JOURNAL OF COMPUTATIONAL PHYSICS 99, 115–134 (1992)

Vorticity Errors in Multidimensional Lagrangian Codes

JOHN K. DUKOWICZ

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

AND

BERTRAND J. A. MELTZ

Département de Mathématiques Appliquées, Centre d'Etudes de Limeil-Valenton, B.P. 27, 94195 Villeneuve Saint-Georges Cedex, France

Received March 23, 1990; revised October 17, 1990

We investigate the apparent paradox, as exemplified by the well-known Saltzman test problem, of multidimensional lagrangian codes experiencing mesh tangling when computing one-dimensional irrotational flows. We demonstrate that the cause is the generation of spurious vorticity, or vorticity error, by a nonuniform mesh. Based on this, we investigate two methods of constructing improved lagrangian vertex velocities by removing, or filtering out, this spurious vorticity, rather than by the more common practice of introducing artificial viscosity. The first method reconstructs the velocity from the known flow divergence and from the true vorticity computed by means of a transport equation. The second method, which is much simpler and more efficient, subtracts a divergence-free correction from the velocity, such that the resulting velocity possesses the correct vorticity. We then successfully apply this method to solve a two-dimensional shock refraction problem, a problem which exhibits nonzero intrinsic vorticity.

1. INTRODUCTION

We are interested in computational hydrodynamics methods where the mesh is not stationary, but may move with a velocity that is computed as a function of time and position in the course of the calculation. Such methods, therefore, imply the use of distorted or nonuniform meshes. Particular examples include lagrangian methods, or ALE (arbitrary Lagrangian–Eulerian) methods which contain a lagrangian phase.

In lagrangian hydrodynamics methods, a computational cell moves with the flow velocity. In practice, this means that the cell vertices move with a computed velocity, the cell faces being uniquely specified by the vertex positions. It is well known that most multidimensional lagrangian calculations can only be continued for a finite time before the mesh is destroyed by "tangling," or crossing of mesh lines. This is to be expected because in the presence of shear and rotation a lagrangian fluid element will eventually become so stretched and distorted that it may no longer be adequately represented by a discrete computational cell, such as one based on simple polygonal and polyhedral elements. More frequently, however, a premature tangling failure is caused by numerical errors. There are many types of numerical errors that may affect mesh motion. One type is associated with the presence of a null space of the discrete operators (the so-called "hourglass," "checkerboard," or "herringbone" modes). These errors have received much attention in the literature and we will not be concerned with them here. Another type is associated with the use or presence of an irregular or distorted mesh; this type of error is relatively poorly understood and it is this type of error with which we will be concerned.

On the other hand, lagrangian methods avoid a source of numerical error due to the advection terms in the conservation equations. For this reason, lagrangian methods are frequently preferred in one-dimensional computations where mesh distortion plays no role. Thus, in practical multidimensional applications it is often the trade-off between these two sources of error that determines the success of a lagrangian calculation. The ALE method computations attempt a compromise by permitting some freedom in mesh movement but, at the same time, not allowing the mesh to become overly distorted. It is clear that a reduction in the mesh-distortion error would not only extend the scope of lagrangian calculations but improve the accuracy of the ALE computations as well.

The problem that we wish to address is related to an aspect of the error introduced by the use of nonuniform, irregular meshes. A longstanding paradox in lagrangian hydrodynamics concerns the computation of irrotational flows. A lagrangian computation of a flow with zero vor-

0021-9991/92 $3.00

ticity is expected to be particularly long lasting and resistant to mesh tangling. In particular, a one-dimensional flow, inherently irrotational, should be free from mesh tangling. Nevertheless, it is found in practice that a one-dimensional flow, computed using a nonuniform two-dimensional mesh, will usually develop increasing mesh distortion and will eventually fail due to mesh tangling. This is best demonstrated by a frequently used and widely known test problem, called the Saltzman problem [1]. This problem concerns a strong one-dimensional shock wave, propagating in an initially specified, nonuniform two-dimensional plane mesh. Most lagrangian codes have difficulty with this test problem, particularly since deviations from the expected, one-dimensional behavior are so easy to detect. The standard treatment for such mesh difficulties involves introducing artificial viscosities [2, 3], which unfortunately have the effect of preventing or reducing mesh distortion whether that distortion is legitimate or not.

Treating the Saltzman problem as a paradigm, we investigate its pathologies in some detail. We demonstrate that the difficulty in computing accurate lagrangian vertex velocities is caused by a spurious vorticity generated in the presence of an irregular mesh. We further argue that there exists a mechanism for "healing" errors in the divergence of velocity, while errors in vorticity are merely transported and persist in time, thereby accounting for the above-mentioned mesh distortion. Thus, we take the approach of filtering out, or removing, the effects of the spurious vorticity, while retaining the effects of the real or intrinsic vorticity, if present. This, incidentally, highlights a difficulty with the Saltzman test problem. Any correction method which damps or removes vorticity will perform well on this test problem, which should have zero vorticity, but may not necessarily perform well on other problems where it is important to retain the intrinsic vorticity.

In this paper we apply these ideas in the context of a cell-centered mesh code. The same ideas may be applied to staggered meshes, a fact that is best illustrated by the "Turn function and vorticity" method of O'Rourke [4], a work that is similar in spirit to ours, although greatly different in purpose and method.

2. DESCRIPTION OF THE PROBLEM

2.1. The CAVEAT Code

As the vehicle for our study we have chosen to use the CAVEAT code [5], primarily for reasons of convenience, and because we believe that the choice of a code is not important since the problem under study is quite generic to many lagrangian formulations. CAVEAT is a two-dimensional, ALE-method code, based on an arbitrary quadrilateral mesh. In the ALE method, the computational cycle is composed of a lagrangian step, followed by a remapping (or advection) step, in which the quantities calculated in the lagrangian step are remapped or transferred from the lagrangian mesh to an arbitrarily specified mesh. Here we are interested only in the lagrangian step. The lagrangian step is based on the Godunov method, which implies that all conserved quantities, including momentum, and hence cell velocity, are cell-centered, and that cell-face quantities, including a face-normal component of velocity, are available from the solution of a Riemann problem (in this case, an approximate solution) at each cell face.

Thus, we have available a cell-centered average velocity $\mathbf{u}$, a face-centered normal component $\mathbf{n}\cdot\mathbf{u}^*$, and we need to determine a vertex velocity $\mathbf{v}'$ (see Fig. 1). Guided by the equation for the rate of change of a lagrangian volume element:

$$\frac{dV}{dt}=\int \nabla\cdot\mathbf{u}\,d\tau=\int \mathbf{n}\cdot\mathbf{u}^*\,dS,$$

where V is the volume of a moving lagrangian element and the integration is over the surface of the element, we are led to choose the Riemann normal component of velocity $\mathbf{n}\cdot\mathbf{u}^*$ at a face as most appropriate for defining the vertex velocities. There are many possibilities for extracting the vertex velocities. A simple algorithm might be constructed by requiring that the vertex velocity, projected in the direction of a face normal, should equal the Riemann velocity on that face [1]. In CAVEAT we use a weighted least squares algorithm, based on the fact that the problem is overdetermined in the typical case of four faces meeting at a vertex. Setting $w_i=\mathbf{n}\cdot\mathbf{u}_i^*$, where i is the index for faces meeting at a vertex j, we minimize

$$\sum_i W_i[\mathbf{n}_i\cdot\mathbf{v}_j'-w_i]^2$$

with respect to the unknown components of the vector $\mathbf{v}_j'$, where W_i is a weight, defined on each face (currently taken

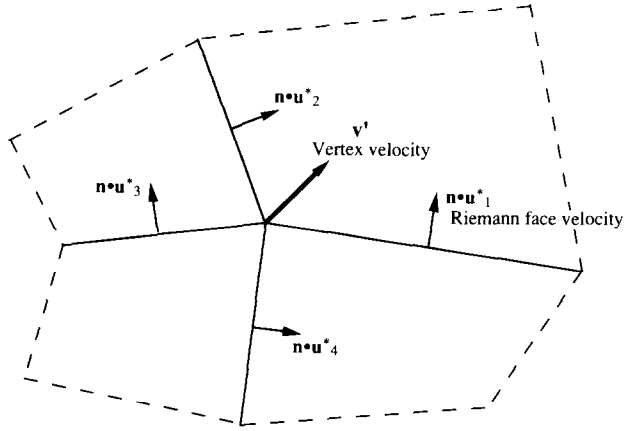


FIG. 1. Velocities associated with a cell vertex.

to be the average of the cell densities on each side of the face). This yields a set of simple linear equations on each face which are solved to obtain the vertex velocities $\mathbf{v}'_j$. This algorithm is independent of the number of faces at a vertex ($i \geqslant 2$), is easily extendible to arbitrary dimensionality, provides a degree of smoothing, is simple and efficient, and has proven to be reliable in practice. It has the obvious fault that it reproduces only constant velocity fields exactly and therefore is at best first-order. For example, for the case of face normal velocities derived from a single component velocity field the algorithm is capable of generating additional spurious components in the vertex velocity field.

2.2. The Saltzman Test Problem

The Saltzman test problem [1] tests the ability of a code to retain a one-dimensional solution to a one-dimensional problem on a nonuniform two-dimensional mesh. The problem consists of a rectangular box or cylindrical whose walls form reflective boundaries and whose left-hand side wall acts as a piston, initially driving a strong shock wave towards the right. Since the upper and lower boundaries are reflective, and the initial conditions are independent of the vertical coordinate direction, the problem is expected to be one-dimensional, independent of the width of the box or the diameter of the cylinder. As time progresses, the shock wave undergoes a series of reflections from both the right- and left-hand walls.

The one-dimensional symmetry is broken by the mesh. The initial mesh is 10 cells high in the y-direction and 100 cells wide in the x-direction and is defined by

$$\left.\begin{aligned} x_{ij} &= (i-1)^* dx + (11-j)^* dy^* \sin\frac{\pi(i-1)}{100}, \\ y_{ij} &= (j-1)^* dy, \end{aligned}\right\} \qquad \begin{aligned} i &= 1, 2, \ldots, 101; \\ j &= 1, 2, \ldots, 11, \end{aligned}$$

where $dx = dy = 0.01$. This initial mesh is displayed in Fig. 2. The working fluid is assumed to be an ideal gas with $\gamma = \frac{5}{3}$, compressed by a piston moving to the right with a velocity of 1. The initial conditions involve a stationary gas with a density of 1 and an internal energy of 10^{-4}. The expected post shock conditions are given by a pressure of 1.333, a density of 4.0, an internal energy of 0.5, and a shock speed of 1.333. The computation may be carried out in plane or cylindrical geometry, with the x-axis as the axis of cylindrical symmetry.

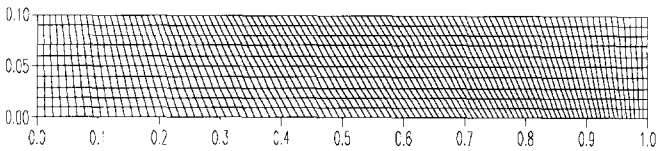


FIG. 2. Initial mesh for the Saltzman test problem (compressed horizontal scale—the horizontal and vertical scales in all figures are labelled in nondimensional units).

2.3. The Standard Result

Figures 3 and 4 illustrate the lagrangian mesh and the density contours obtained with CAVEAT in plane geometry at times of $t = 0.5$ and 0.7. The problem fails due to excessive mesh distortion at a time of about 0.8, shortly after the first shock reflection. Similar results are obtained with other codes [1, 3].

