mesh. CAVEAT-GT contains two rezone/remap algorithms, namely, a "near-Lagrangian" algorithm and a global algorithm. Because the global rezone/remap algorithm is more costly, it is used infrequently and only when necessary, such as when the mesh topology changes. The "near-Lagrangian" algorithm is used on every cycle. In fact, the global algorithm uses the "near-Lagrangian" rezone as its starting point.

3.1. *Lagrangian Phase*

During the Lagrangian phase, the rates of change of volume, mass, momentum, and energy are computed assuming that the computational volumes are following the material motion. That is, we evaluate the right-hand sides of the Lagrangian conservation equations of volume, mass, momentum, and total energy in control volume form, which are

$$\begin{aligned} \frac{d}{dt}\int_{V_L} dV &= \oint_{S_L} \mathbf{u}^* \cdot \mathbf{n}\, dS, \\ \frac{d}{dt}\int_{V_L} \rho\, dV &= 0, \\ \frac{d}{dt}\int_{V_L} \rho\mathbf{u}\, dV &= -\oint_{S_L} p^*\mathbf{n}\, dS + \oint_{S_L} \boldsymbol{\tau}^* \cdot \mathbf{n}\, dS, \end{aligned} \tag{1}$$

and

$$\frac{d}{dt}\int_{V_L} \rho E\, dV = -\oint_{S_L} p^*\mathbf{u}^* \cdot \mathbf{n}\, dS + \oint_{S_L} \mathbf{u}^* \cdot \boldsymbol{\tau}^* \cdot \mathbf{n}\, dS - \oint_{S_L} \mathbf{q}^* \cdot \mathbf{n}\, dS.$$

An equation of state and constitutive equations for the stress tensor and energy flux are required to close this system of equations. The notation $V_L(t)$ denotes a Lagrangian control volume (i.e., a computational cell), with surface $S_L(t)$, moving at the local material velocity. The unit normal vector directed outward from the surface is $\mathbf{n}$. The operator d/dt is the Lagrangian (material) time derivative. The quantities ρ, e, p, $\mathbf{u}$, E, τ, and $\mathbf{q}$ are the density, specific internal energy, pressure, material velocity, specific total energy ($E = e + \frac{1}{2}\mathbf{u} \cdot \mathbf{u}$), deviatoric stress tensor, and heat flux, respectively. An asterisk is used to denote a quantity located at a cell face. Currently, CAVEAT-GT does not implement the terms involving the stress tensor and heat flux.

CAVEAT-GT is completely cell-centered; variables that specify the material state (including momentum) are assumed located at the centroids of the computational cells. This makes it feasible to compute material slip without the need to include a logical slideline. The accuracy of the method is dependent on the assumed spatial variation of a representative intensive quantity $\phi(\mathbf{x})$ about the cell centroid $\bar{\mathbf{x}}_k$, i.e.,

$$\phi(\mathbf{x}) = \phi(\bar{\mathbf{x}}_k) + \nabla_k \phi \cdot (\mathbf{x} - \bar{\mathbf{x}}_k) + O(\Delta x^2). \tag{2}$$

The computational procedure is denoted to be spatially "first-order" if all quantities are assumed constant within a computational cell, that is, if $\phi(\mathbf{x}) = \phi(\bar{\mathbf{x}}_k)$. The method is denoted to be spatially "second-order" if the gradient $(\nabla_k \phi)$ exists, that is, if a linear variation for the variable is assumed within a cell. Note that this does not necessarily imply the corresponding order of numerical accuracy of the overall algorithm.

Three options are available for computing the cell-centered gradients $(\nabla_k \phi)$. A temporary trial gradient located at the cell vertices $(\mathbf{x}_n)$ is first computed. Using the divergence theorem, this is taken to be the area-averaged gradient:

$$\langle \nabla_n \phi \rangle = \frac{1}{A_n} \int_{A_n} \nabla \phi \, dA,$$

$$= \frac{1}{A_n} \oint_{C_n} \phi \mathbf{n} \, dA, \tag{3}$$

where A_n is the area of the triangle connecting the centroids of the three computational cells surrounding the vertex for an interior cell (Fig. 1). For cells lying on region boundaries A_n is the area of the quadrilateral whose vertices are the cell centroids and the boundary points (Fig. 1). The symbols C_n and $\mathbf{n}$ represent the contour and unit outward normal to A_n, respectively. Values of ϕ are assumed located at the cell centroids $(\phi(\bar{\mathbf{x}}_k))$ in the mesh interior. At the boundary points, values of ϕ are deduced from the appropriate boundary conditions. A piecewise linear variation of ϕ along the contour C_n is assumed. This is consistent with the assumption that $\phi(\mathbf{x})$ is a linear function. An area weighted average of the vertex trial gradients associated with a computational cell then gives a trial gradient at the cell centroid $(\bar{\mathbf{x}}_k)$:

$$\langle \nabla_k \phi \rangle = \frac{\sum_n A_n \langle \nabla_n \phi \rangle}{\sum_n A_n}. \tag{4}$$

This gradient may be used as one option; however, this destroys monotonicity and severe overshoots or undershoots in the vicinity of steep gradients may be produced. To preserve monotonicity, the gradients must be limited;

$$\nabla_k \phi = \alpha_k \langle \nabla_k \phi \rangle, \tag{5}$$

where α_k is a limiting coefficient $(0 \leqslant \alpha_k \leqslant 1)$. The use of gradient limiting reduces the order of accuracy of the calculation and adds numerical dissipation to the algorithm. Two limiting algorithms are used as the remaining two options. A multidimensional extension of van Leer's [1, 3, 11] 1-dimensional limiter permits the steepest possible gradients but is not always monotone. A less radical but somewhat more diffusive monotone limiter is also available [1, 11].

Evaluation of the right-hand sides of Eq. (1) requires the pressure (p^*) and the velocity normal to the control volume surface $(w^* = \mathbf{u}^* \cdot \mathbf{n})$ at the midpoint of each cell face. In general, an extrapolation using Eq. (2) produces a discontinuity at the

cell face. These discontinuities are resolved by solving an approximate Riemann problem [4] at each cell face. This, in effect, is an extension of the original 1-dimensional Godunov method.

Solutions of the Riemann problem provide normal material velocities, $\mathbf{w}^* = (\mathbf{u}^* \cdot \mathbf{n})\mathbf{n}$, on the cell sides. These velocities are used not only for the solution of the conservation equations, but also to propagate interfaces and boundaries by the use of Huygens' construction. For each interior cell side there is one such velocity. However, there are two Riemann velocities associated with the straight line segments located along the region boundaries (Fig. 3). These velocities are positioned midway between the cell point and the boundary point at the locations $\boldsymbol{\alpha}'$ and $\boldsymbol{\beta}'$. This can be used to define a linear distribution of the normal velocity at each boundary segment. Assuming that the magnitude of the normal velocities remains constant, Huygens' construction predicts that the boundary segment will remain a straight line during a time step. The two boundary-segment velocities are used to construct "wavefronts" with radius $w^* \Delta t$, whose envelope is the new segment location according to Huygens' construction. The new boundary-segment position is specified by the points $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ that lie on the tangent to the two circular "wavefronts" emanating from the points $\boldsymbol{\alpha}'$ and $\boldsymbol{\beta}'$. Restrictions are placed on the time-step size (Δt) to prevent the new segment from rotating more that 90°. That is, the larger circular front on a segment is prevented from overtaking the smaller one in an unphysical fashion.

Having found the new locations of the boundary segments, the new boundary-cell points might be located at the intersections of the segments. This is, in fact, what is done for the location of the "fixed points." There are two difficulties with this procedure, however. First, in the case of a triple point, there are potentially three different intersection points. This ambiguity is resolved by linearly combining the three positions using a density weighting. Second, the intersections are not defined when the segments are parallel or collinear. The problem may be easily regularized if posed in a variational form. A variational formulation for the intersection of two lines follows from the realization that the intersection point is simultaneously the shortest distance from both lines. Therefore, the point of intersection ($\mathbf{x}_k$) is determined by minimizing the functional (Fig. 4),

$$I_{Ik} = d^2_{m,k} + d^2_{m+1,k}, \tag{6}$$

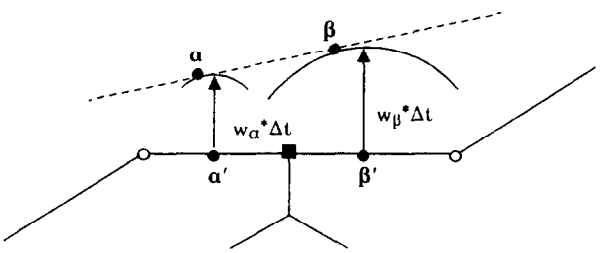


FIG. 3. Interface propagation by Huygens' construction.

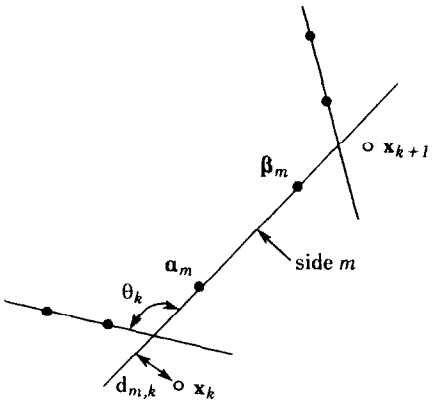


FIG. 4. Boundary-cell point solution.

where $d^2_{m,k}$ is the square of the distance from the point $\mathbf{x}_k$ to the line segment m given by

$$d^2_{m,k} = |\mathbf{e}_z \cdot (\mathbf{x}_k - \boldsymbol{\alpha}_m) \times \boldsymbol{\delta}_m|^2 / |\boldsymbol{\delta}_m|^2, \tag{7}$$

where $\mathbf{e}_z$ is the unit vector normal to the computational plane, and $\boldsymbol{\delta}_m = \boldsymbol{\beta}_m - \boldsymbol{\alpha}_m$ is the distance between the two points obtained from the Huygens' construction. All the intersection points may be found by taking the variation of the sum of these terms over all boundary segments with respect to the free variables. The free variables are the coordinates of all boundary points other than the "fixed points." This variational problem also is singular for parallel or collinear boundary segments. However, the problem is now regularized easily by including the additional term

$$I_{Em} = |\mathbf{x}_{k+1} - \mathbf{x}_k|^2 \tag{8}$$

in the functional. The resulting functional is

$$I(x_k) = \sum_k \omega_k I_{Ik} + \varepsilon \sum_m \omega_m I_{Em}. \tag{9}$$

Again, the variation of this functional provides a system of equations for the boundary-point positions. Provided the points are ordered appropriately, the system of equations is in tridiagonal form and is solved easily.

