JOURNAL OF COMPUTATIONAL PHYSICS **135,** 203–216 (1997)
ARTICLE NO. CP975702

An Arbitrary Lagrangian–Eulerian Computing Method for All Flow Speeds*

C. W. Hirt, A. A. Amsden, and J. L. Cook†

University of California, Los Alamos Scientific Laboratory, Los Alamos, New Mexico 87544

Received November 10, 1972

A new numerical technique is presented that has many advantages for obtaining solutions to a wide variety of time-dependent multidimensional fluid dynamics problems. The method uses a finite difference mesh with vertices that may be moved with the fluid (Lagrangian), be held fixed (Eulerian), or be moved in any other prescribed manner, as in the Arbitrary Lagrangian–Eulerian (ALE) technique. In addition, it employs an implicit formulation similar to that of the Implicit Continuous-fluid Eulerian (ICE) technique, making it applicable to flows at all speeds.

This paper describes the basic methodology, presents finite difference approximations, and discusses such matters as stability, accuracy, and zoning. In addition, illustrations are included from a number of representative calculations.

1. INTRODUCTION

There have been many finite difference techniques devised for the solution of fluid dynamic problems. As catalogued in [1], nearly all of these techniques can be classified as falling into one of two basic categories, depending on whether they are written primarily in terms of Lagrangian or Eulerian coordinates. Within each of these categories it is further possible to distinguish between those techniques applicable to high speed flows and those applicable to low speed. Further subdivisions are mostly matters of individual taste, prejudice, or specialization for specific applications.

In this paper a technique is presented for the solution of the Navier–Stokes equations that is both Lagrangian and Eulerian, and that is applicable to flows at all speeds. The method uses a finite difference mesh with vertices that may move with the fluid (Lagrangian), be held fixed (Eulerian), or be moved in any other prescribed way. Because of this flexibility the method is referred to as an Arbitrary Lagrangian–Eulerian (ALE) technique [2]. A scheme of this nature has previously been reported by Trulio [3] for compressible flow problems. This new technique, however, may be applied to flows at any speed, since it has an implicit formulation similar to that in the Implicit Continuous-fluid Eulerian (ICE) method [4]. In particular, in the limit of infinite sound speed, the difference equations reduce to a generalization of the Marker-And-Cell (MAC) equations for the incompressible Navier–Stokes equations [5].

The advantages of the ICED-ALE method include its ability to resolve arbitrary confining boundaries, to have variable zoning for purposes of obtaining optimum resolution, to be almost Lagrangian for improved accuracy in problems where fully Lagrangian calculations are not possible, and to operate with time steps many times larger than possible with explicit methods.

The basic ICED-ALE method has been separated into three distinct parts called phases. This separation is described in Section II. Finite difference approximations are discussed in Section III for Cartesian and cylindrical coordinates. Also in Sections II and III is an interpretation of the ICE methodology, which leads to an estimate for the number of iterations necessary to solve the implicit difference equations. In Section IV some discussion is directed toward matters of stability, accuracy, choice of mesh, etc. This section is illustrated with a number of representative calculations.

An attempt has been made to concisely summarize a considerable amount of material in this paper. However, for the reader interested in a more complete description of the difference equations, a flow chart, and a complete FORTRAN computer listing for a code based on the method described in this paper, Ref. [6] is available upon request.

II. BASIC METHODOLOGY

The finite difference mesh used here consists of a network of quadrilateral cells with vertices labeled by integer

Reprinted from Vol. 14, Number 3, March 1974, pages 227–253.

* This work was performed under the joint auspices of the United States Atomic Energy Commission and the Defense Nuclear Agency (DNA Subtask HC-061, DNA Work Unit No. 15—Calculations at Low Altitude, and under Contract DNA001-72-C-0106, NWED Subtask Code HC-061, Work Unit No. 50.)

† Now with Science Applications, Incorporated, P.O. Box 2351, La Jolla, California 92037.

0021-9991/97 $25.00

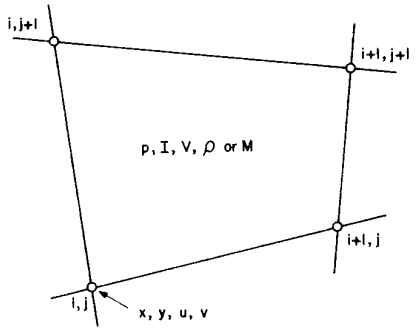

FIG. 1. The assignment of the variables about a cell.

pairs (i, j), denoting column i and row j. Fluid variables are assigned to staggered locations in the mesh as shown in Fig. 1. Pressures (p), specific internal energies (I), cell volumes (V), and densities (ρ) or masses (M) are all assigned to cell centers. Coordinates (x, y) and velocity components (u, v) are assigned to cell vertices.