It is obvious that the solution is not one-dimensional. The most striking feature observed in the results is the skewed and rotated nature of the lagrangian mesh, contrary to expectations. Since we know that the same problem computed on an initially orthogonal, undistorted mesh remains perfectly one-dimensional, it is clear that the cause of this abnormal behavior is the initial mesh distortion. On the other hand, the density contours are surprisingly close to the expected result, a uniform density behind the shock, in spite of the highly distorted mesh.

The nature of the distortion of the mesh suggests that it arises as a result of the presence of vorticity in the vertex velocity field. The density, on the other hand, is closely related to the lagrangian cell volume, and therefore to the divergence of the velocity field. Examining these two quantities, the vorticity and divergence, by themselves is not very enlightening, however. In the following section we look at them from a somewhat different perspective.

2.4. The Volume and Skewness Integrals

We wish to display and examine quantities that are directly related to the divergence and vorticity and which, at the same time, take account of the fact that the *effects* of divergence and vorticity, being intimately connected to the mesh velocity, are integrated or accumulated as a function of time. For this purpose we choose to use the lagrangian cell volume to represent the effects of divergence, and we define an analogous quantity to represent the effects of vorticity, which we call the cell skewness.

Recalling that the rate of change of the volume V of a lagrangian element is given by the equation

$$\frac{dV}{dt} = \int \nabla \cdot \mathbf{v}' \, d\tau = \int \mathbf{n} \cdot \mathbf{v}' \, dS,$$

where $d\tau$ is a differential volume element and dS is a corresponding surface element, we define a *skewness integral* by the analogous equation

$$\frac{d\boldsymbol{\Omega}}{dt} = \int \nabla \times \mathbf{v}' \, d\tau = \int \mathbf{n} \times \mathbf{v}' \, dS.$$

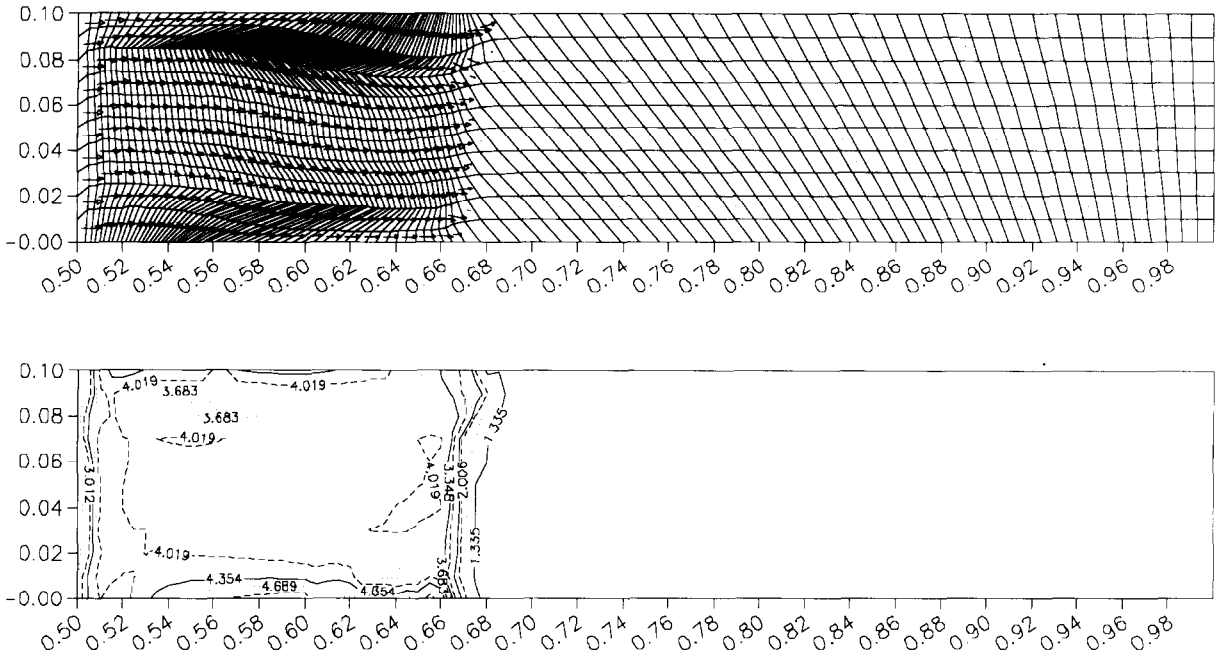


FIG. 3. Lagrangian mesh, velocity vectors, and density contours for the Saltzman problem obtained with the standard CAVEAT code, at time $t = 0.5$.

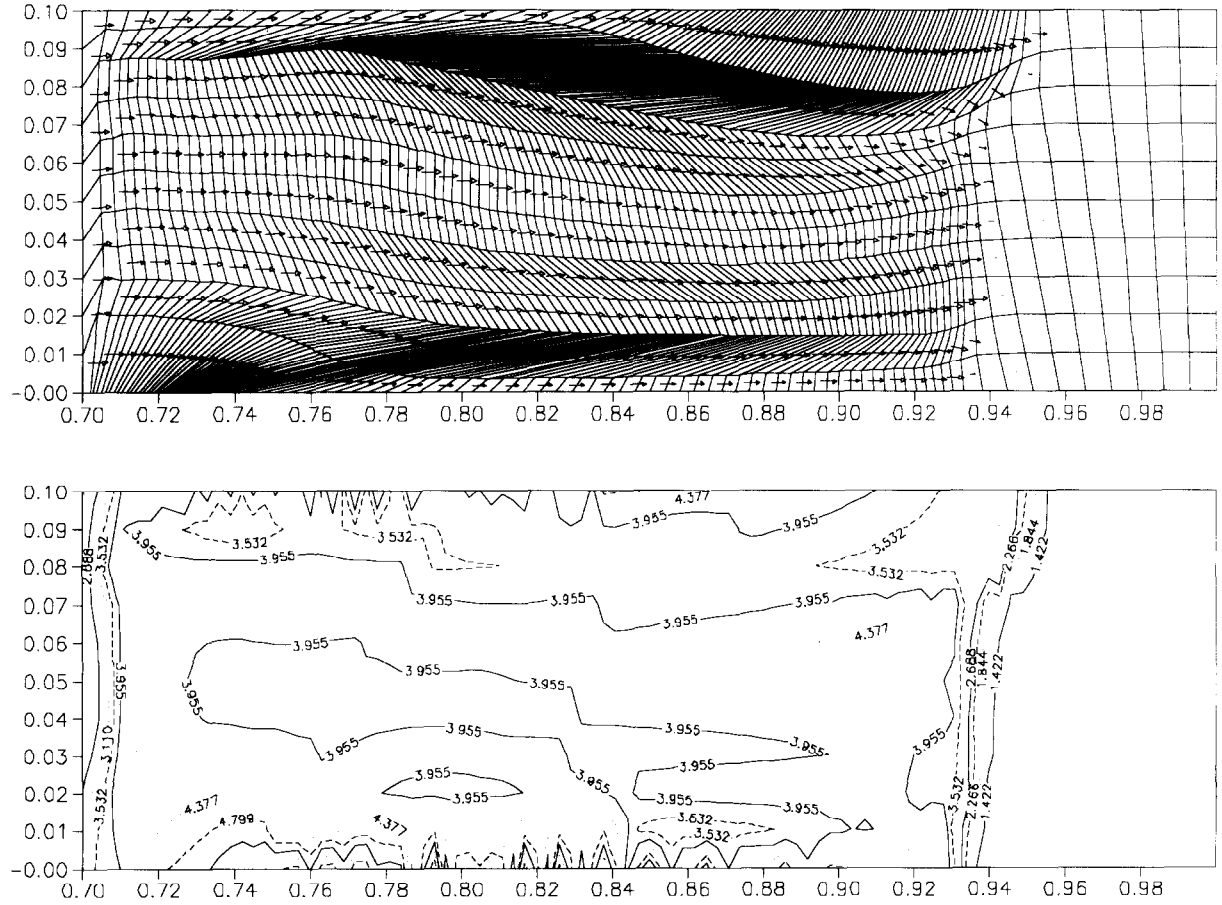


FIG. 4. Lagrangian mesh, velocity vectors, and density contours for the Saltzman problem, obtained with the standard CAVEAT code, at time $t = 0.7$.

Thus, we have two quantities, V and $\mathbf{\Omega}$, which are directly related to the divergence and the vorticity of the lagrangian velocity field, respectively, and at the same time are quantities integrated with respect to time. In the plane two-dimensional case the equation for the skewness can be considerably simplified. Writing $\mathbf{\Omega} = \mathbf{k}\Omega$, where $\mathbf{k}$ is the unit vector normal to the plane and Ω is the magnitude of the skewness vector, we obtain

$$\frac{d\Omega}{dt} = \oint \mathbf{v}' \cdot d\boldsymbol{\lambda}, \tag{2.1}$$

where $d\boldsymbol{\lambda}$ is a vector differential length tangent to the cell boundary.

We integrate Eq. (2.1) forward in time to obtain the quantity Ω. There is no need to similarly integrate the equation for the cell volume since the equation can be shown to be an exact differential and so V may be obtained directly from vertex coordinates. The cell volume, normalized by the initial cell volume, is displayed in Fig. 5 and the cell skewness Ω is displayed in Fig. 6 for problem time $t = 0.7$. It is quite remarkable to observe that the cell skewness Ω is large in two bands near the top and bottom of the mesh, precisely

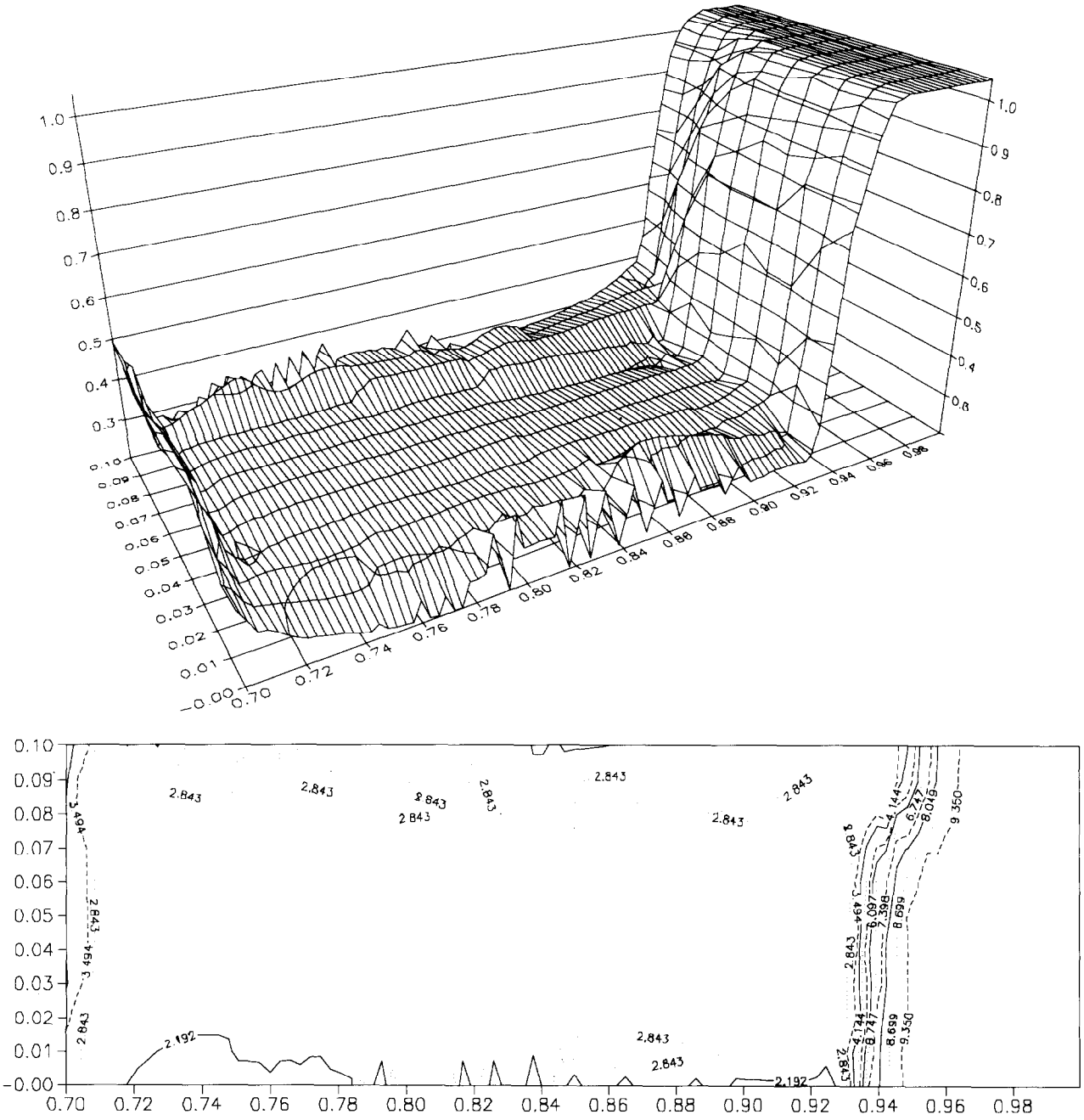


FIG. 5. The normalized lagrangian cell volume for the Saltzman test problem, obtained with the standard CAVEAT code, at time $t = 0.7$.