Equation (9) contains two weight functions, which may be used to control point distribution, and a scaling factor. The weight function ω_k is

$$\omega_k = \sin^2 \theta_k, \tag{10}$$

where θ_k is the angle between the two boundary segments (Fig. 4). This choice for ω_k weights the functional I_{Ik} less as the boundary segments approach parallelism,

and this has been found useful to control the singularity at a single boundary point when the local curvature undergoes a change of sign. The second weight function is

$$\omega_m = \rho'_m/(\rho'_m\, d_m)_0, \tag{11}$$

where $\rho'_m = \frac{1}{2}(\rho_k + \rho_{k+1})$ is the average density along the segment and $d_m = |\mathbf{x}_k - \mathbf{x}_{k-1}|$. The denominator of Eq. (11) contains values at the time of the last global remap. With the weight defined by Eq. (11), the regularizing term attempts to equidistribute a "1-dimensional" mass distribution along parallel or collinear boundaries where this term dominates. Therefore, boundary-cell points are spaced closer in regions of high density and so the points move in a "near-Lagrangian" fashion.

3.2. *Rezone Phase*

The rezoning phase of the CAVEAT-GT algorithm specifies a new mesh. The positions of the boundary-cell points are determined first and are then used as boundary conditions for the algorithm that determines the positions of the interior vertices.

Two methods of rezoning are used. The "near-Lagrangian" method constructs a mesh whose cells have nearly the same mass as the cells of the Lagrangian mesh, but with reduced distortion. Because the mesh is nearly Lagrangian this method minimizes advection errors, but it is usable only when the mesh topology does not change. The global rezone scheme is invoked when new mesh points must be added or deleted, or when the interior cells have become sufficiently distorted. The global rezone constructs an entirely new mesh based on smoothness criteria. Typically, the mesh topology will change. The global rezone is more complex and costly than the "near-Lagrangian" rezone, but it is invoked less frequently.

3.2.1. *Boundary rezone.* Advancement of the interfaces and boundaries that enclose each region is accomplished by the interface construction technique (Section 3.1). This construction uses the velocities normal to the boundary segments (i.e., the velocities obtained from the Riemann problem) to position the new boundary. However, the location of boundary-cell points tangentially along the boundary contour is not specified and is arbitrary, provided the boundary is sufficiently well resolved. The original interface construction may not distribute boundary points in a satisfactory manner. This job is performed by the boundary rezone algorithm.

Again, there are two boundary rezone algorithms. Provided the boundary curvature is resolved sufficiently with the existing points, the "near-Lagrangian" placement locates boundary points to preserve the original mass distribution along the boundary segments. This ensures that the associated advection across cell sides that intersect the boundary is minimized. The method used for this purpose is analogous to the formulation developed to regularize the interface construction technique (Eqs. (8) and (11)). Because the boundary contour is known, the

problem is 1-dimensional in the arc length (s). Therefore, the boundary-cell points are located along the contour at positions given by the arc lengths s_k which minimize the variational functional

$$I = \sum_m \omega_m \Delta s_m^2, \tag{12}$$

where $\Delta s_m = s_k - s_{k-1}$. The weighting function is defined by Eq. (11). This boundary rezone typically is followed by a "near-Lagrangian" interior rezone.

In case the boundary is under-resolved or over-resolved with the existing number of points, the general topology mesh allows one to add or delete points without unduly influencing the rest of the mesh. The rezoning procedure which both determines the required number of points and also locates the points makes use of a point distribution function N satisfying the ordinary differential equation

$$\frac{dN}{ds} = f(s, \kappa, \nabla\phi, \ldots), \tag{13}$$

where the right-hand side is a specified point-distribution density function, as described below. Equation (13) is integrated along boundary contours between "fixed points." The resulting values for $N(s)$ are scaled to ensure that the final value for $N(s = L)$ is an integer. That is, the positions of the "fixed points" are not altered. The boundary points then are placed along the boundary contour at positions where $N(s)$ has integer values. Solutions for $N(s)$ are obtained every time step. The solution is tested to determine if the existing boundary-point distribution sufficiently resolves the boundary contour. That is, if boundary-point addition or deletion or gross vertex migration is unnecessary, the boundary-point positions resulting from the "near-Lagrangian" description are accepted. However, if the "near-Lagrangian" positions are not adequate to accurately resolve the boundary contour, then the final positions of the boundary points are specified by Eq. (13), and this must be followed by a global rezone of the interior mesh.

The point-distribution density function is defined as

$$f = f_0 \left[1 + \alpha_k (\Delta s \kappa)_k + \alpha_a \left| \frac{\Delta s \nabla\phi}{\phi} \right|_k \right] \Big/ \Delta s_{\max}, \tag{14}$$

where f_0 is a normalizing coefficient, the variable $\Delta s_{\max}$ is a user supplied maximum spacing allowed for each region, and α_k and α_a are user supplied weight constants. The point-distribution density function is chosen to equally distribute boundary points along the interfaces in the absence of any distinguishing features. Otherwise, boundary points are forced to migrate into regions with large values of the boundary curvature (κ) or the gradient ($\nabla\phi$) of a prescribed variable, such as pressure. The boundary curvature (κ) is calculated by passing a circle through the kth boundary point and its two neighbors. At multiply defined boundary points along region interfaces, only the maximum value of the point-distribution density

function is retained. Further, before being used in Eq. (13) the distribution function (f) is smoothed around the entire region boundary using a convolution with a Gaussian of prescribed width.

3.2.2. *Interior rezone.* The interior rezone algorithms construct a mesh in the interior of each region using the boundary-point positions, obtained from the boundary rezone computation, as boundary conditions. Two interior rezoning algorithms are used, a "near-Lagrangian" and a global algorithm, corresponding to the two boundary rezoning algorithms. Both of the interior-rezone algorithms are based on the dual triangulation, rather than on the computational mesh.

Unfortunately, it is not feasible to obtain sufficiently accurate fluid velocities at cell vertices in the interior of the mesh, corresponding to the Riemann normal velocities at cell faces. It is therefore not possible to define a purely Lagrangian mesh motion and thus avoid remapping or advection errors. Instead, we attempt to define mesh velocities that are close to Lagrangian velocities, while at the same time attempting to minimize mesh distortions. The "near-Lagrangian" rezone algorithm thus attempts to preserve Lagrangian cell volumes and also to maintain a smooth mesh.

We make use of the vector identity

$$\nabla^2 \mathbf{u} = \nabla D - \nabla \times \boldsymbol{\omega}, \tag{14a}$$

where $D = \nabla \cdot \mathbf{u} = 1/v\ dv/dt$ is the divergence, $\boldsymbol{\omega} = \nabla \times \mathbf{u}$ is the vorticity, and v is the specific Lagrangian volume. Realizing that mesh distortion is associated primarily with vorticity, we drop the last term on the right-hand side and define mesh velocities ($\mathbf{u}_m$) by the equation

$$\nabla^2 \mathbf{u}_m \equiv \nabla D. \tag{15}$$

The mesh velocity field $\mathbf{u}_m$ obtained from this equation has a divergence equal to the divergence of the fluid velocity, and its vorticity satisfies a Laplacian equation, making it a smooth function in the region interior. Thus, this formulation preserves Lagrangian volumes and smooths out mesh distortions due to vorticity.

Equation (15) may be expressed in the form

$$\nabla^2\, \delta \mathbf{x}_k = \nabla(\delta v_k / v_k). \tag{16}$$

This is a linear Poisson equation in which $\delta \mathbf{x}_k$ is the change of the position of the cell-center (i.e., triangle vertex) from its position at the start of the time step. The displacement $\delta \mathbf{x}_k$ is known on the boundary; hence, we have Dirichlet boundary conditions. The discretization on the triangulation is accomplished using a finite element formulation. That is, the functional

$$I = \int_\Omega [\|\nabla\, \delta \mathbf{x}_k\|^2 + 2\delta \mathbf{x}_k \cdot \nabla(\delta v_k / v_k)]\, dA \tag{17}$$

is minimized with respect to $\delta \mathbf{x}_k$. Linear elements are used. Two uncoupled matrix equations are obtained for δx_k and δy_k, respectively. The matrices for the two systems of equations are identical, and they are symmetric and positive-definite. We find that a diagonally scaled conjugate gradient method is an efficient way to solve these equations. Once the new triangle vertex positions have been determined, the cell vertices are obtained using linear interpolation.

In spite of its tendency to smooth the mesh, the "near-Lagrangian" rezone still may produce an unacceptable mesh (typically, only after a number of time steps). Further, because the "near-Lagrangian" rezone assumes no change in mesh topology it cannot be used when mesh topology is changed by adding or deleting points. In these circumstances a global-rezone method is employed to redefine and produce a smooth mesh. No attempt is made to retain the existing mesh, except for the boundary or interface shape as defined by the boundary rezone. The object is to produce a smooth mesh by all means available, including mesh topology changes. A global rezone is invoked if solutions to Eq. (17) produce triangles with negative areas, if an interior triangle contains an angle less than 10°, when a decision is made to add or delete mesh points, or when there is gross boundary point migration.