The differential equations to be solved are

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{u} = 0 \tag{1}$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot \rho \mathbf{u}\mathbf{u} = -\nabla p + \rho \mathbf{g} \tag{2}$$

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot \rho E \mathbf{u} = -\nabla \cdot p\mathbf{u} + \rho \mathbf{g} \cdot \mathbf{u}, \tag{3}$$

where $E = \frac{1}{2}\mathbf{u} \cdot \mathbf{u} + I$ and I is the material specific internal energy. In Eqs. (2) and (3) $\mathbf{g}$ is a body acceleration (usually gravity) and p is the fluid pressure given by the equation of state

$$p = f(\rho, I). \tag{4}$$

For problems involving shock waves it is necessary to add to p an artificial viscous pressure, q. A suitable form for q that is linear in the velocity divergence is

$$q = -\lambda \rho \nabla \cdot \mathbf{u}. \tag{5}$$

Usually q is replaced by zero in expanding cells, that is, where $\nabla \cdot \mathbf{u}$ is positive. In Ref. [6] details are given for including a complete viscous stress, but in this paper these complications are omitted in order to simplify the presentation of the essential ideas of the ICED-ALE method.

The conservation statements of mass, momentum, and energy contained in Eqs. (1)–(3) are more convenient for our purposes when integrated over a volume V, which may be moving with an arbitrarily prescribed velocity. Denoting the surface of V by S and the outward normal on S by $\mathbf{n}$ these equations are (see, for example, Ref. [7])

$$\frac{d}{dt}\int_V \rho \, dV - \int_S \rho(\mathbf{U} - \mathbf{u}) \cdot \mathbf{n} \, dS = 0 \tag{6a}$$

$$\frac{d}{dt}\int_V \rho \mathbf{u} \, dV - \int_S \rho \mathbf{u}(\mathbf{U} - \mathbf{u}) \cdot \mathbf{n} \, dS + \int_V \nabla p \, dV - \int_V \rho \mathbf{g} \, dV = 0 \tag{6b}$$

$$\frac{d}{dt}\int_V \rho E \, dV - \int_S \rho E(\mathbf{U} - \mathbf{u}) \cdot \mathbf{n} \, dS + \int_S \nabla p \mathbf{u} \cdot \mathbf{n} \, dS - \int_V \rho \mathbf{g} \cdot \mathbf{u} \, dV = 0. \tag{6c}$$

In these expressions $\mathbf{U}$ is the velocity of the surface S. When $\mathbf{U} = 0$ the equations are Eulerian, and when $\mathbf{U} = \mathbf{u}$ the equations are Lagrangian. The pressure gradient term in Eq. (6b) could be written as a surface integral, but for cylindrical coordinates a simpler finite difference approximation is obtained directly from the volume integral and this is advantageous for the implicit formulation of the difference equations.

The finite difference formulae presented in Section III are written as approximations to these equations in which the integration volumes are the cells of a moving finite difference mesh. In particular, the V in Eqs. (6a) and (6c) is the volume of a cell in the mesh, and the V in Eq. (6b) is a volume surrounding a vertex. A typical cross section for the latter is indicated by the dashed line in Fig. 2. This difference in integration volumes is dictated by having defined fluid densities and energies at cell centers while velocities are defined at cell vertices. In Section III discrete approximations are described for all the terms in Eqs. (6a)–(6c).

The calculations necessary to advance a solution one step in time, δt, are separated into three distinct phases. The first phase consists of an explicit Lagrangian calcula-

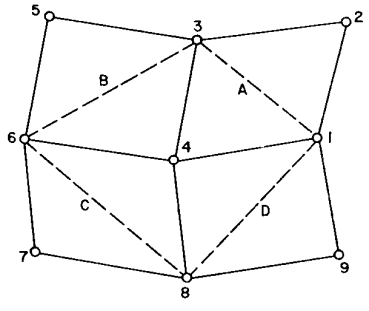

FIG. 2. The dashed line encloses the momentum integration volume used for vertex 4. The notation is that used for typical vertex and cell finite difference equations appearing in the text.

tion, except mesh vertices are not moved. Second, an iteration phase adjusts the pressure gradient forces to the advanced time level. This phase, which is optional, eliminates the usual Courant-like numerical stability condition that limits sound waves to travel no further than one cell per time step. The mesh vertices are moved to their new Lagrangian positions after this phase. Finally, in the third phase, which is also optional, the mesh can be moved to a new configuration. In this (rezone) phase convective fluxes must be computed to account for the movement of fluid between cells as the mesh moves. The calculations in this last phase are automatically iterated if zones try to move too far in any single step, so that gross rezoning can be accomplished without introducing numerical instabilities.