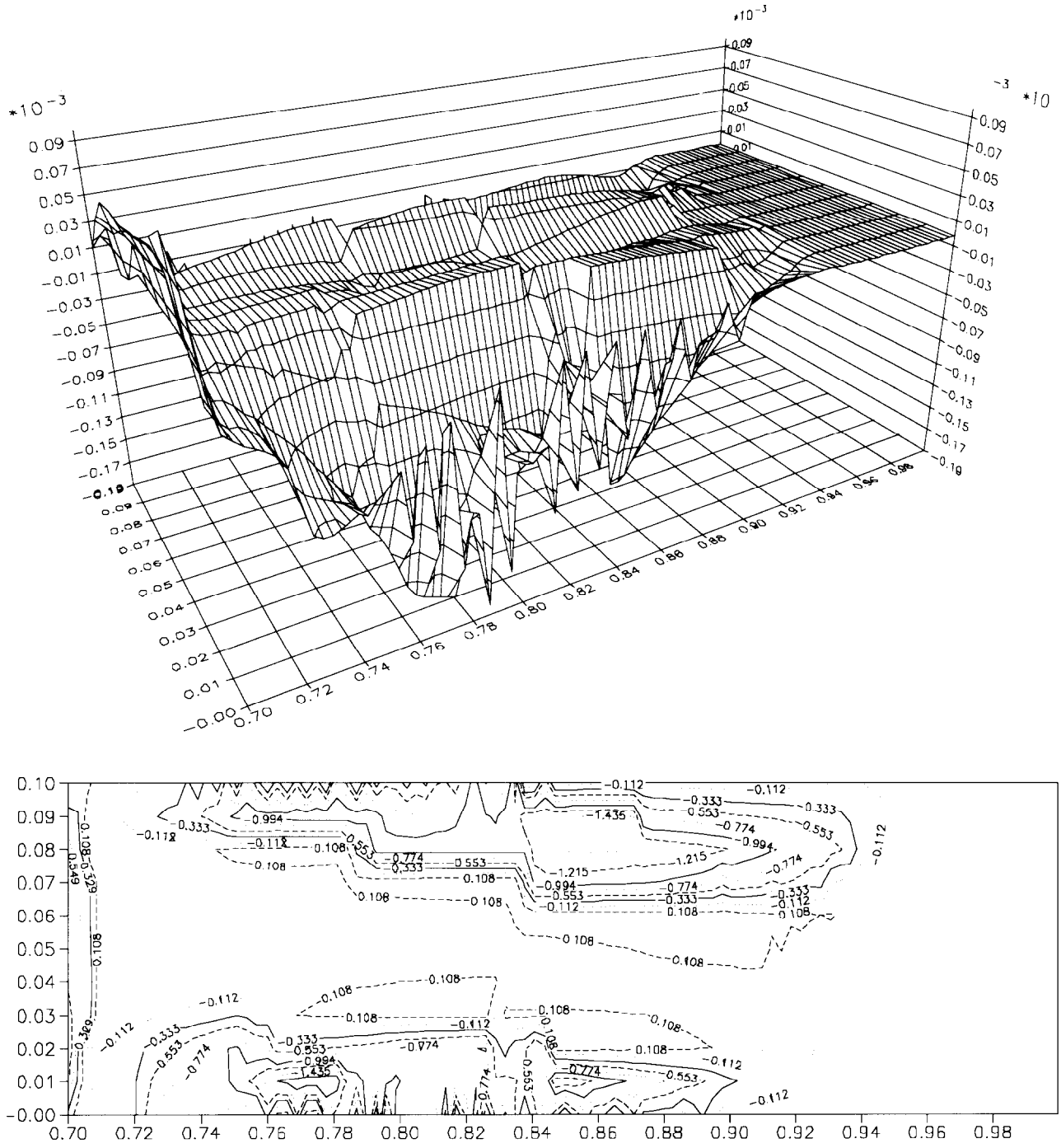


FIG. 6. The skewness integral for the Saltzman test problem, obtained with the standard CAVEAT code, at time $t = 0.7$.

where the mesh is most distorted; it thus appears that the skewness is closely related to the mesh distortion. The cell volume, by contrast, is quite uniformly compressed behind the shock, exactly as one would expect for this problem.

These results indicate that vorticity is being generated in the presence of a nonuniform mesh, and they provide strong evidence that this vorticity is primarily responsible for the subsequent mesh distortion. The divergence, however, appears to be well behaved. Since vorticity and divergence are similar quantities, in the sense that both are derived from velocity gradients, we must now account for the apparent fact that spurious vorticity is being generated and persists in the flow, while the corresponding divergence error is being suppressed. The final proof that the spurious

vorticity is indeed responsible for the mesh distortion will be provided by the success of our method which subtracts out the effects of the spurious vorticity.

3. ANALYSIS OF THE LAGRANGIAN VELOCITY FIELD

3.1. The Role of Divergence and Vorticity

Consider the divergence $D = \nabla \cdot \mathbf{u}$ and vorticity $\boldsymbol{\omega} = \nabla \times \mathbf{u}$ of a vector field $\mathbf{u}$ (in our case, the velocity field). According to the Hodge decomposition theorem [6] (or the Helmholtz theorem [7], in infinite space), a vector field can always be decomposed into a divergence-free component that contains all the vorticity and an irrotational component that contains all the divergence. Conversely, by a theorem stated in Arfken [8] a vector field can be uniquely determined from its divergence and vorticity, and its normal component on the boundary of any finite region. Clearly, then, vorticity and divergence describe important and complementary properties of the velocity field. The divergence, moreover, constitutes the source of lagrangian volume and isentropic pressure changes, according to

$$\frac{dv}{dt} = v \nabla \cdot \mathbf{u} \tag{3.1}$$

and

$$\frac{dp}{dt} = -\rho c^2 \nabla \cdot \mathbf{u}, \tag{3.2}$$

where $v = 1/\rho$ is the specific volume, ρ is the density, p is the pressure, and c is the speed of sound.

Equations for the divergence and vorticity may be easily derived from the equations of motion. Consider the momentum equation from the set of Euler equations,

$$\frac{d\mathbf{u}}{dt} = -\frac{1}{\rho} \nabla p. \tag{3.3}$$

Differentiating this equation, we obtain

$$\frac{dD}{dt} = -\nabla \mathbf{u} : \nabla \mathbf{u} - \nabla \frac{1}{\rho} \cdot \nabla p - \frac{1}{\rho} \nabla \cdot \nabla p \tag{3.4}$$

and

$$\frac{d\boldsymbol{\omega}}{dt} = -\boldsymbol{\omega} D + (\boldsymbol{\omega} \cdot \nabla)\, \mathbf{u} - \nabla \frac{1}{\rho} \times \nabla p \left\{ -\frac{1}{\rho} \nabla \times \nabla p \Rightarrow 0 \right\}. \tag{3.5}$$

These equations may be viewed as expressing the transport of divergence and vorticity, respectively, in the presence of sources (here we lump all the terms involving first-order derivatives and call them sources; strictly speaking, only the third term on the right-hand side of Eq. (3.5) may be considered a vorticity source, because without it an initially irrotational flow will always remain irrotational) and second-order terms involving pressure. The second-order term on the right-hand side of the divergence equation plays a very important role, as discussed in the following section. The second-order term in the vorticity equation (shown in parentheses) is identically zero, except for discretizations on nonuniform meshes [9], when it may be considered as a numerical or spurious source of vorticity. Special discretizations may be devised which do not have this error [10]; however, this is not so in the typical case. Further, there may be other truncation errors in the discretizations of both Eqs. (3.4) and (3.5), particularly in the presence of a nonuniform mesh, that may be spurious sources of divergence and vorticity.

This suggests that we may expect the spurious generation of both vorticity and divergence. This is in contrast to what we observed in Section 2; namely, while we found the existence of vorticity error, the divergence behaved in the expected manner, indicating that there was little if any divergence error. We address this question in the following section.

3.2. Qualitative Behavior of the Equations of Vorticity and Divergence

Let us consider the equation for the divergence, Eq. (3.4). Together with Eq. (3.2) it may be written as

$$\frac{dp}{dt} = -\rho c^2 D,$$

$$\frac{dD}{dt} = S_d - \frac{1}{\rho} \nabla \cdot \nabla p,$$

where S_d represents the source terms. Eliminating the divergence D, we obtain

$$\frac{d}{dt}\left[\frac{1}{\rho c^2} \frac{dp}{dt}\right] = -S_d + \frac{1}{\rho} \nabla \cdot \nabla p,$$

which we can recognize as a nonlinear wave equation for the pressure, representing the propagation of acoustic waves. Because of the relationship between pressure and divergence represented by Eq. (3.2), we may expect the divergence to also satisfy a similar if more complicated nonlinear wave equation. The vorticity equation, on the other hand, because of the lack of a corresponding second-order term, satisfies a simple transport equation

$$\frac{d\boldsymbol{\omega}}{dt} = S_\omega,$$

where S_ω represents the vorticity sources. The above implies that in the absence of sources, disturbances in pressure or divergence are smoothed out, or dispersed, by acoustic waves, while disturbances in vorticity are merely transported. From another point of view, the situation is explained by the fact that an inviscid fluid, such as the one we are considering here, is capable of supporting longitudinal or acoustic waves, but not transverse or shear waves.

From the point of view of a numerical code the above facts imply that there exist means to damp or disperse extraneous perturbations of divergence, while perturbations of vorticity are likely to coast and persist in time. This is precisely the situation that we observe in Section 2 in relation to the Saltzman problem.

4. CONSTRUCTING THE VERTEX VELOCITY FIELD

4.1. Obtaining the Desired Vorticity

In order to reconstruct the velocity field we will need to obtain accurate vorticity and divergence distributions. The problem is less severe for the divergence. We have seen from the Saltzman test problem example that, although there may be a production of erroneous divergence there is also a fluid dynamic mechanism, through the pressure and associated acoustic waves, for dispersing these errors. We can therefore compute cell-average divergence directly, using available information such as the Riemann face velocities, by means of

$$D = \langle \nabla \cdot \mathbf{u} \rangle = \frac{1}{V} \int \mathbf{n} \cdot \mathbf{u}^* \, dS, \tag{4.1}$$

where V is the cell volume, and $\mathbf{n} \cdot \mathbf{u}^*$ is the face Riemann velocity. Alternatively, the provisional vertex velocity field, as obtained using the least squares algorithm (Section 2.1), may also be considered to possess the correct divergence.