In a manner similar to established rezoning techniques [5], a variational formulation using a composite functional defined so as to produce desirable mesh characteristics is used. The functional incorporated into the CAVEAT-GT method is

$$I = I_\theta + \alpha I_\delta, \tag{18}$$

where I_θ and I_δ are separate measures of mesh distortion. The relative weight (α) is used primarily as a scaling coefficient.

The functional I_θ is defined as

$$I_\theta = \sum_n (d_{12}^2 + d_{23}^2 + d_{31}^2)_n / A_n. \tag{19}$$

The sum is over triangles. The area of the triangle is A_n and d_{ij} is the length of the side of the triangle. It may be shown that each term in Eq. (19) is equal to the sum of the cotangents of the interior angles of a triangle, and so I_θ is a measure of the departure of the triangle angles from 60° and therefore promotes the formation of equilateral triangles. However, since the functional is not sensitive to triangle areas it is found that triangle areas do not vary smoothly over the mesh. In an attempt to alleviate this difficulty, the additional functional I_δ is introduced,

$$I_\delta = \sum_m d_m^2, \tag{20}$$

where the sum is taken over all triangle sides and d_m is the length of each side. The functional I_δ is minimized when the lengths of all triangle sides are equal, and therefore it provides the necessary sensitivity to triangle size.

The complete functional (Eq. (18)) is minimized with respect to $\mathbf{x}_k$, that is, with respect to all triangle vertex coordinates except those on the boundaries. The mesh obtained from the "near-Lagrangian" interior rezone is used as the initial estimate for the global rezone. The resulting variational problem provides a coupled system of nonlinear equations for the coordinates x_k and y_k in each region. An iterative scheme is used to solve these equations during which changes in the connectivity of the mesh take place. That is, as the cell points are moved during the iteration, the angles not associated with the diagonal of every quadrilateral formed by two neighboring triangles are checked to ensure their sum is less than 180° (cf., angles θ_3 and θ_4 in Fig. 5). If the sum is greater than 180°, then the diagonal is switched. This reconnection tends to produce a Delaunay triangulation [6], the dual of the Voronoi mesh [7]. The change of topology allows the mesh extra degrees of freedom in order to relax to a smoother configuration. Since the method is nonlinear, mesh changes must be limited and, further, equation coefficients must be periodically recomputed, particularly if the topology changes.

3.3. *Remapping Phase*

Although the motion in the direction normal to the region boundaries is Lagrangian, the mesh motion tangent to the boundaries and in the interior differs from the material motion. Consequently, it is necessary to remap (or transfer) the results of the Lagrangian phase onto the mesh produced by the rezoning algorithms. The quantities transferred are the fundamental conserved quantities such as mass, momentum, and total energy. Therefore, any method used should preserve conservation. If the relative displacement of the Lagrangian and the rezoned meshes is sufficiently small (so as not to violate the stability of the numerical technique) then the remapping of the variables may be expressed in terms of fluxes across cell faces. This process is referred to as advection [11]. Advection, therefore, is appropriate when remapping to the mesh generated by the "near-Lagrangian" rezoning technique. However, a different remapping method must be used to transfer variables to the mesh created by the global rezoning scheme. Both remapping methods introduce a diffusion error. This error is reduced by increasing the order

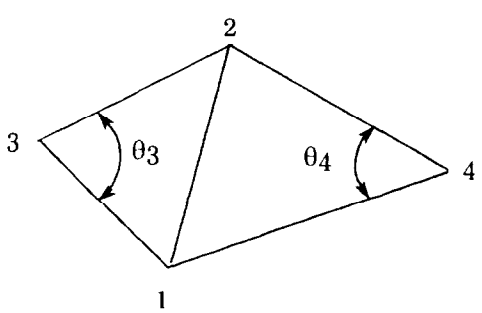


FIG. 5. Mesh restructuring.

of accuracy of the approach or by reducing the relative difference between the material and mesh velocities.

The expression for the rate of change of the integral of a representative intensive quantity $\rho\phi$, where ϕ may represent unity, velocity, or specific total energy, as a result of advection is

$$\frac{d}{dt}\int_V \rho\phi \, dV = -\oint_S \rho\phi(\mathbf{u}-\mathbf{u}_m)\cdot\mathbf{n}\, dS, \tag{21}$$

where $\mathbf{u}$ and $\mathbf{u}_m$ are the material and mesh velocities, respectively. The integrals refer to a computational cell with volume V whose surface and unit outward normal are S and $\mathbf{n}$, respectively. This equation is discretized in a straightforward fashion comparable to the method used in [1].

The global remap contained in CAVEAT-GT is an extension of previously developed methods for quadrilaterals [8–11] to the geometry of a general topology mesh. The object is to obtain a conservative transfer of conserved variables between two arbitrary meshes. There is no restriction on either the mesh topology or the time-step size. Because the variables are remapped from the "near-Lagrangian" positions to the new mesh, a global rezone/remap always is preceded by a "near-Lagrangian" rezone and advection.

In this type of remapping the cell-valued extensive quantity Φ_k^* on the new mesh, such as cell mass, momentum, or total energy, is obtained from the integral

$$\Phi_k^* = \int_{V_k^*} \phi(\mathbf{x})\, dV, \tag{22}$$

where $\phi(\mathbf{x})$ is the known distribution of the corresponding conserved quantity (i.e., ρ, $\rho\mathbf{u}$, or ρE) on the old mesh, and V_k^* is the volume of the kth cell of the new mesh. The problem, then, is to compute volume integrals in the presence of arbitrary overlapping of cells of the two meshes. The solution is facilitated by converting Eq. (22) into surface integrals using the divergence theorem

$$\Phi_k^* = \oint_{S_k^*} \mathbf{F}\cdot\mathbf{n}\, dS, \tag{23}$$

where we have introduced a vector flux function

$$\nabla\cdot\mathbf{F} = \phi(\mathbf{x}). \tag{24}$$

The quantity $\phi(\mathbf{x})$ is assumed to have at most a linear distribution within each cell of the old mesh (Eq. (2)), and the flux function is found locally in each cell. The integral of Eq. (23) is evaluated by tracing over the cell sides of the new mesh. However, the component of the flux function directed normally to the cell sides ($F_n = \mathbf{F}\cdot\mathbf{n}$) is, in general, not continuous across cell edges. Consequently, the surface integral must include the cell sides of the old mesh in order to subtract out the contributions resulting from this discontinuity. A major factor contributing to the

effectiveness of this scheme for quadrilateral meshes is the ability to trace the entire mesh uniquely by continuous paths along cell edges. This eliminates the need to search for the location of the points of one mesh within the cells of the other. It is rather remarkable (because it is not obvious at first sight) that it is possible to identify corresponding unique paths in an arbitrary triangulation, such as used in CAVEAT-GT, thus retaining essentially all the effectiveness of the method in a quadrilateral mesh.

4. Example Problems

Two relatively simple test problems are presented to illustrate some of the features and capabilities of the general topology formulation of CAVEAT-GT. The first problem, a blast wave, illustrates the general isotropy of the mesh and the properties of the global remapping and rezoning algorithms. The second problem, an impact problem, although not really a multimaterial problem, contains an interface (a free surface) and therefore illustrates the methods of interface propagation and boundary rezoning. Both problems are posed in dimensionless form in planar geometry.

4.1. *Blast Wave*

The blast wave problem is defined on the initial mesh shown in Fig. 6. The problem domain is a 2×2 square, occupied by a gamma-law gas ($\gamma = \frac{5}{3}$), with an "obstacle" located in the lower left corner. The mesh is relatively coarse, containing

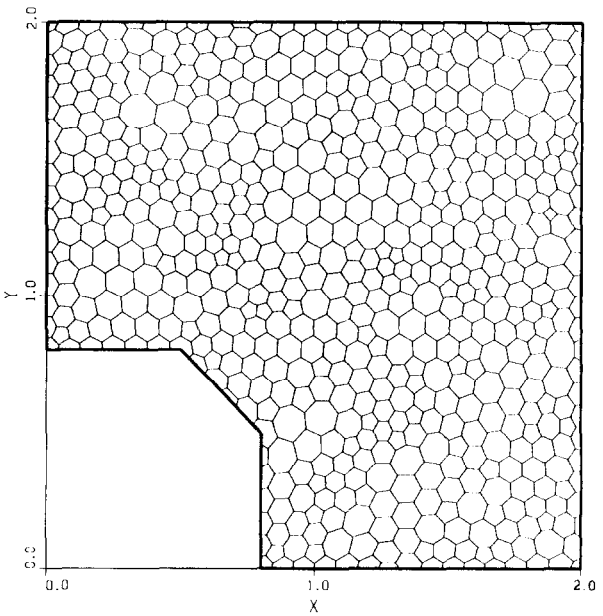
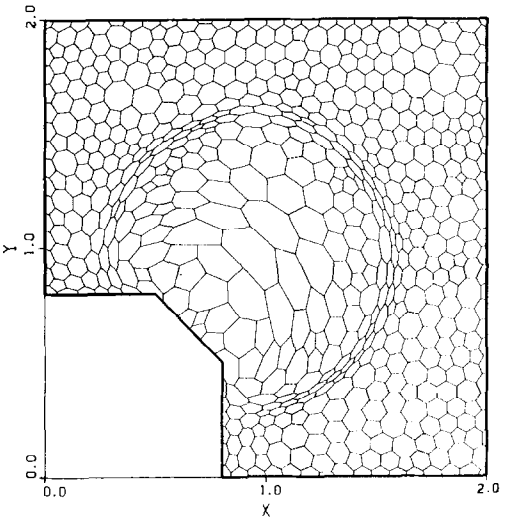
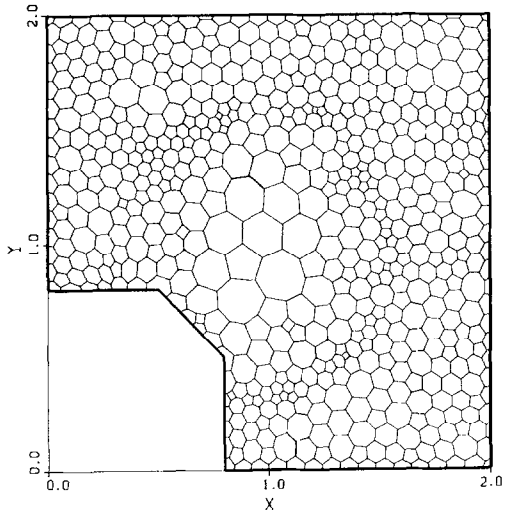


FIG. 6. Initial mesh geometry for the blast wave problem.