On the other hand, errors in vorticity persist and accumulate in time to produce gross distortions in the lagrangian mesh. The test problem is irrotational, so that we know that *any* vorticity produced is extraneous. However, in more general situations the flow is not irrotational, and so we have the problem of identifying and separating the extraneous vorticity from that which is naturally generated by the flow.

Typically, in such problems an artificial viscosity is introduced to damp the mesh distortion, and since no attempt is made to separate out the extraneous vorticity, it is inevitable that some of the intrinsic vorticity is damped as well. In our case, we take the approach of filtering out only the extraneous vorticity by attempting to calculate the intrinsic vorticity directly from its sources by means of a transport equation for the vorticity, such as Eq. (3.5),

$$\frac{d\boldsymbol{\omega}}{dt} = -\boldsymbol{\omega} D + (\boldsymbol{\omega} \cdot \nabla)\, \mathbf{u} - \nabla \frac{1}{\rho} \times \nabla p,$$

which may be written as

$$\frac{dv\boldsymbol{\omega}}{dt} = v \left[(\boldsymbol{\omega} \cdot \nabla)\, \mathbf{u} - \nabla \frac{1}{\rho} \times \nabla p \right],$$

where $v = 1/\rho$ is the specific volume, or if preferred, in the case of a finite volume formulation, which may be expressed as

$$\frac{d}{dt} \int \boldsymbol{\omega} \, d\tau = \int \mathbf{u}(\mathbf{n} \cdot \boldsymbol{\omega}) \, dS - \int \nabla \frac{1}{\rho} \times \nabla p \, d\tau,$$

$$= \int \mathbf{u}(\mathbf{n} \cdot \boldsymbol{\omega}) \, dS - \int \mathbf{n} \times \frac{1}{\rho} \nabla p \, dS. \tag{4.2}$$

In two dimensions, because only one component of vorticity exists, orthogonal to the plane, these equations simplify considerably

$$\frac{d\omega}{dt} = -\omega D - \mathbf{k} \cdot \left(\nabla \frac{1}{\rho} \times \nabla p \right), \tag{4.3}$$

$$\frac{dv\omega}{dt} = -v\mathbf{k} \cdot \left(\nabla \frac{1}{\rho} \times \nabla p \right), \tag{4.4}$$

$$\frac{d}{dt} \int \omega \, d\tau = -\int \mathbf{k} \cdot \left(\nabla \frac{1}{\rho} \times \nabla p \right) d\tau,$$

$$= -\oint \rho^{-1} \, dp, \tag{4.5}$$

where ω is the magnitude of the single remaining vorticity component and $\mathbf{k}$ is the unit vector normal to the plane. These equations are further modified for numerical purposes, as described in Section 4.4.

4.2. Velocity Reconstruction Methods

a. *Reconstructing the Velocity Field from the Divergence and Vorticity*

We are assured by Arfken's theorem [8] that, given the vorticity and divergence in a region of space and the normal velocity on the boundary of that region, we can uniquely reconstruct the velocity in that region. Further, the Hodge decomposition tells us that we can write an arbitrary vector field $\mathbf{v}$ as

$$\mathbf{v} = \mathbf{v}_d + \mathbf{v}_r; \qquad \nabla \times \mathbf{v}_d = 0, \qquad \nabla \cdot \mathbf{v}_r = 0;$$

that is, $\mathbf{v}_d$ is irrotational and $\mathbf{v}_r$ is solenoidal or divergence-free. (We will attempt to maintain the convention

intruduced in Section 2, whereby we use the symbol $\mathbf{u}$ to represent cell-centered velocities and $\mathbf{v}$ to represent vertex velocities. This distinction is only meaningful for discrete velocity fields; there will of course be no distinction in the differential case. In this subsection it is useful to use $\mathbf{v}$ for velocity fields even in differential equations, since they are intended to be used to construct vertex velocity fields.) Aris [11] shows that any irrotational vector field may be written as the gradient of a scalar field and any solenoidal vector field may be written as the curl of a solenoidal vector field. That is, we may always write

$$\mathbf{v} = \nabla\varphi + \nabla \times \mathbf{A}; \qquad \nabla \cdot \mathbf{A} = 0, \tag{4.6}$$

or, in other words,

$$\begin{aligned} \mathbf{v}_d &= \nabla\varphi, \\ \mathbf{v}_r &= \nabla \times \mathbf{A}; \qquad \nabla \cdot \mathbf{A} = 0, \end{aligned} \tag{4.7}$$

where φ is called the scalar potential and $\mathbf{A}$ is the vector potential.

Taking $\mathbf{v}$ to be the desired velocity field, it remains to determine the potentials φ and $\mathbf{A}$. The potentials obviously satisfy the equations

$$\begin{aligned} \nabla \cdot \nabla\varphi &= D, \\ \nabla \times \nabla \times \mathbf{A} &= \boldsymbol{\omega}. \end{aligned} \tag{4.8}$$

Assuming that D and ω are specified divergence and vorticity distributions, these equations may be viewed as determining the two potentials, given suitable boundary conditions. In view of the identity,

$$\nabla \times \nabla \times \mathbf{A} = \nabla(\nabla \cdot \mathbf{A}) - \nabla \cdot \nabla\mathbf{A},$$

and Eq. (4.7), the equation for the vector potential may rewritten in a simpler form so that the equations become

$$\begin{aligned} \nabla \cdot \nabla\varphi &= D, \\ \nabla \cdot \nabla\mathbf{A} &= -\boldsymbol{\omega}. \end{aligned} \tag{4.9}$$

A set of boundary conditions specifying the normal component of velocity on the boundary is sufficient to completely determine the solution, according to Arfken's theorem [8]. Such a specification of the boundary conditions is not unique, but a convenient set of boundary conditions, equivalent to the above, is

$$\left.\begin{aligned} &\mathbf{n} \cdot \nabla\varphi = w, \quad w \text{ specified}, \\ &\mathbf{n} \cdot \nabla \times \mathbf{A} = 0, \end{aligned}\right\} \text{ on the boundary.} \tag{4.10}$$

In other words, the scalar potential supports the entire normal velocity component on the boundary, while the vector potential contributes a velocity with zero normal component on the boundary. However, using Stokes's theorem

$$\int_S \mathbf{n} \cdot \nabla \times \mathbf{A}\, dS = \oint_C \mathbf{A} \cdot d\boldsymbol{\lambda},$$

applied to an arbitrarily small contour on the boundary surface, we can see that the above boundary conditions are equivalent to the simpler set

$$\left.\begin{aligned} &\mathbf{n} \cdot \nabla\varphi = w, \quad w \text{ specified}, \\ &\mathbf{A} = 0, \end{aligned}\right\} \text{ on the boundary.} \tag{4.11}$$

In summary, knowing the divergence D and the vorticity $\boldsymbol{\omega}$, we must solve Poisson equations (Eq. (4.9)) for the scalar and vector potentials, respectively, the former with Neumann boundary conditions, and the latter with Dirichlet boundary conditions (Eq. (4.11)). In two dimensions, there will be two equations to solve, one for the scalar potential and one for the remaining single component of the vector potential. In three dimensions, there will be, in principle, only three equations to be solved since the components of the vector potential are not independent but are connected by the gauge condition ($\nabla \cdot \mathbf{A} = 0$). Having obtained the potentials, the velocity is then calculated by the use of Eq. (4.6).

What we have done, in effect, is to obtain the unique velocity field $\mathbf{v}$ consistent with the divergence D, obtained from Eq. (4.1) and therefore from Riemann velocities which directly determine the vertex velocity field $\mathbf{v}'$ (Section 2.1), and the vorticity $\boldsymbol{\omega}$, coming from a vorticity transport equation such as Eq. (4.5), for example. We have therefore filtered out the vorticity error, $\Delta\boldsymbol{\omega} = \boldsymbol{\omega}' - \boldsymbol{\omega}$, where the vorticity $\boldsymbol{\omega}'$ is the one associated with the vertex velocity field $\mathbf{v}'$. (Had we used $\boldsymbol{\omega}'$ instead of $\boldsymbol{\omega}$, we would have obtained a velocity field identical to $\mathbf{v}'$, except for numerical errors.)

b. *Constructing the Velocity by Subtracting the Vorticity Error*

An alternative and computationally cheaper procedure involves correcting the vertex velocity field $\mathbf{v}'$, obtained from the least-squares algorithm (Section 2.1), for example, for the error in vorticity only. According to our arguments, the errors in divergence tend to be corrected by the acoustic wave mechanism of the hydrodynamics, so that, in the previous section we computed the expected vorticity from a transport equation and neglected to do the same for the divergence. Taking this argument further, we can accept the irrotational part of the velocity and correct only the solenoidal part.

Assume that we compute a vorticity $\boldsymbol{\omega}'$ from the vertex velocity $\mathbf{v}'$, and a vorticity $\boldsymbol{\omega}$ from a transport equation, as above. We then have a vorticity error, $\Delta\boldsymbol{\omega} = \boldsymbol{\omega}' - \boldsymbol{\omega}$. We can compute a solenoidal velocity correction as follows:

$$\begin{aligned} &\Delta\mathbf{v} = \nabla\times\mathbf{A}, \qquad \nabla\cdot\mathbf{A} = 0, \\ &\nabla\cdot\nabla\mathbf{A} = -\Delta\boldsymbol{\omega}, \qquad \mathbf{A} = 0 \text{ on the boundary}, \end{aligned} \tag{4.12}$$

and, therefore, the desired velocity is

$$\mathbf{v} = \mathbf{v}' - \Delta\mathbf{v}. \tag{4.13}$$

The advantage, here, is that, in two dimensions, for example, only one Poisson equation needs to be solved. Further, this equation is easier to solve when using iterative techniques because it satisfies Dirichlet and not Neumann boundary conditions.

4.4. The Numerical Algorithm

a. *The Vorticity Equation*

We may base our discretization on Eq. (4.5), in the form,

$$\frac{d}{dt}\omega V = -\mathbf{k}\cdot\left(\nabla\frac{1}{\rho}\times\nabla p\right)V$$

or

$$\frac{d}{dt}\omega V = -\oint \rho^{-1}\,dp, \tag{4.14}$$

where V is the cell volume, and the gradients are assumed to be cell-averaged quantities. The difference is that the second equation is in conservation form while the first is not. The accurate computation of the source term in the above equations is crucial to the success of our method. However, we have found that the source term in the above more-or-less standard form suffers from numerical errors that make its use problematical in practice.

There are two primary reasons why this difficulty exists. In the first case, in regions of low Mach number the pressure is essentially constant, the gradient of pressure is very small, and therefore the numerical evaluation of the gradient may be inaccurate due to loss of precision. Second, for materials whose equations of state are dominated by density changes (very large bulk modulus of elasticity), pressure gradients are correspondingly dominated by contributions from density gradients, which, however, should make no contribution to the above vorticity source term. This large gradient swamps or hides the small component that does contribute to the source term, with a resulting loss of numerical accuracy.

This difficulty may be avoided by writing the gradient of the pressure as

$$\nabla p = \frac{\partial p}{\partial\rho}\nabla\rho + \frac{\partial p}{\partial e}\nabla e,$$

where e is the specific internal energy. Based on this, the source term becomes

$$\nabla\frac{1}{\rho}\times\nabla p = -\frac{1}{\rho^2}\frac{\partial p}{\partial e}\nabla\rho\times\nabla e,$$

where, for an ideal gas, for example, the thermodynamic derivative is $\partial p/\partial e = (\gamma - 1)\rho$. We have found that the above source term works well in regions of smooth flow but not in the presence of discontinuities such as shocks or contact surfaces, in which case the source term in conservation form is much better. Therefore, for the same reasons as above, Eq. (4.14) is written in the form

$$\frac{d}{dt}\omega V = -\oint \rho^{-1}\frac{\partial p}{\partial\rho}\,d\rho - \oint \rho^{-1}\frac{\partial p}{\partial e}\,de, \tag{4.15}$$

where the second integral vanishes for the case of an ideal gas. Because state quantities are defined at cell centers we define the discrete vorticity equation at cell vertices as

$$\begin{aligned} \frac{\omega^{n+1}V^{n+1} - \omega^n V^n}{\Delta t} = &-\sum_k \left\langle \rho^{-1}\frac{\partial p}{\partial\rho}\right\rangle_k \Delta_k\rho \\ &-\sum_k \left\langle \rho^{-1}\frac{\partial p}{\partial e}\right\rangle_k \Delta_k e, \end{aligned} \tag{4.16}$$

where the superscript n is a time index, Δt is the time step, the index k refers to a face of the vertex-centered cell (connecting cell centers), and $\Delta_k\rho$ and $\Delta_k e$ are density and internal energy differences corresponding to cell-face k, in an anti-clockwise direction. The cell centered values of $\omega^{n+1}V^{n+1}$ are obtained by simple averaging of the corresponding quantities from its four vertices. Note that we do not need to know the advanced time volume V^{n+1} separately since, as will be seen in the next section, only the product of vorticity and cell volume is needed.

b. *The Laplacian Operator on a Nonuniform Mesh*

Since we are dealing with a nonuniform mesh, formulating an accurate Laplacian operator and the associated boundary conditions requires some care. An important requirement is that the operator should not possess a null space; in other words, there should be no functions, other than constants, that zero the operator. If they exist, such extraneous functions could then appear in the solution, damaging its usefulness. It is also desirable that the operator

be symmetric, in analogy with the differential operator, but we have found this requirement to be in conflict with other considerations, particularly with the need to be consistent with normal velocity boundary conditions on general curved boundaries. That is, the normal velocity at a boundary deduced from the solution should be the same as the corresponding velocity applied as a boundary condition. This was the primary consideration that determined the construction of the operator described in the following.

The equations that we wish to discretize are the Poisson equations for the scalar and vector potentials (Eq. 4.9). Using the equation for the scalar potential as an illustration, we discretize it in the form

$$\int \nabla \cdot \nabla\varphi \, d\tau = \int D \, d\tau, \tag{4.17}$$

where the integration is performed over a computational cell. Applying the divergence theorem, this is converted to

$$\int \mathbf{n} \cdot \nabla\varphi \, dS = \int \mathbf{n} \cdot \mathbf{v} \, dS, \tag{4.18}$$

where $\mathbf{n}$ is the unit outward normal on each cell face, $\mathbf{n} \cdot \mathbf{v}$ is the normal component of velocity on a cell face, and the integration is over the faces of the cell. Considering a typical mesh geometry in the neighborhood of a cell, illustrated in Fig. 7, we define a secondary mesh whose vertices are partly the computational cell vertices and partly the cell centers. The cell-center coordinates are defined to be

$$\hat{\mathbf{r}}_k = \frac{1}{4} \sum_{i=1}^{4} \mathbf{r}_i,$$

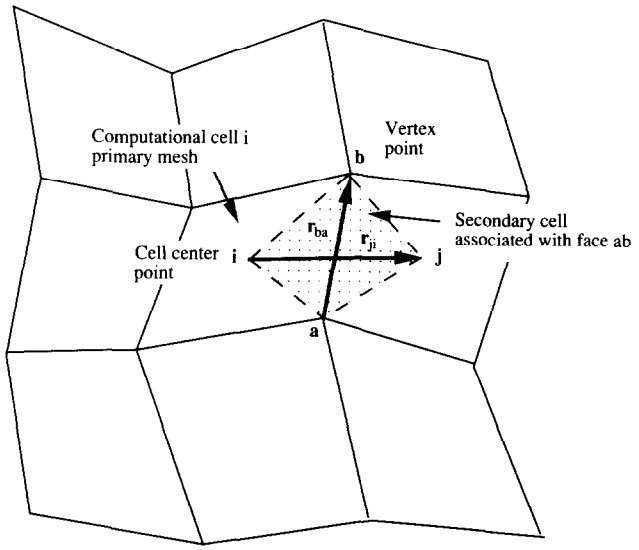


FIG. 7. The secondary cell used for the construction of face gradients.

where $\mathbf{r}_i$ are the cell-vertex coordinates. We assume that the potentials φ are located at the cell-centers. Dropping the circumflex notation, a typical secondary-mesh cell has vertices at cell-centers $\mathbf{r}_i$ and $\mathbf{r}_j$ with potentials φ_i and φ_j, and cell-vertices $\mathbf{r}_a$ and $\mathbf{r}_b$ with potentials φ_a and φ_b, interpolated from surrounding cell-center potentials, as described later. The gradient of the potential is conveniently obtained from the cell area-average

$$\nabla\varphi = \frac{1}{A} \int_{\text{cell}} \mathbf{n}\varphi \, d\lambda. \tag{4.19}$$

With the notation $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, this becomes

$$\nabla\varphi = \frac{1}{2A} \mathbf{k} \times [(\varphi_j - \varphi_i)\, \mathbf{r}_{ab} + (\varphi_b - \varphi_a)\, \mathbf{r}_{ji}], \tag{4.20}$$

where $\mathbf{k}$ is the unit normal orthogonal to the plane, and $2A$ is twice the cell area, given by

$$2A = \mathbf{k} \cdot (\mathbf{r}_{ij} \times \mathbf{r}_{ab}).$$

The outwardly directed surface area is

$$\mathbf{n}\, dS = R_{ab} \mathbf{k} \times \mathbf{r}_{ab}, \tag{4.21}$$

where R_{ab} is the radial coordinate of the face center in axisymmetric geometry ($R_{ab} = 1$ for a cartesian mesh).

The method of interpolation to obtain vertex potentials is closely related to one described in the CAVEAT manual [5]. Consider a typical vertex point a and its four surrounding cell points. The interpolation may be expressed as

$$\varphi_a = \sum_{i=1}^{4} c_i \varphi_i, \tag{4.22}$$

where

$$c_i = \frac{\Lambda(k, j; a)\, \Lambda(l, k; a)}{\Lambda(k, j; i)\, \Lambda(l, k; i)},$$

$$\Lambda(\alpha, \beta; \gamma) = (x_\gamma - x_\alpha)(y_\gamma - y_\beta) - (x_\gamma - x_\beta)(y_\gamma - y_\alpha),$$

and $i = 1, 2, 3, 4$; $j = i + 1 = 2, 3, 4, 1$; $k = i + 2 = 3, 4, 1, 2$; $l = i + 3 = 4, 1, 2, 3$; that is, i, j, k, l represent the four cell points in cyclic order. The coefficients c_i may be viewed as originating from "shape" functions associated with the quadrilateral formed by the four cell points.

Combining these results, the Laplacian operator associated with cell i may be expressed as

$$L_i\varphi = \sum_{j=1}^{9} a_{ij} \varphi_j, \tag{4.23}$$

where a_{ij} is a coupling coefficient between points i and j, and j ranges over all nine points in the neighborhood of point i. The coupling coefficients a_{ij} are related because

$$\sum_{j=1}^{9} a_{ij} = 0,$$

due to the vanishing of the operator for constants, so that the coefficient a_{ii} may be eliminated, and Eq. (4.23) may be alternatively expressed as

$$L_i\varphi = \sum_{j=1,\, j\neq i}^{8} a_{ij}(\varphi_j - \varphi_i), \tag{4.24}$$

where the summation is over only the eight neighbors of point i (the reason for the coupling coefficient terminology now becomes apparent). On a simple orthogonal mesh the operator reduces to the standard five-point Laplacian operator, which is known to be free of a null space. The operator is a matrix operator, $L = \{a_{ij}\}$; the matrix is nonsymmetric ($a_{ij} \neq a_{ji}$) on a nonuniform mesh.

The right-hand sides are evaluated assuming that the divergence and the vorticity are constant within cell i, so that the equations become

$$\begin{aligned} \sum_{j=1,\, j\neq i}^{8} a_{ij}(\varphi_j - \varphi_i) &= V_i D_i = \sum_{k=1}^{4} (\mathbf{n}\cdot\mathbf{v})_k R_k \lambda_k, \\ \sum_{j=1,\, j\neq i}^{8} a_{ij}(A_j - A_i) &= -V_i \omega_i, \end{aligned} \tag{4.25}$$

where φ_i is the scalar potential, A_i is the vector potential, D_i is the divergence, ω_i is the vorticity, V_i is the cell volume (or area, in the cartesian case) of cell i, R_k is the pseudo-radius of Eq. (4.21), and λ_k is the length of face k.

c. *Boundary Conditions*

Strictly speaking, the above equations apply only in the interior of the mesh, away from the boundaries. We follow conventional finite difference practice and view Eq. (4.24) as a numerical "stencil" for the operator, applying boundary conditions explicitly to the numerical equations. The boundary conditions must be consistent with the manner in which the right-hand side of Eq. (4.25) is evaluated. Noting the structure of Eq. (4.18), we observe that there is a one-to-one correspondence between face potential gradients and face normal velocity components, making this consistency easy to achieve. Indeed, as pointed out earlier, this was the main reason for basing the discretization on Eq. (4.18).

For convenience in specifying the boundary conditions we assume that the mesh is surrounded by a layer of fictitious or "ghost" cells. We are free to locate the ghost cell point in a relatively arbitrary manner. Thus, cell points are reflected orthogonally across the boundary face, since this corresponds to the mesh being locally orthogonal at the boundary. For a locally orthogonal mesh the gradient given by Eq. (4.20) becomes a function of only the two cell-point potentials, φ_i and φ_b, where φ_i is the interior point potential and φ_b is the boundary or ghost point potential. The boundary conditions then become, simply,

$$\begin{aligned} \varphi_b &= \varphi_i, && \text{symmetry,} \\ \varphi_b &= -\varphi_i, && \text{reflective, or zero potential,} \\ \varphi_b &= \varphi_i + d_{ib}^{*} w && \text{specified normal gradient,} \end{aligned}$$

where d_{ib} is the distance from the interior to the boundary point, and w is the specified normal gradient (normal component of velocity in the scalar potential case).

d. *Solution Method*

We have chosen to use a standard direct method [12] for solving our matrix equations because we are primarily interested in demonstrating the principle of our method and less in its efficiency. Further, because we are applying the method to two-dimensional problems only, and because iterative methods are less attractive due to the nonsymmetry of our matrices, a direct method becomes both feasible and attractive. However, care must be taken because a direct method of solution of the Poisson equation for the scalar potential may break down. The scalar potential equation is in the form of a Neumann problem, implying that the matrix possesses a zero eigenvalue (associated with a constant potential). That is, the potential is indeterminate up to a constant. Nevertheless, a solution exists provided a consistency condition, namely,

$$\int_{\text{domain}} D\, d\tau = \int_{\text{boundary}} \mathbf{n}\cdot\mathbf{v}\, dS,$$

is satisfied. Fortunately, this consistency condition is implicitly built into the problem formulation, based on Eq. (4.18). However, a direct method would still fail, because, strictly speaking, the system is singular. An easy solution is to regularize the system by pinning the potential at some point in the mesh to a fixed value; in effect removing one equation from the system.