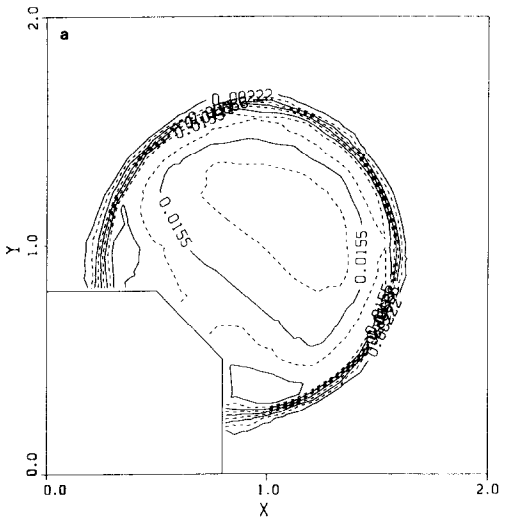


(a)

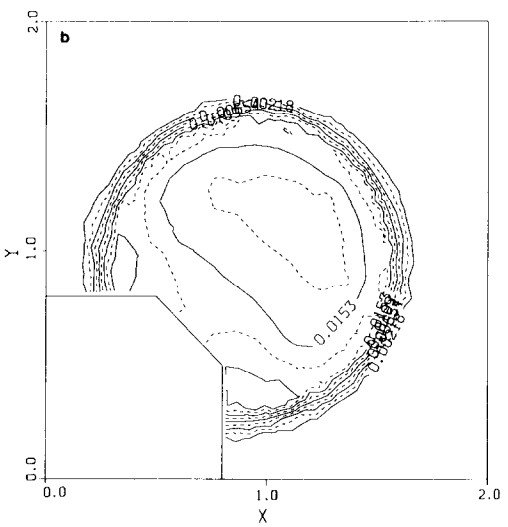


(b)

FIG. 7. Mesh geometry preceding and following a global rezone: (a) cycle 170; (b) cycle 171.



(a)



(b)

FIG. 8. Pressure contours preceding and following a global remap for a "first-order" calculation: (a) cycle 170; (b) cycle 171.

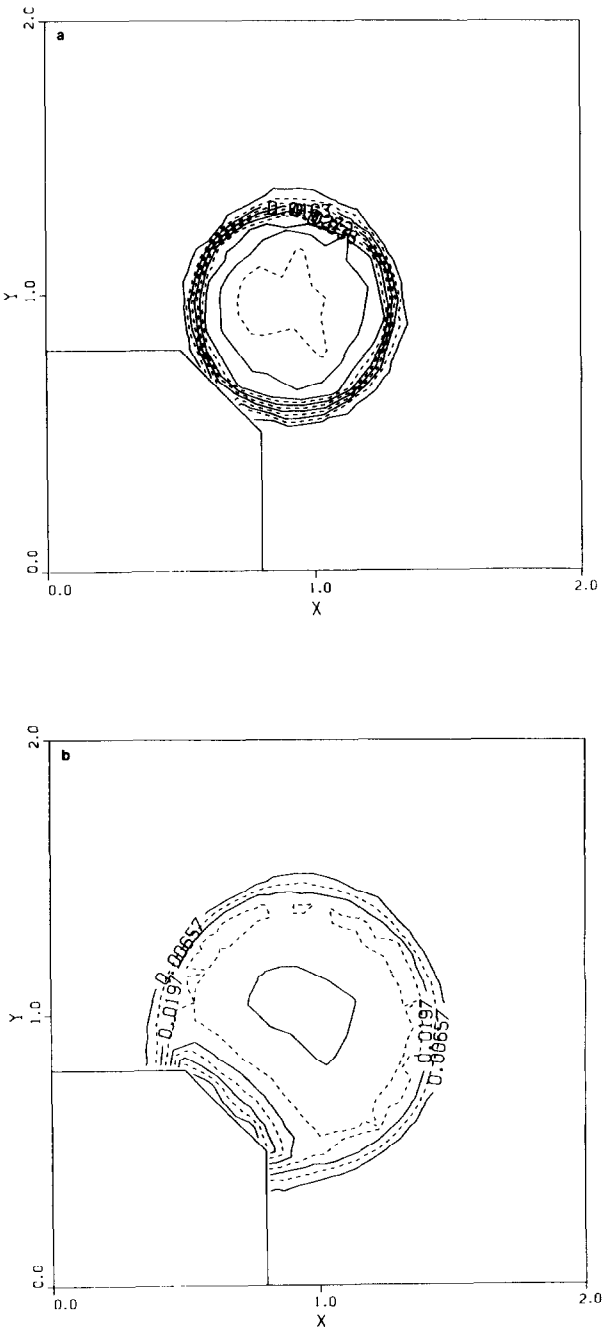
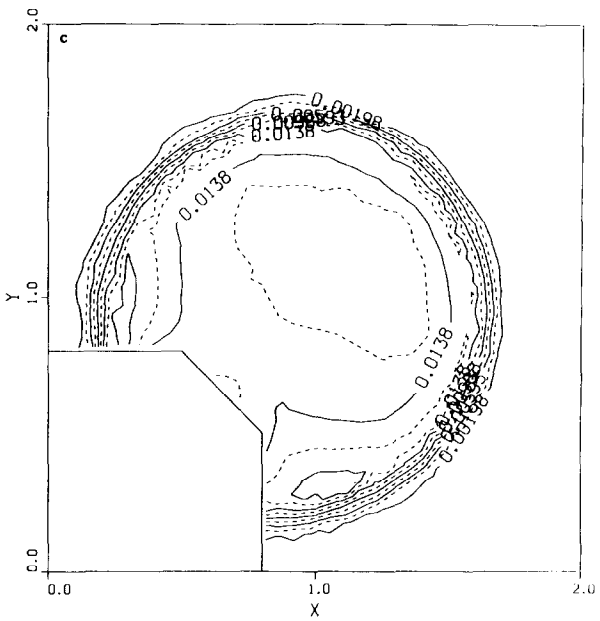


FIG. 9. Pressure contours for the blast wave problem using a "first-order" calculation: (a) $t = 0.5$; (b) $t = 1.0$; (c) $t = 2.0$; (d) $t = 2.5$.