e. *Vertex Velocities*

Following the solution for the potentials, the vertex velocities are obtained by means of Eq. (4.6). Defining a secondary cell whose four vertices are the cell centers sur-

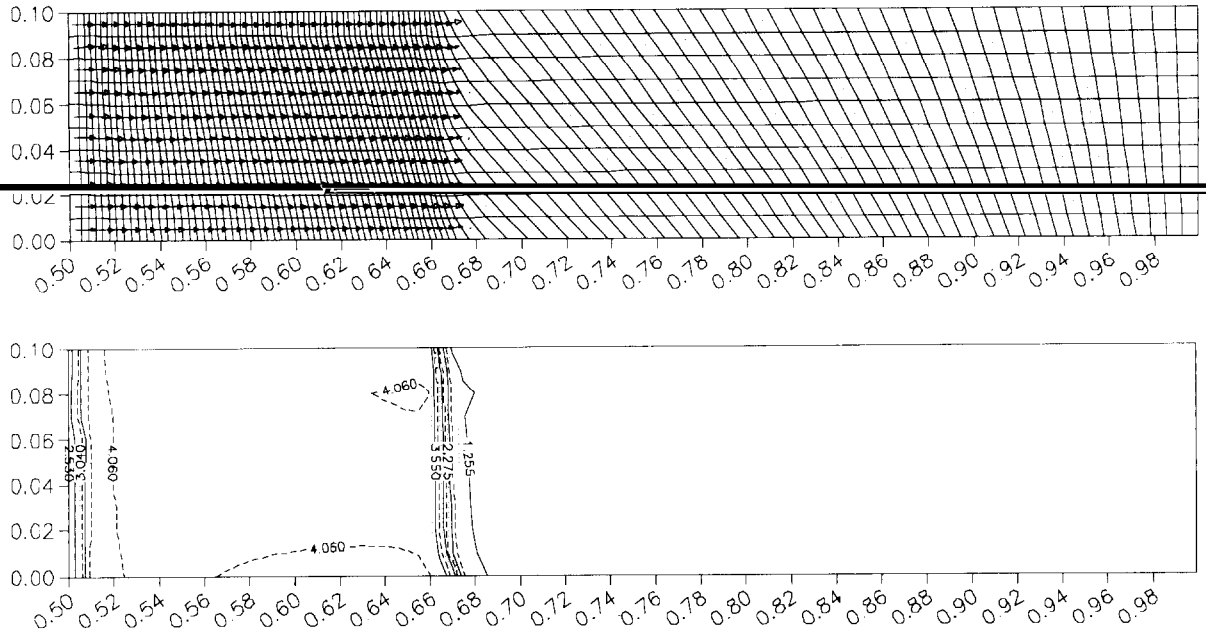


FIG. 8. Lagrangian mesh, velocity vectors, and density contours for the Saltzman problem, obtained using method (a) applied to the CAVEAT code, at time $t = 0.5$.

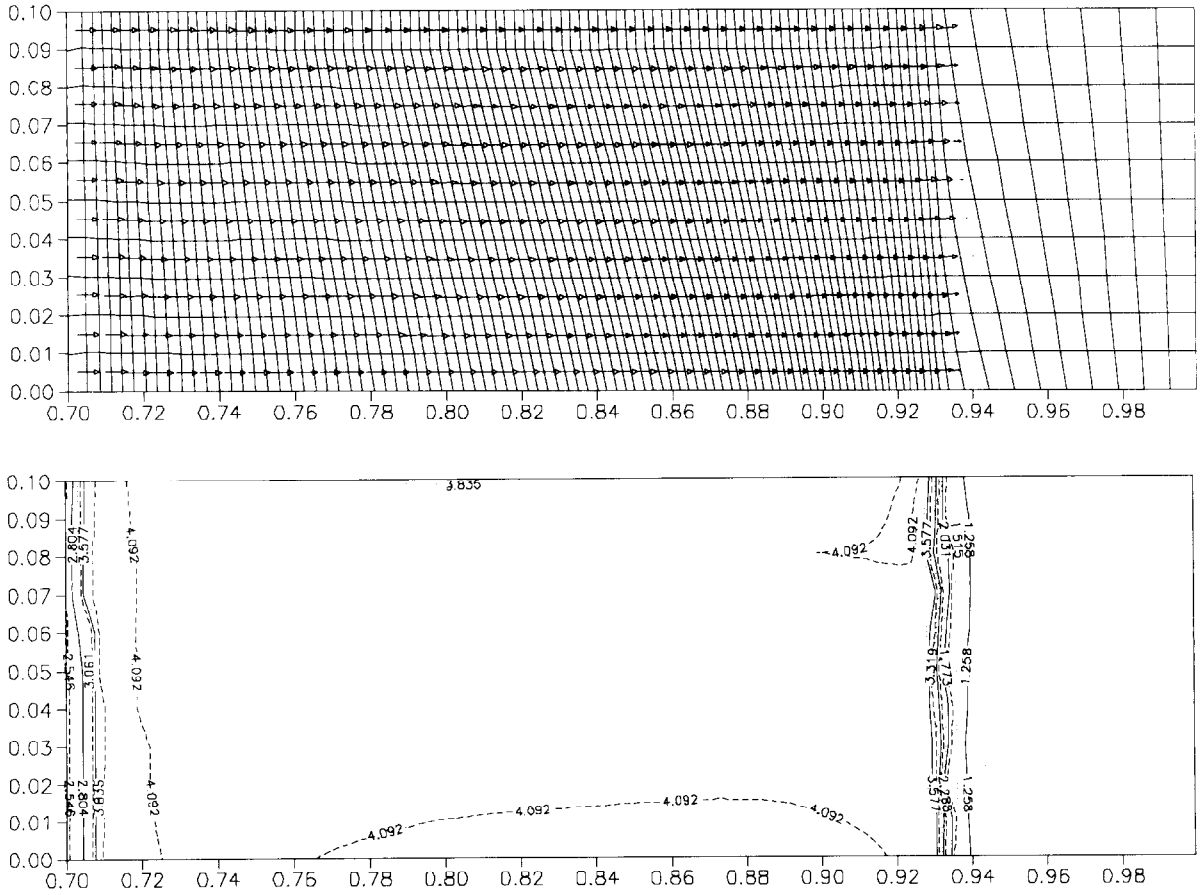


FIG. 9. Lagrangian mesh, velocity vectors, and density contours for the Saltzman problem, obtained using method (a) applied to the CAVEAT code, at time $t = 0.7$.

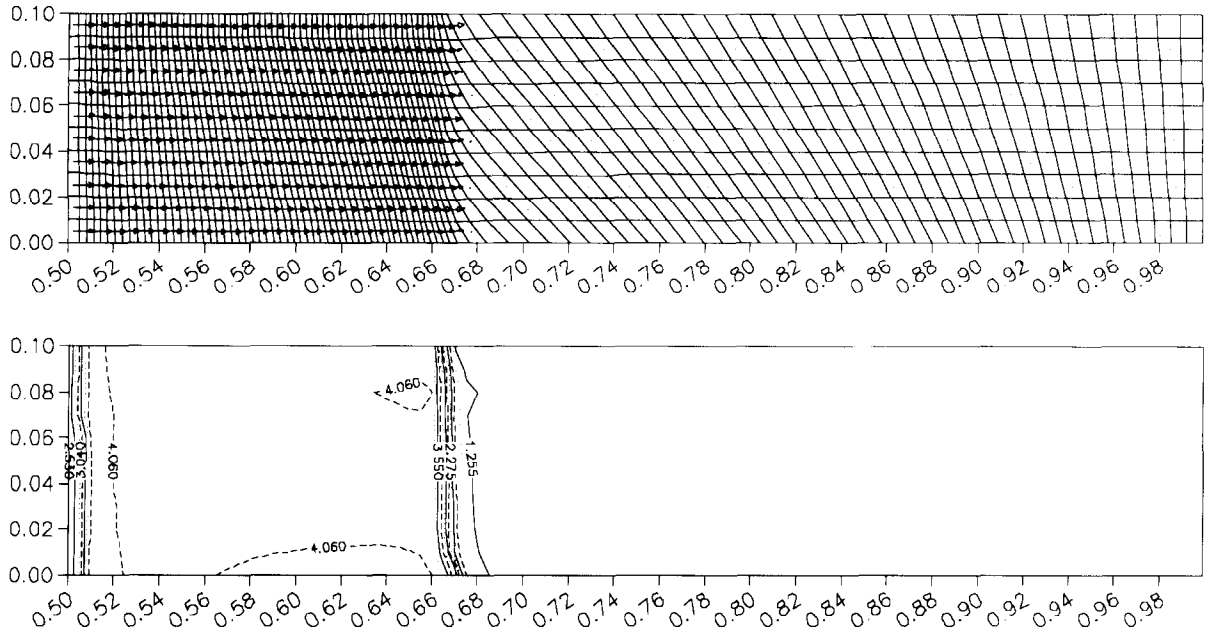


FIG. 10. Lagrangian mesh, velocity vectors, and density contours for the Saltzman problem, obtained using method (b) applied to the CAVEAT code, at time $t = 0.5$.