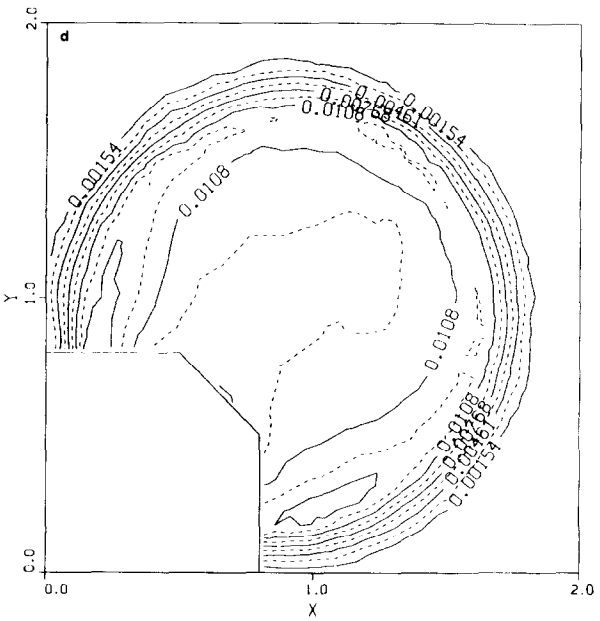


FIG. 9—*Continued*

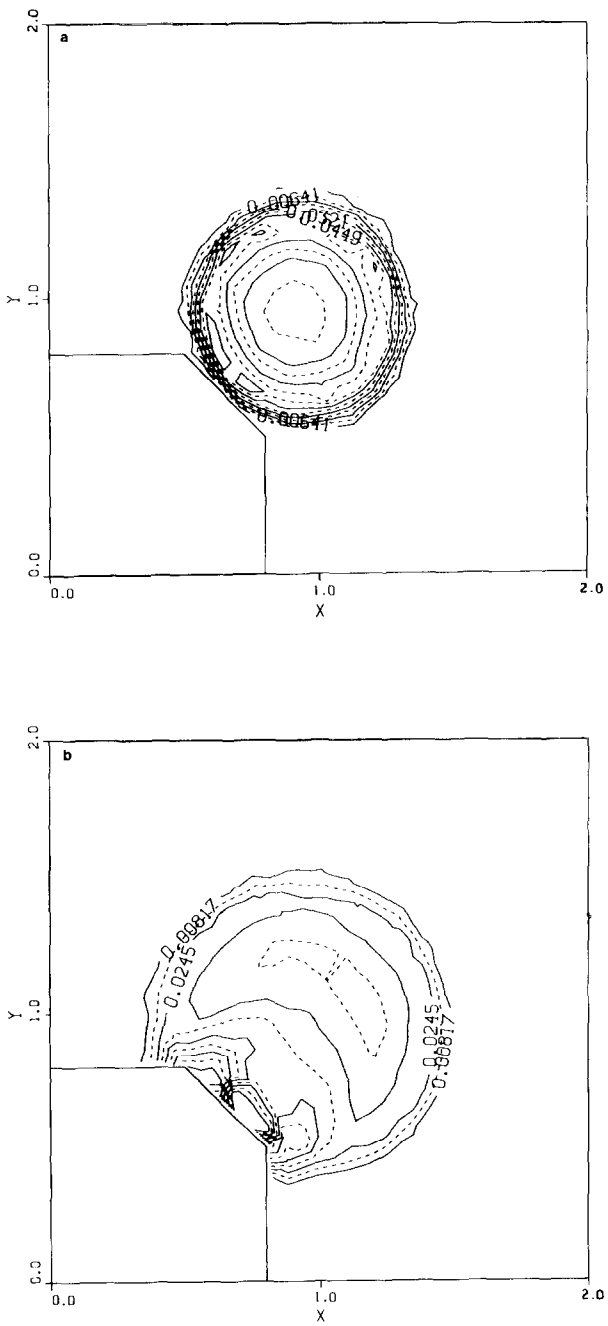
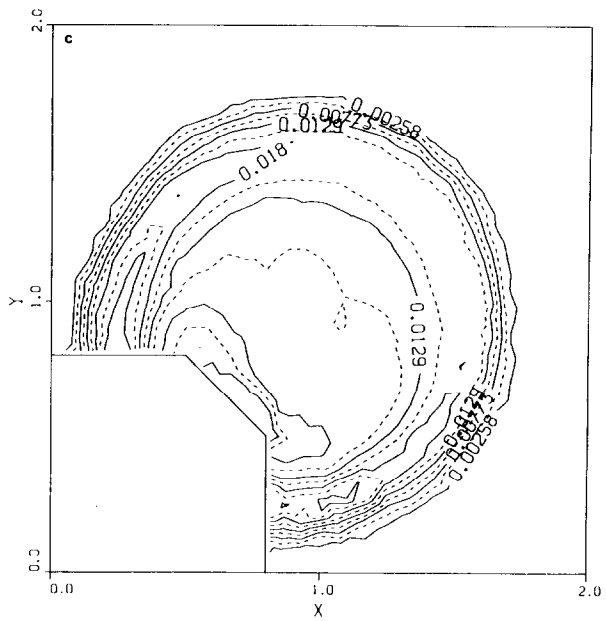


FIG. 10. Pressure contours for the blast wave problem using a "second-order," van Leer limited calculation: (a) $t = 0.5$; (b) $t = 1.0$; (c) $t = 2.0$; (d) $t = 2.5$.



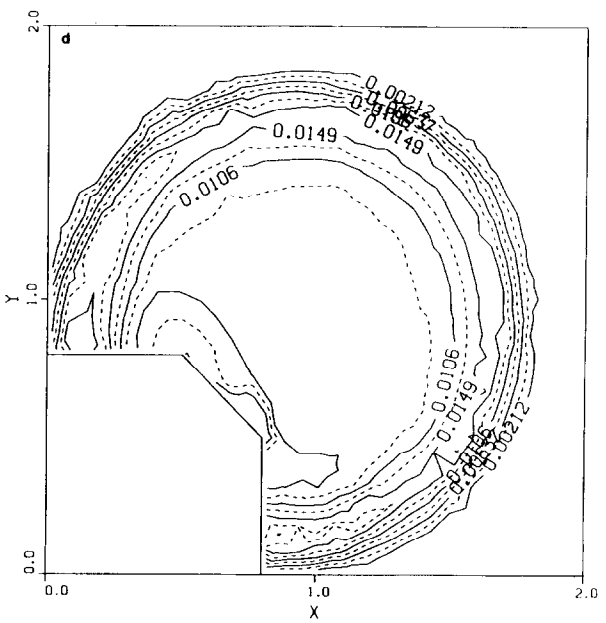


FIG. 10—*Continued*

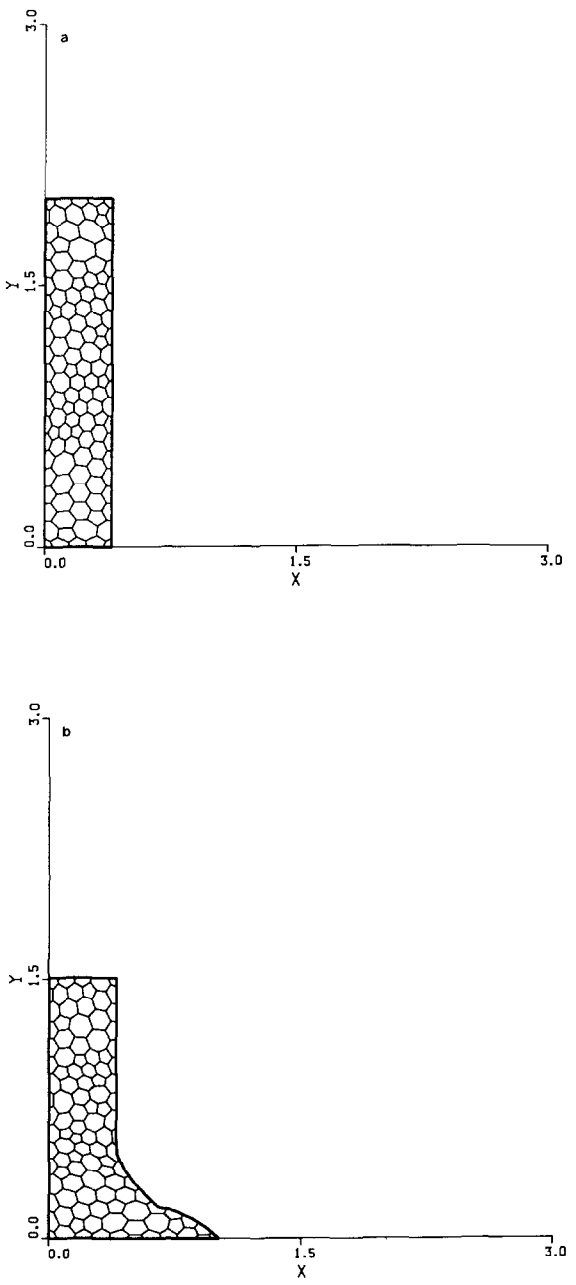
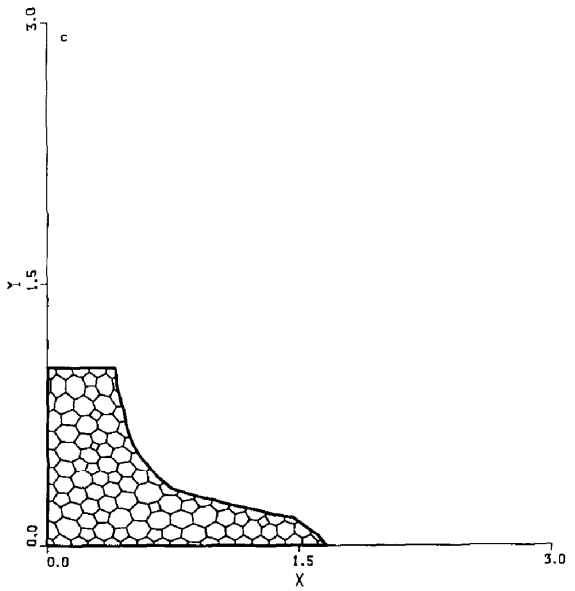


FIG. 11. Evolution of the general-topology mesh for the impact problem using a "second-order," van Leer calculation: (a) initial mesh; (b) $t = 2.5$; (c) $t = 5.0$; (d) $t = 7.5$.



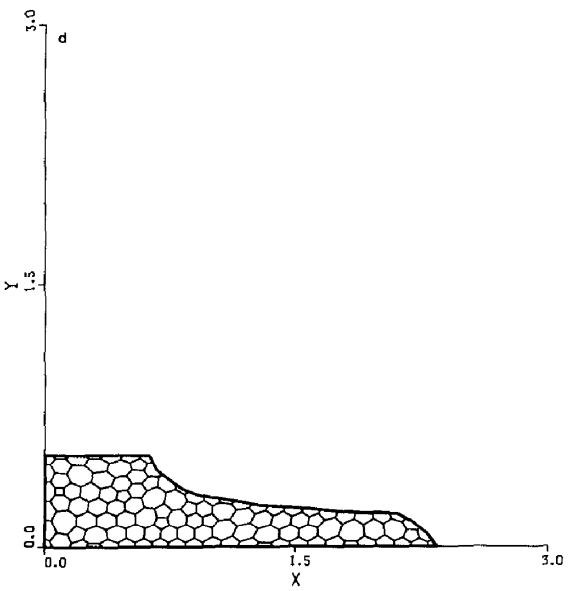
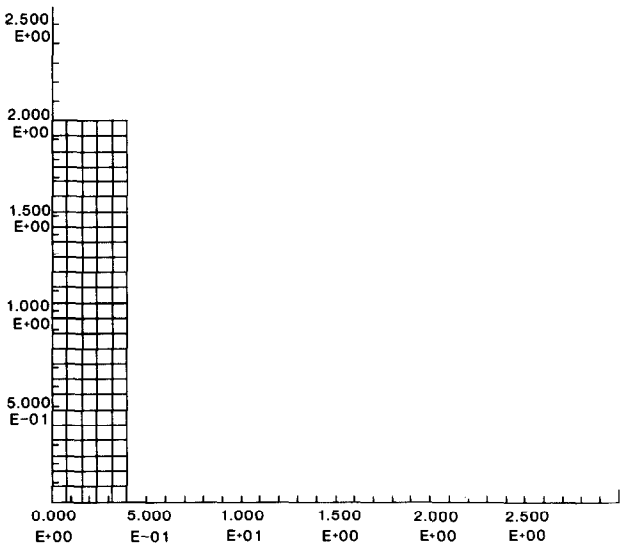
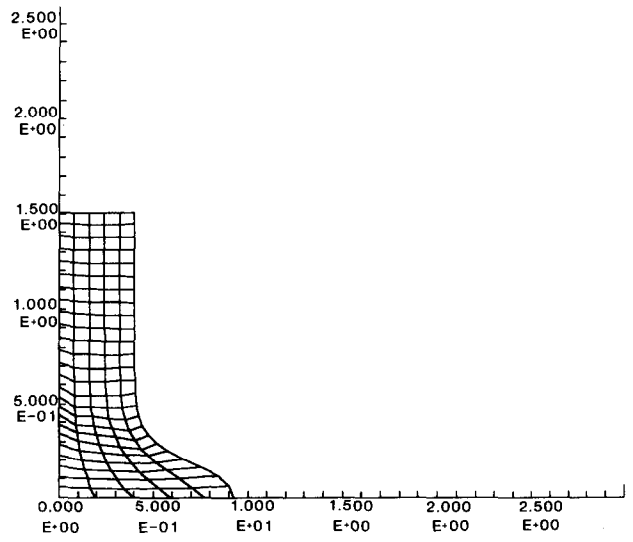