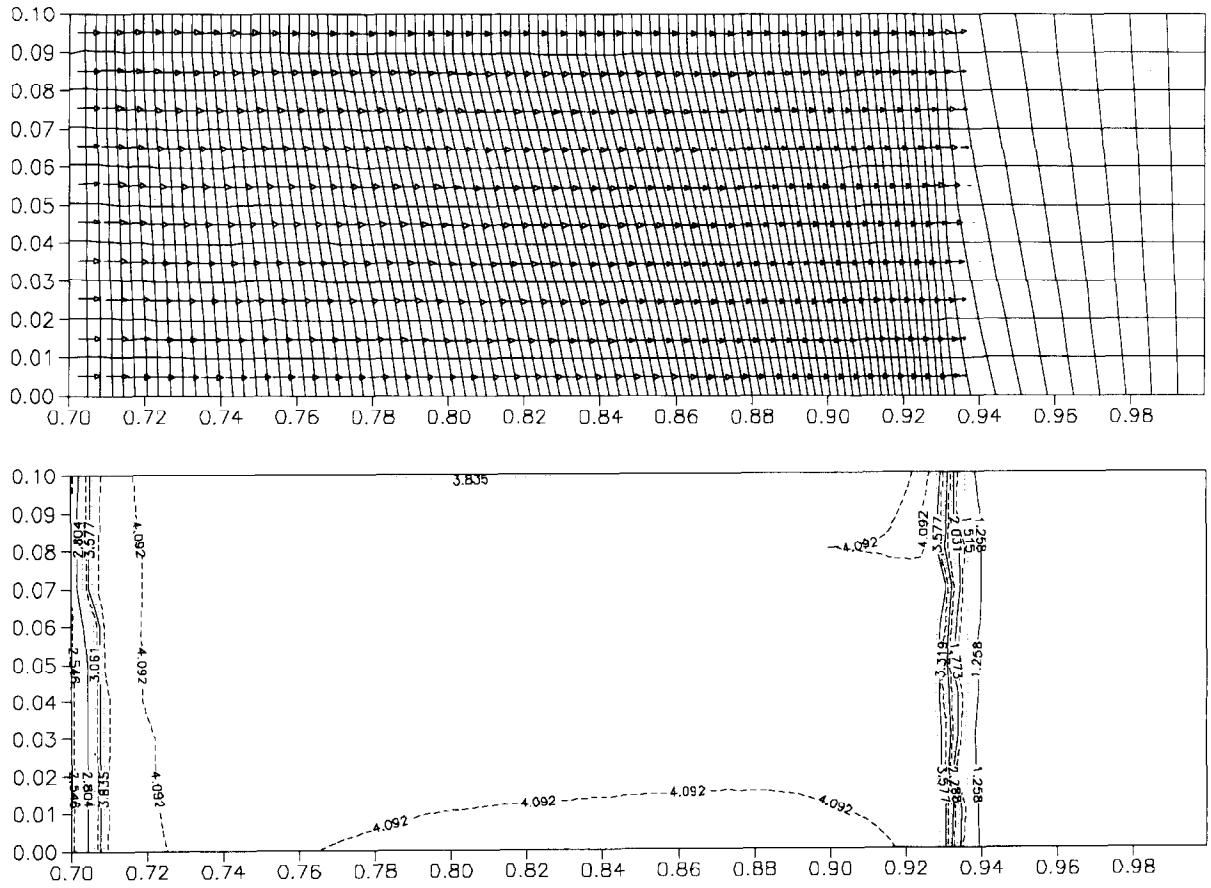
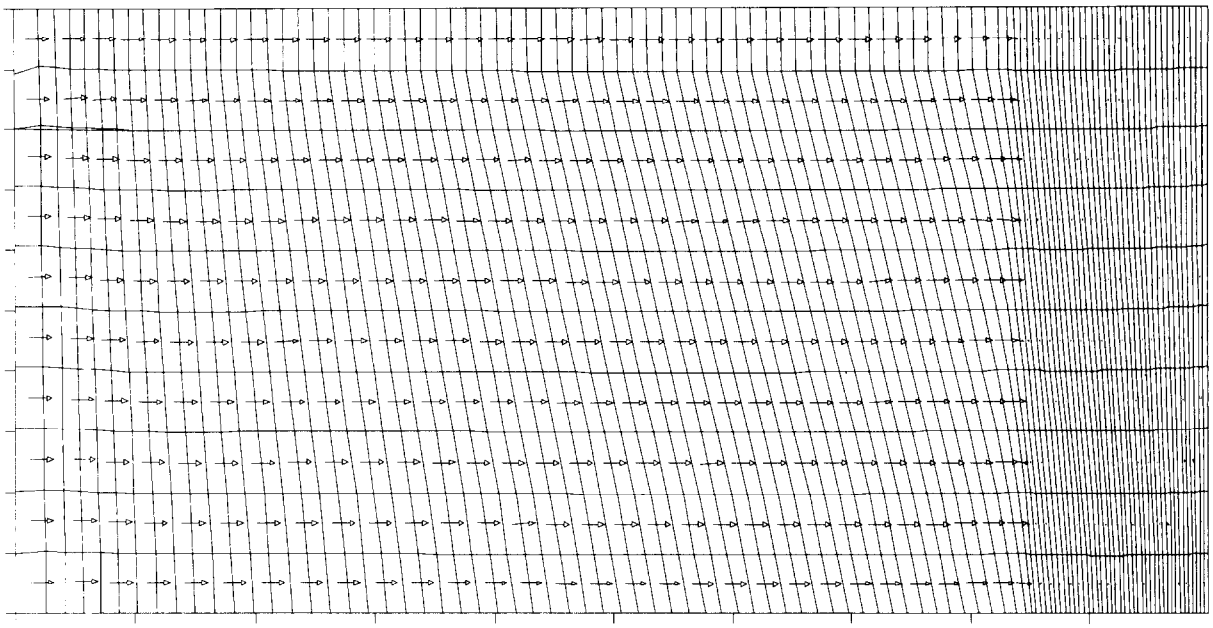
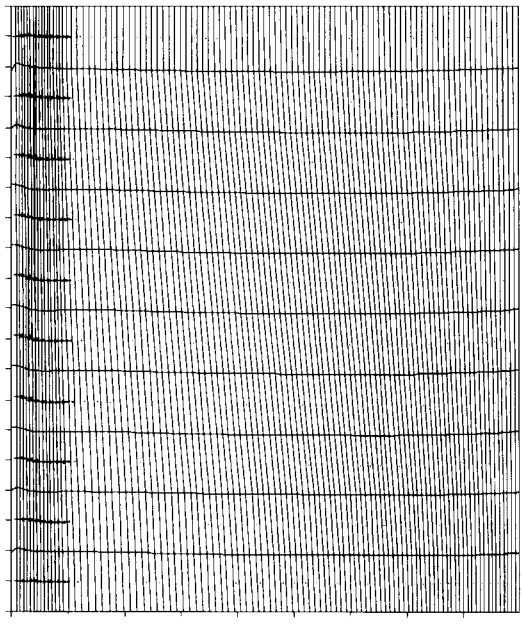


FIG. 11. Lagrangian mesh, velocity vectors, and density contours for the Saltzman problem, obtained using method (b) applied to the CAVEAT code, at time $t = 0.7$.



(a) t=0.8



(b) t=0.9

FIG. 12. Lagrangian mesh obtained using method (b) applied to the CAVEAT code, at times $t = 0.8$ and 0.9, shortly after the first and second shock reflections, respectively.

rounding vertex a, the vertex velocity at point a is obtained from cell averaged gradients analogous to Eq. (4.20)

$$\mathbf{v}_a = \nabla\varphi + \nabla \times \mathbf{A},$$

$$\nabla\varphi = \frac{1}{2A}\,\mathbf{k} \times [(\varphi_j - \varphi_i)\,\mathbf{r}_{kl} + (\varphi_l - \varphi_k)\,\mathbf{r}_{ji}], \quad (4.26)$$

$$\nabla \times \mathbf{A} = \frac{1}{2A}\,[(A_j - A_i)\,\mathbf{r}_{kl} + (A_l - A_k)\,\mathbf{r}_{ji}],$$

where i, k, j, l are the cell points in cyclic order, and

$$2A = \mathbf{k} \cdot (\mathbf{r}_{ij} \times \mathbf{r}_{kl}).$$

5. COMPUTATIONAL EXAMPLES

5.1. The Saltzman Problem

The Saltzman problem was described in Section 2 and the results using the standard CAVEAT method are shown in Figs. 3 and 4. The corresponding results using method (a) are shown in Figs. 8 and 9, and using method (b) they are shown in Figs. 10 and 11. The improvement is striking. It may be noted that both of the new methods give equally good results. However, in view of the fact that method (a) is more complex and more costly, we will henceforth restrict our attention to method (b) only.

Recalling that the standard calculation failed due to mesh tangling at approximately the time of the first shock reflection, we show the results of method (b) calculations in Fig. 12 at $t = 0.8$, which is a time shortly after the first reflection, and at $t = 0.9$, which is a time shortly after second reflection, or the first reflection from the piston.

The Saltzman problem is insufficient to demonstrate the wider utility of the method because of the absence of intrinsic vorticity. To complete the demonstration we need a test problem in which the flow field generates nonzero vorticity.

5.2. A Shock Refraction Problem

It is well known that a vortex sheet is generated by the interaction of a shock wave with an inclined interface which possesses a density discontinuity [13]. Because such interactions may be very complex, it is convenient to work with the so-called "regular" refraction in which all waves are shock waves, since this case may be easily studied using shock-polar analysis. The problem is illustrated in Fig. 13.

For computational convenience we have chosen an interface between two ideal gases, both with $\gamma = 1.4$. The nominal conditions specifying our test problem are: incident shock Mach number (laboratory frame) $M_s = 2$, interface density ratio $\rho_2/\rho_1 = 1.5$, and shock-interface angle of incidence $= 60°$. The resulting shock-polar solution is given in Table I.

The initial mesh is composed of two adjacent regions, each initially containing gases of different densities but equal pressures. Region 1 is a 36×30 mesh, with the left boundary vertical and the right boundary slanted at 60° to represent the interface. Region 2 is a 40×30 mesh uniformly slanted at 60°. The upper and lower boundaries are reflective, and the left boundary is a piston, which moves to the right with a velocity of 1.48 units, driving a Mach 2 shock into region 1. Note, however, that shortly after the incident shock reaches the interface a weak reflected shock is formed which modifies the conditions in front of the piston, making the incident shock unsteady, so that the Mach number is only nominally equal to 2.

The shock is allowed to propagate for a time $t = 1.3$, and the results showing the lagrangian mesh, the velocity vectors, and the density and pressure contours are given in Figs. 14 and 15 for the standard CAVEAT method and for method (b), respectively. The improvement is obvious from a qualitative examination of the mesh. The mesh in Fig. 14 is noticeably distorted, particularly in the vicinity of the

TABLE I

Nominal Test Problem Conditions (Steady-State Frame)

Region	1	2	3	4	5
Angle (deg.)	60	131.02	21.90	90.96	56.12
Density	1	1.5	4.29	2.93	2.67
Pressure	1	1	5.15	5.15	4.50
Mach number	2.31	2.83	1.49	0.94	1.06
Velocity (magn.)	2.73	2.73	1.93	1.48	1.63

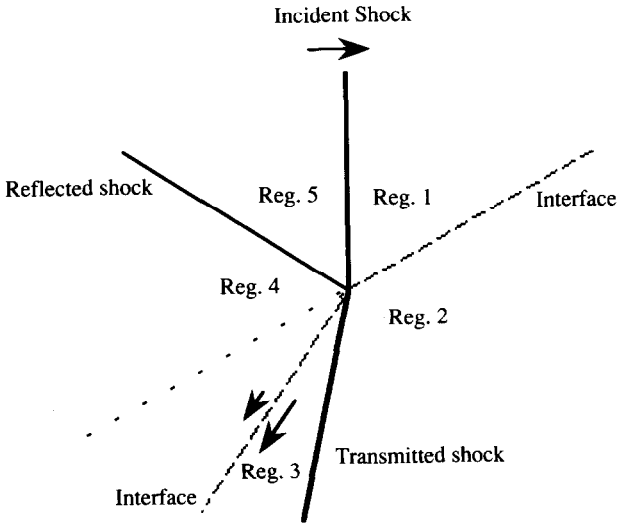


FIG. 13. The shock refraction test problem.