FIG. 11—*Continued*

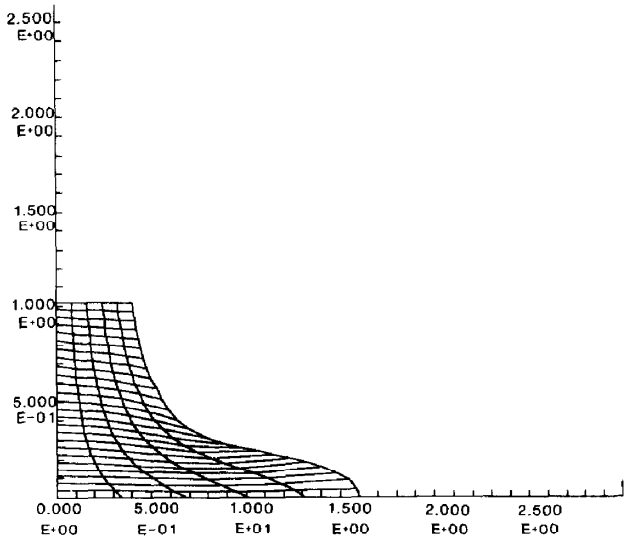


(a)

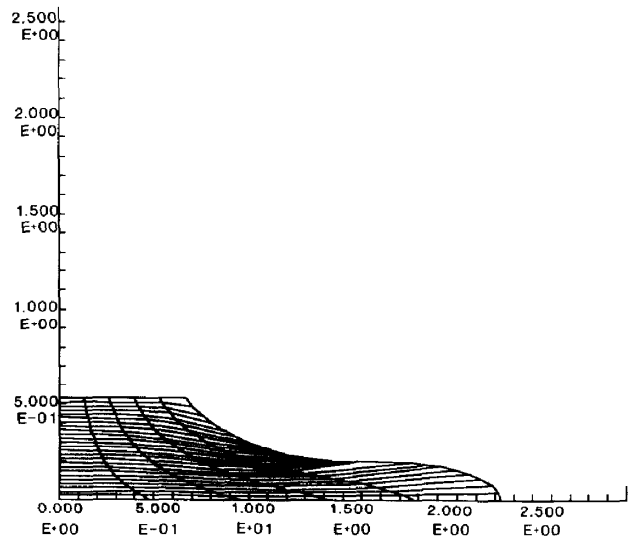


(b)

FIG. 12. Evolution of the fixed-connectivity mesh for the impact problem using a "second-order," van Leer limited calculation: (a) initial mesh; (b) $t = 2.5$; (c) $t = 5.0$; (d) $t = 7.5$.



(c)



(d)

FIG. 12—*Continued*

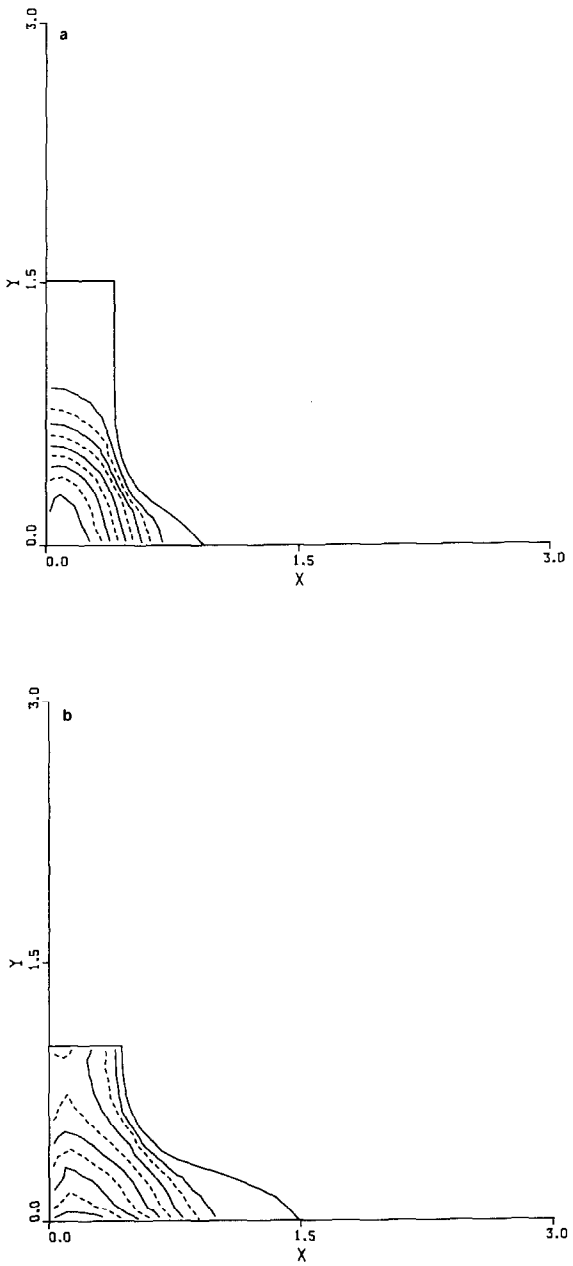


FIG. 13. Pressure contours of the general-topology method for the impact problem using a "first-order" calculation: (a) $t = 2.5$; (b) $t = 5.0$; (c) $t = 7.5$.

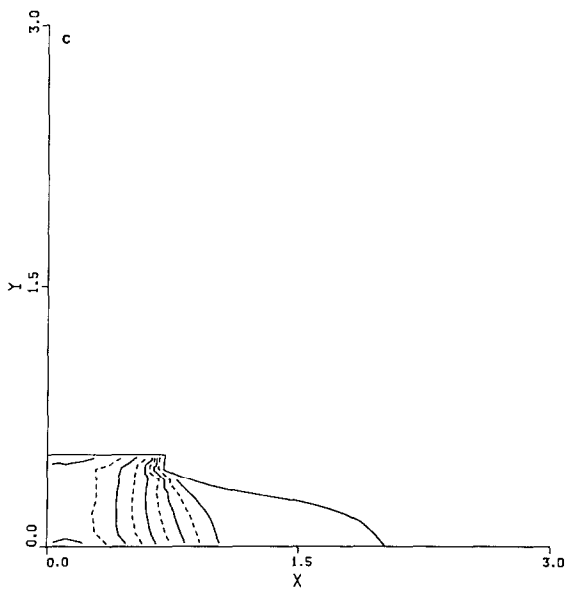


FIG. 13—*Continued*

593 computational cells, with a characteristic cell dimension of approximately 0.08. Initially, the fluid is uniform and quiescent with density and specific internal energy of 1 and 1.5×10^{-10} (dimensionless units), respectively. At $t = 0$, a source of energy is applied to the cell located at approximately $x = y = 0.95$. This computational cell instantaneously acquires an internal energy of 10.

As the calculation evolves, a cylindrical shock wave emanates from the energy source. The "near-Lagrangian" rezone algorithm concentrates cells in the vicinity of the wave, as would be the case in a Lagrangian calculation. As cells are compressed in the neighborhood of the wave, distorted triangles are generated. When this occurs, a global rezone is performed and a more regular mesh is generated. Mesh geometries preceding and following a global rezone are shown in Fig. 7. It is observed that although a smoother mesh is produced, a hint of the original mesh remains. Pressure profiles preceding and following the global remap are shown in Fig. 8 for a "first-order" calculation. The diffusion produced by the remapping is evident.

The evolution of the pressure profiles for "first-order" and "second-order" van Leer limited calculations is shown in Figs. 9 and 10, respectively. The increased amount of diffusion produced in the "first-order" calculation may be observed, particularly at early times. For the complete calculation, the "first-order" and "second-order" calculations invoke the global rezone/remap procedures 38 and 130 times, respectively. Diffusional errors are introduced each time a global remap is used. Consequently, the advantages of a "second-order" calculation are less evident at later times. It may be observed that the wave is not propagated preferentially in

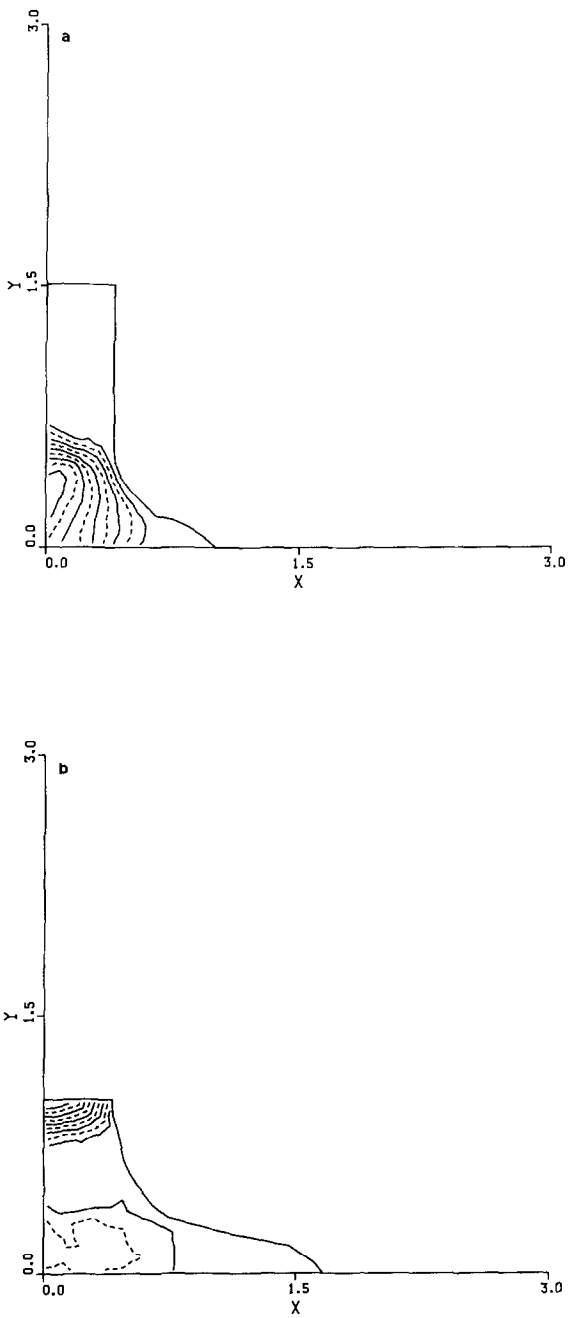


FIG. 14. Pressure contours of the general-topology method for the impact problem using a "second-order," van Leer limited calculation: (a) $t = 2.5$; (b) $t = 5.0$; (c) $t = 7.5$.

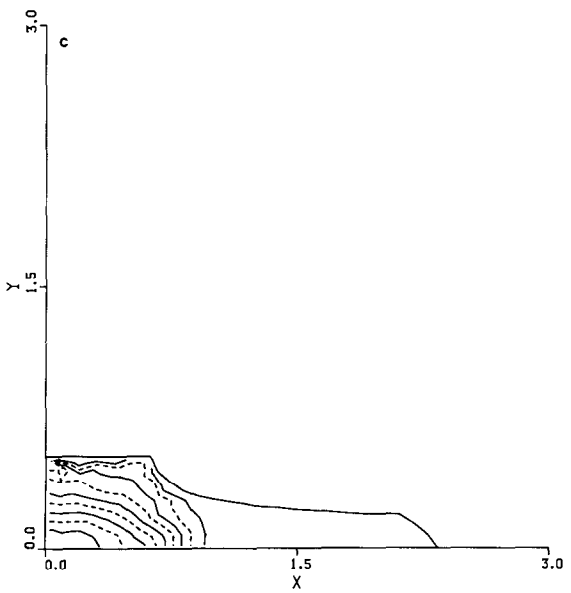


FIG. 14—*Continued*

any one direction on the internal mesh. This demonstrates the improved isotropy anticipated for this method, as compared to the related CAVEAT code [1], based on a quadrilateral mesh, which typically shows differences in shock propagation in directions which are skew to the mesh directions.

4.2. *Impact Problem*

We consider a plate with dimensions 0.4 by 2.0, a density of 8.9, and traveling at the uniform velocity of -0.196 in the vertical (downward) direction. At $t=0$ the plate encounters a rigid wall. The phenomena that one observes are that a shock wave propagates vertically from the point of impact and is then overtaken by a rarefaction wave as the plate "splatters" against the rigid wall. For comparison, this impact problem is computed using both the general topology method (CAVEAT-GT) and a fixed-connectivity, quadrilateral mesh (CAVEAT [1]). The intitial general-topology and fixed-connectivity meshes are shown in Figs. 11a and 12a. Both meshes use computational cells with a characteristic dimension of 0.1. The left and bottom boundaries are symmetry boundaries. The right boundary is a free surface ($p=0$), while the top boundary is a specified velocity boundary ($v=-0.196$). The material obeys the Chaplygin equation-of-state,

$$p=k^2\left(\frac{1}{\rho_0}-\frac{1}{\rho}\right), \tag{25}$$

where $k=3.49$ and $\rho_0=8.9$. The initial velocity corresponds to a Mach number of 0.5, based on the undisturbed sound speed.

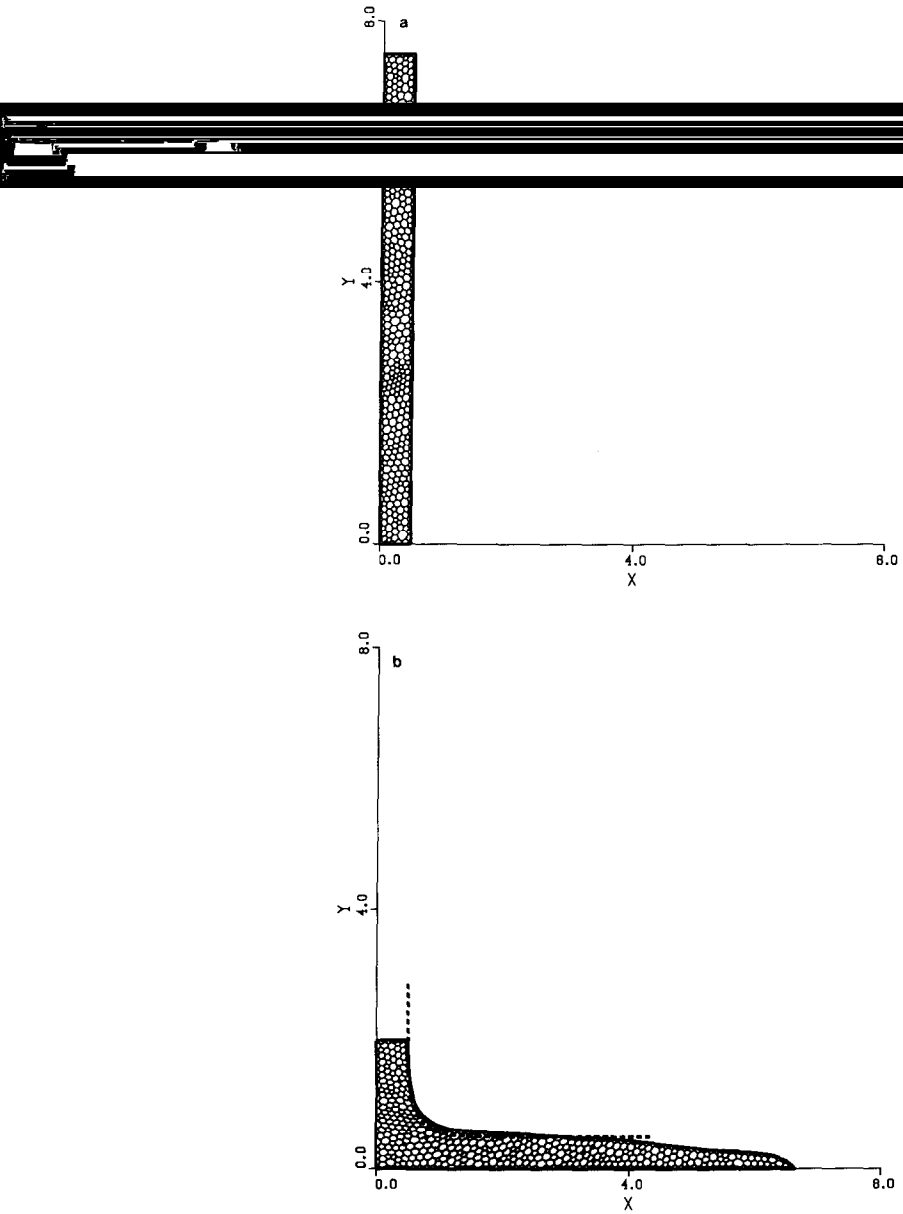


FIG. 15. Solution for the steady-state impact problem using a "first-order," calculation: (a) initial mesh; (b) surface position and mesh; (c) density distribution; (d) velocity distribution along horizontal axis. ———theory; ∘ ∘ calculation.