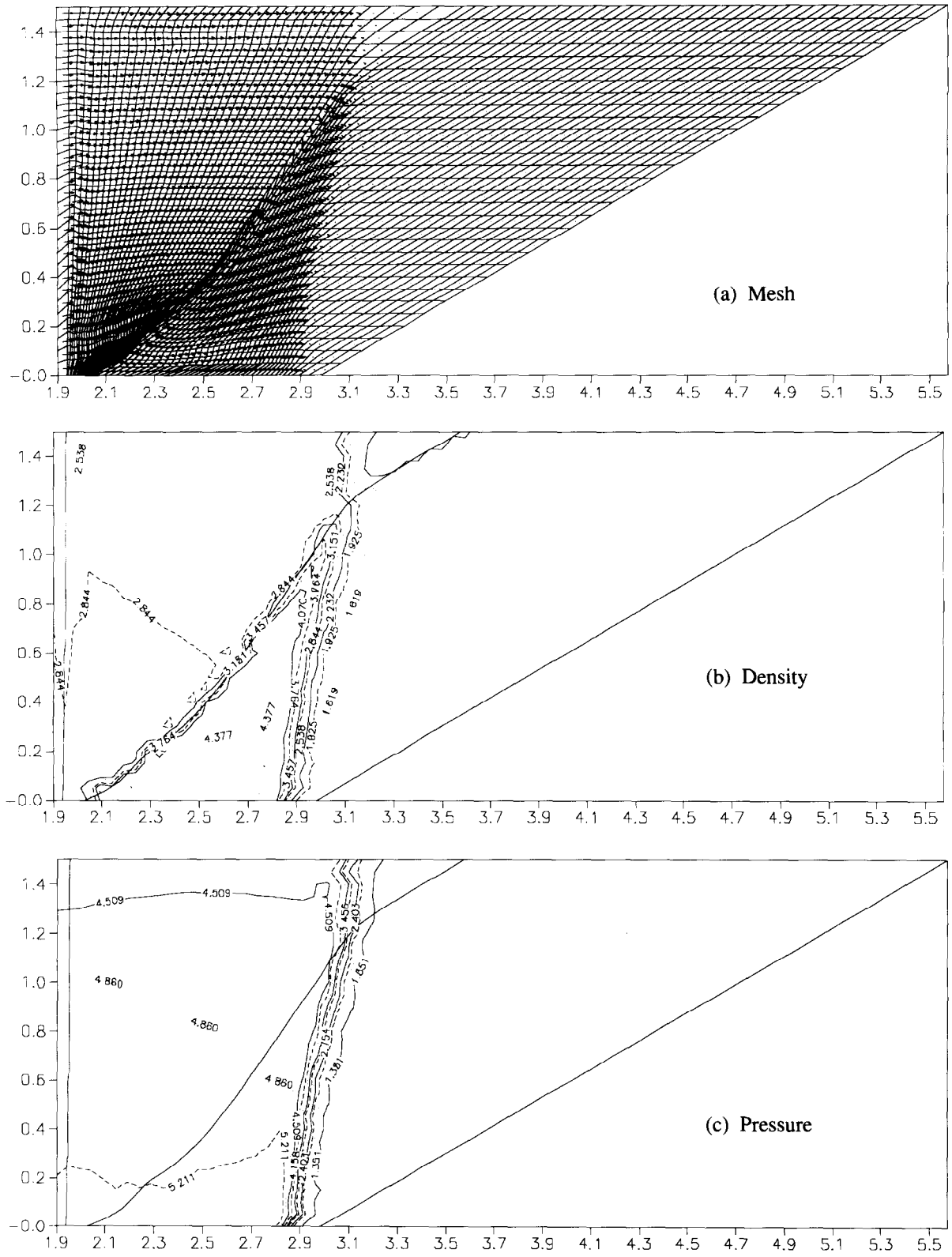


FIG. 14. Lagrangian mesh, velocity vectors, density, and pressure contours for the shock refraction problem, obtained with the standard CAVEAT code, at time $t = 1.3$.

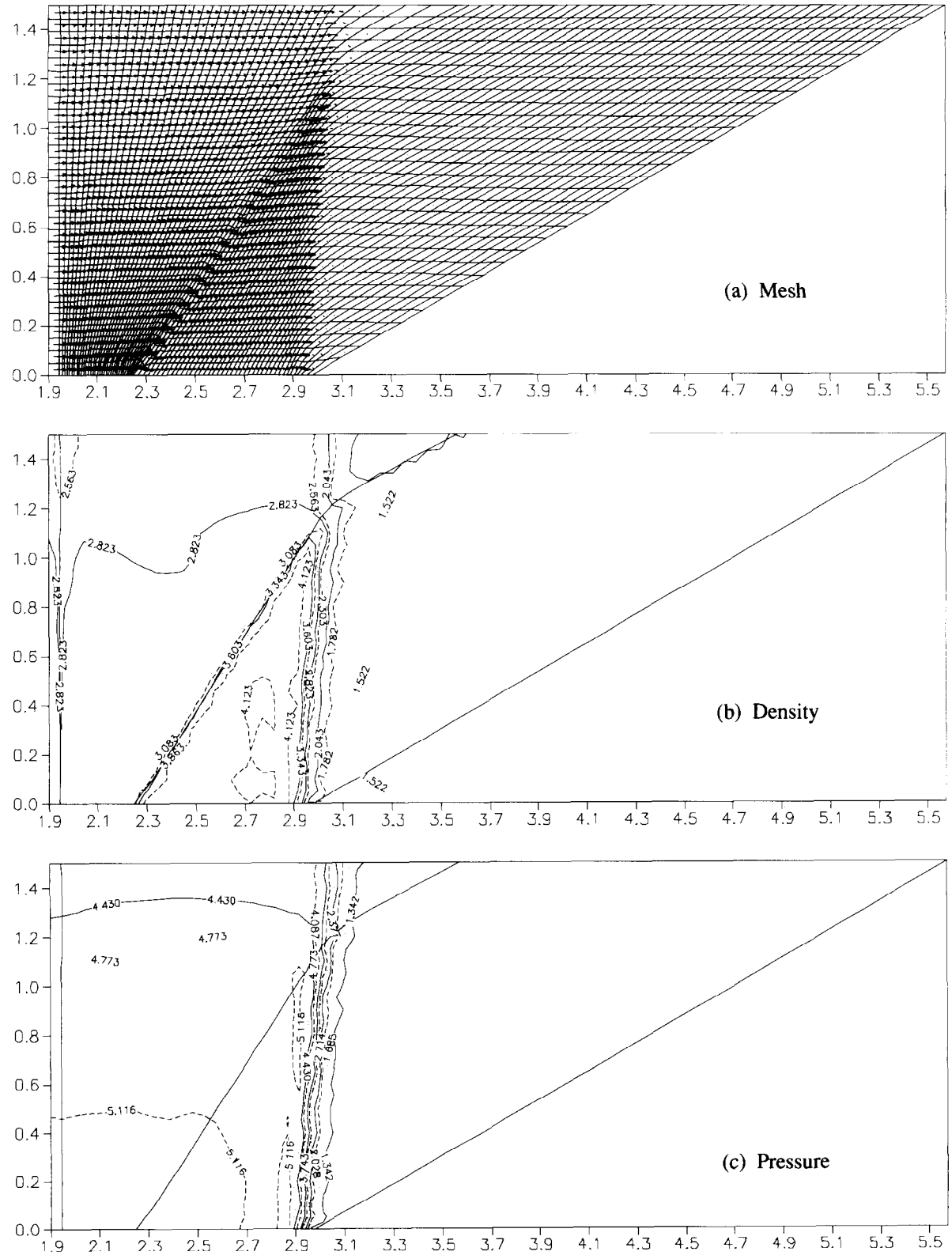


FIG. 15. Lagrangian mesh, velocity vectors, density, and pressure contours for the shock refraction problem, obtained using method (b) applied to the CAVEAT code, at time $t = 1.3$.

TABLE II

A Comparison of Computed versus Estimated Results for Regions 3 and 4

	Region 3		Region 4	
	Computed	Estimated	Computed	Estimated
Standard CAVEAT ($M_s = 1.987$)				
Pressure	4.99	5.06	5.02	5.06
Density	4.24	4.25	2.85	2.91
Horiz. velocity	1.22	1.30	1.49	1.54
Vert. velocity	−0.24	−0.26	0.13	0.12
Method (b) ($M_s = 1.959$)				
Pressure	4.99	4.89	5.09	4.89
Density	4.17	4.18	2.91	2.85
Horiz. velocity	1.25	1.26	1.45	1.51
Vert. velocity	−0.17	−0.25	0.15	0.11

lower part of the interface where the mesh is on the verge of tangling. Both computations show the shear in the horizontal mesh lines at the interface due to the presence of the vortex sheet, although this is masked in the CAVEAT calculation by the much greater distortion. This severe mesh distortion appears to be associated with a pronounced curvature of the interface, in contrast to the rather undistorted shape of the interface in Fig. 15. Although strictly speaking it is not possible to say which is the more correct because of the unknown interaction with the bottom boundary, the pressure contours suggest that the regular or Mach-stem reflection from the bottom boundary is so weak that it is unlikely to bend the interface. It is possible to make a rough quantitative comparison of the two computations, however, based on a shock-polar calculation using an estimated effective incident shock Mach number (the effective Mach number is estimated from the actual pressure ratio across the shock). The results are shown in Table II.

These results do not permit one to distinguish between the two computations, but method (b) is clearly superior from the point of view of rubustness and the quality of the lagrangian mesh.

6. SUMMARY AND DISCUSSION

We have shown that mesh distortion in multidimensional lagrangian codes is at least partly caused by a spurious vorticity, introduced by numerical or discretization errors. On the other hand, we have found that there is an acoustic mechanism which appears to correct the corresponding divergence errors. Based on these findings, we introduced two methods for reconstructing the correct mesh velocity from the computed true or intrinsic vorticity, while assuming that the divergence was correct. Both methods were found to be equally successful in eliminating the extraneous mesh distortion, although the second method, involving a divergence-free velocity correction, was much more efficient. The key to the success of both methods turned out to be an accurate vorticity transport equation in conservation form, in which a particular form of the source term, analogous to the cross product of the gradients of density and internal energy, appears to be essential.

The standard approach to the problem that we have addressed is by the use of artificial viscosity. Many clever and effective artificial viscosities have been developed [2, 3]. Artificial viscosities were originally developed for the purpose of resolving shock waves on discrete meshes, and for this there is ample physical and numerical justification. This type of artificial viscosity should be a bulk viscosity and it therefore should not affect vorticity. There is much less justification when artificial viscosities are used for the purpose of controlling mesh distortion. This type is usually a form of shear viscosity and it damps vorticity. If the mesh distortion is caused by the "null space" problem alluded to in the Introduction, then an artificial viscosity tuned to damp the particular mode in question would be appropriate. However, when the distortion is caused by an extraneous vorticity then the use of artificial vorticity runs the risk of damping both the extraneous and the intrinsic vorticity. On the other hand, artificial viscosity methods have the overwhelming advantage of being simple and inexpensive. We do not view the methods developed here to be competitive with artificial viscosity type methods in this regard, but only as a step towards developing improved artificial-viscosity-like methods in the future. Our primary purpose had been to understand the mechanism behind this common type of lagrangian error and to experiment with methods specifically aimed at correcting it. In this paper we have been interested primarily in demonstrating the principle of the method rather than in its efficiency. Future developments may lead to improvements which would make the method more competitive and therefore more artificial-viscosity-like in the sense of being explicit and, therefore, simpler and cheaper. A substitution of an efficient iterative method for the solution of the Poisson equation would be an immediate improvement. Moreover, one may look forward to further developments which would make the method fully explicit and more practical, such as the use of partial convergence with one or two iterations of an effective iterative method.

ACKNOWLEDGMENTS

We thank Bucky Kashiwa who contributed ideas and helped us try out the method early in its development.

REFERENCES

1. J. Saltzman and P. Colella, LA-UR-85-678, Los Alamos National Laboratory, Los Alamos, NM, 1985 (unpublished).
2. M. L. Wilkins, *J. Comput. Phys.* **36**, 281 (1980).
3. L. G. Margolin, UCRL-53882, Lawrence Livermore National Laboratory, Livermore, CA, 1988 (unpublished).
4. P. J. O'Rourke, *J. Comput. Phys.* **53**, 359 (1984).
5. 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, LA-10613-MS, Los Alamos National Laboratory, Los Alamos, NM, 1986 (unpublished).
6. R. Abraham and J. E. Marsden, *Foundations of Mechanics*, 2nd ed. (Benjamin–Cummings, Reading, MA, 1978), p. 154.
7. P. M. Morse and H. Feshbach, *Methods of Theoretical Physics* (McGraw–Hill, New York, 1953), Part. 1, p. 52.
8. G. Arfken, *Mathematical Methods for Physicists*, 2nd ed. (Academic Press, New York, 1970), p. 66.
9. J. U. Brackbill, *Comput. Phys. Commun.* **47**, 1 (1987).
10. M. J. Fritts and J. P. Boris, *J. Comput. Phys.* **31**, 173 (1979).
11. R. Aris, *Vectors, Tensors, and the Basic Equations of Fluid Mechanics* (Prentice-Hall, Englewood Cliffs, NJ, 1962), p. 63.
12. J. J. Dongarra, C. B. Moler, J. R. Bunch, and G. W. Stewart, *LINPACK Users' Guide* (SIAM, Philadelphia, 1979).
13. L. F. Henderson, *J. Fluid Mech.* **26**, 607 (1966).