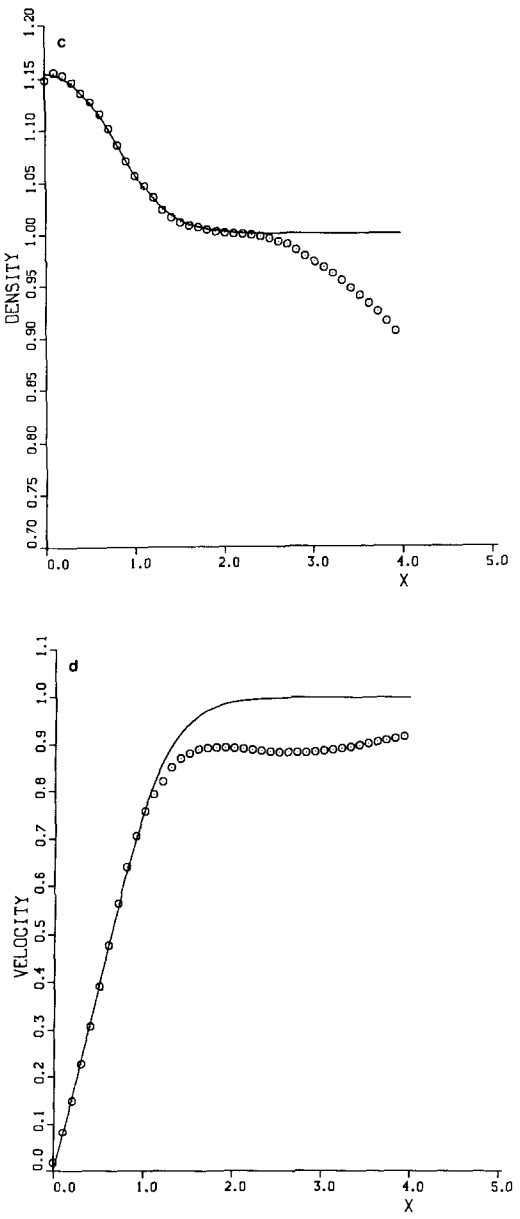


FIG. 15—*Continued*

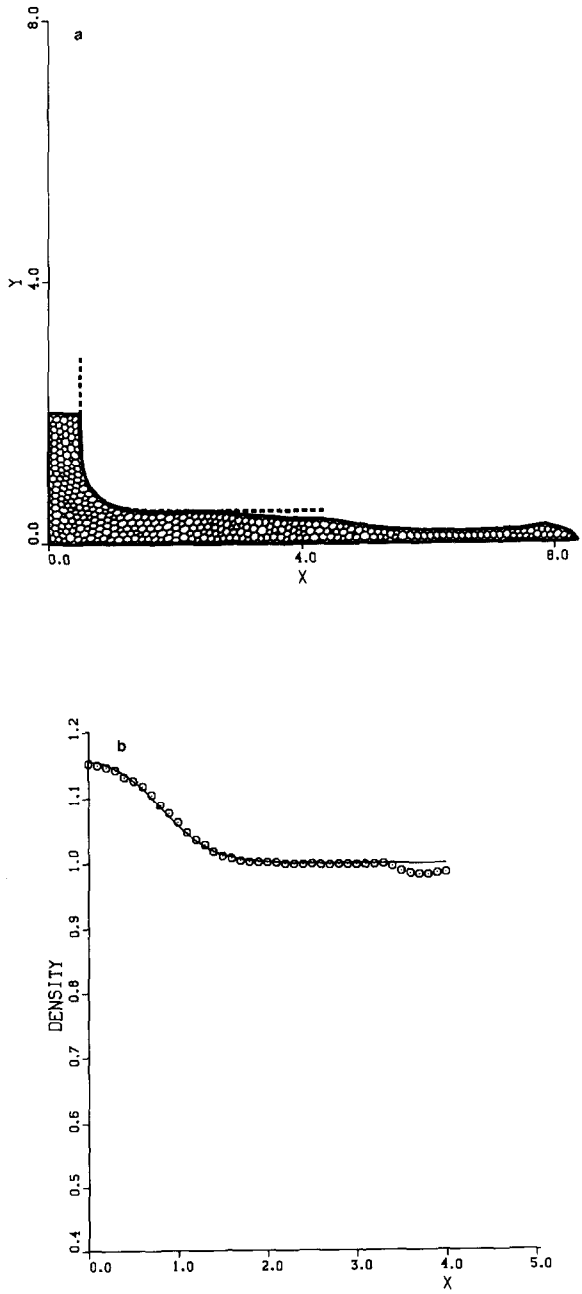


FIG. 16. Solution for the steady-state impact problem using a "second-order," van Leer limited calculation: (a) surface position and mesh; (b) density distribution; (c) velocity distribution along horizontal axis. ———theory; ◦ ◦ calculation.

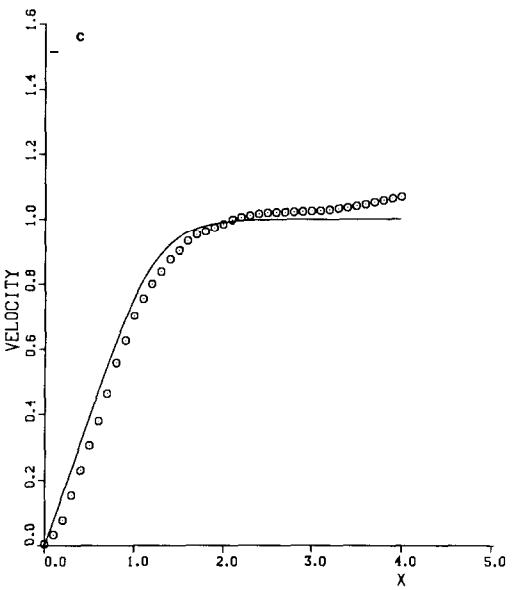


FIG. 16—*Continued*

The evolution of the mesh for a "second-order" van Leer limited calculation using the general-topology mesh is shown in Fig. 11. The ability of the technique to smoothly add and delete computational cells along the boundaries, as well as in the interior, can be observed. For comparison, the evolution of the mesh for the corresponding "second-order" van Leer limited, fixed-connectivity computation is shown in Fig. 12. The fixed-connectivity calculation also utilized a mesh rezoning technique in an effort to maintain a regular mesh in the interior. The limitations of the fixed-connectivity mesh for this type of problem are obvious. One measure of the difference in the computations is the magnitude of the time step required. Since both calculations are explicit, the time-step size decreases as the characteristic cell dimensions decrease, and the computation time correspondingly increases. At $t = 7.5$, the time-step sizes are 2.4×10^{-2} and 3.8×10^{-3} for the general-topology and fixed-connectivity methods, respectively.

Some representative pressure contours for the "first-order" and "second-order" general-topology calculations are shown in Figs. 13 and 14, respectively. In this problem there is a noticeable improvement in using the "second-order" method as compared to the "first-order" method.

This particular impact problem employing a Chaplygin equation of state is advantageous because an analytic solution for the steady state is available [12] and this allows us to assess the accuracy of the calculations. To simulate steady state conditions, the calculation for a plate with a width-to-height ratio of 0.5 : 7.5 (Fig. 15a) is taken to late times. We modify the problem slightly from the earlier one so that, initially, the plate is in uniform motion with velocity equal to -1 and

density equal to 1. The initial mesh is composed of 512 computational cells with a characteristic dimension of 0.1. Again, the Chaplygin equation-of-state (Eq. (25)) is used with $k = 3.49$ and $\rho_0 = 1.0$.

Results for a late time "first-order" and "second-order" van Leer limited calculation are shown in Figs. 15 and 16, respectively. The figures show comparisons of the computed and exact free-surface profiles, as well as the density and velocity distributions on the impact line (i.e., the x-axis). The free-surface position from the "second-order" calculation compares extremely well with the exact free-surface position. Both "first-order" and "second-order" density and velocity comparisons are excellent near the origin. The "second-order" calculation is much more accurate as is evident further out from the origin along the line of impact.

5. Summary and Conclusions

Many fluid dynamics or material deformation problems, particularly those involving interfaces, are severely constrained by the use of a fixed topology mesh. We have successfully adapted an existing 2-dimensional method [1], based on a fixed topology quadrilateral mesh, to the use of a general topology, arbitrary polygonal cell mesh. Several features of the original method helped in this adaptation. These features, which were retained in the new code, include the use of a finite-volume, cell-centered ALE method and the use of the Godunov method in the Lagrangian phase. Several new techniques were developed. These include

(a) the use of Huygens' construction for interface propagation,

(b) several techniques for regularizing, distributing, and smoothing mesh points along interfaces and boundaries,

(c) a "near-Lagrangian" rezoning method which minimizes diffusion due to advection by finding a mesh close to but more regular than the Lagrangian mesh,

(d) a variational method for finding a general smooth mesh within arbitrary boundaries,

(e) an adaptation of the general remapping algorithm [8–11] to a general topology mesh, and

(f) specialized graphics essential for developing and debugging a code using the complex data structure associated with a general topology mesh.

This development effort was largely experimental, intended to produce experimence with such a mesh and to evaluate its usefulness. It is clear that such a code can handle problems not feasible with a conventional Lagrangian or ALE code. In its current state of development, CAVEAT-GT already can handle non-trivial problems. No effort was made to optimize the code for speed and efficiency. It is to be expected that such a code will not be able to compete with conventional codes in this regard because of its additional complexity. Since the code is not

optimized, we do not have an estimate of the penalty involved. (This is a penalty in terms of time/cell/cycle. The code may well execute faster because of improved time step due to a better mesh.) Such a code, therefore, may not supplant existing codes, but it may be expected to supplement them for those problems in which its capability is unique.

REFERENCES

1. F. L. ADDESSIO, D. E. CARROLL, J. K. DUKOWICZ, F. H. HARLOW, J. N. JOHNSON, B. A. KASHIWA, M. E. MALTRUD, AND H. M. RUPPEL, Los Alamos National Laboratory Report No. LA-10613-MS, 1986 (unpublished).
2. C. W. HIRT, A. A. AMSDEN, AND J. L. COOK, *J. Comput. Phys.* **14**, 227 (1974).
3. B. VAN LEER, *J. Comput. Phys.* **32**, 101 (1979).
4. J. K. DUKOWICZ, *J. Comput. Phys.* **61**, 119 (1985).
5. J. U. BRACKBILL AND J. S. SALTZMAN, *J. Comput. Phys.* **46**, 342 (1982).
6. B. DELAUNAY, *Bull. Acad. Sci. USSR (VII), Cl. Sci. Mat. Nat.* **793** (1934).
7. G. VORONOI, *J. Reine Angew. Math.* **134**, 198 (1908).
8. J. K. DUKOWICZ, *J. Comput. Phys.* **54**, 411 (1983).
9. J. D. RAMSHAW, *J. Comput. Phys.* **59**, 193 (1985).
10. J. D. RAMSHAW, *J. Comput. Phys.* **67**, 214 (1986).
11. J. K. DUKOWICZ AND J. W. KODIS, *SIAM J. Sci. Statist. Comput.* **8**, 305 (1987).
12. R. R. KARP, Low Alamos Scientific Laboratory Report No. LA-8371, 1980 (unpublished).