This separation of a calculational cycle into a Lagrangian phase and a convective flux, or rezone, phase originated in the Particle-in-Cell numerical method [8], and has since been used in many hydrodynamic computer codes. In the present technique the different phases can be combined in various ways to suit the requirements of individual problems. For example, in high speed problems, in which the Courant stability condition is not likely to be violated, an explicit calculation is acceptable and the phase two iteration may be omitted, and for an explicit Lagrangian calculation only phase one is used.

Phases one and three are variations of familiar Lagrangian and Eulerian finite difference techniques, although there are some novel features as described in Section III. The phase two iteration, however, is new and requires some preliminary discussion. The purpose of phase two is to get time-advanced pressure forces in the Lagrangian part of a calculation. The reason for this can be appreciated from the following argument. In an explicit method pressure forces can be transmitted only one cell each time step, that is, cells exert pressure forces only on neighboring cells. When the time step is chosen so large that sound waves should travel more than one cell the one cell limitation is clearly inaccurate and a catastrophic instability develops. The instability arises because the explicit pressure gradients lead to excessive cell compressions or expansions when multiplied by too large a time step. This then leads to larger pressure gradients the next cycle, which try to reverse the previous excesses, but since the time step is too large the reversal is also too large and the process repeats itself with a rapidly increasing amplitude. The overresponse to pressure gradients in this fashion is eliminated by using time-advanced pressure gradients, for then cells cannot compress or expand to the point where the gradients are reversed.

Unfortunately, the time-advanced pressures depend on the accelerations and velocities computed from those pressures, so an iterative solution of the equations is necessary. Physically, an iteration offers a means by which pressure signals can traverse across more than one cell in a time step. The iteration is, however, more efficient than a straight explicit calculation with reduced time step, because pressure variations are propagated only to the point where they are producing effects no longer considered significant. This point is discussed in more detail in Sections III and IV.

III. DISCRETE APPROXIMATIONS

A. *The Finite Difference Equations*

A set of finite difference approximations is outlined in this section for two-dimensional Cartesian (x, y) or cylindrical (r, z) coordinates. When Cartesian coordinates are desired, all radii, r, appearing in the following equations should be replaced by unity. When using cylindrical coordinates the equations refer to unit azimuthal angle, not 2π.

Input data to start a calculation consists of mesh vertex coordinates (x, y), velocities (u, v), cell densities (ρ), and internal energies (I).

The variables adjusted at each stage of a calculational cycle are indicated by the entries in Table I. For example, vertex velocities are adjusted explicitly in the first part of phase one, again in the implicit second phase, and finally in phase three, if the mesh is rezoned. In the remainder of this subsection the calculations performed at each cycle, and listed as entries in Table I, are expressed for a typical mesh vertex, labeled 4 in Fig. 2, or for a typical cell, labeled A in Fig. 2. All formulae presented here will employ number or letter subscripts for vertex or cell quantities, respectively, as shown in Fig. 2. In subsection B are described the considerations necessary to impose a variety of boundary conditions.

1. *Initializing Calculations.* For convenience and speed it is desirable to compute and store several auxiliary quantities used repeatedly in the following equations. These quantities include cell volumes, cell total energies, and the masses assigned to vertices. Once established at the start of a calculation, the auxiliary quantities are automatically updated in the course of a calculation cycle.

The volume, V, of a cell such as A in Fig. 2 is

$$\begin{aligned} V_A = {} & \tfrac{1}{6}(r_1 + r_2 + r_3)[x_1(y_2 - y_3) + x_2(y_3 - y_1) \\ & + x_3(y_1 - y_2)] + \tfrac{1}{6}(r_1 + r_3 + r_4)[x_1(y_3 - y_4) \\ & + x_3(y_4 - y_1) + x_4(y_1 - y_3)], \end{aligned} \tag{7}$$

where $r_i = x_i$ for cylindrical coordinates and $r_i = 1$ for plane coordinates, $i = 1, 2, 3$, or 4.

The mass contained in a cell can be obtained from the product of cell volume and density,

$$M_A = \rho_A V_A. \tag{8}$$

TABLE I

	Cycle Steps	Variables Updated								
		x	y	u	v	p	E	M	I	ρ
	(1) Initializing Calculations					p	E	M		
Lagrangian	(2) Phase I–First Part			$\tilde{u}$	$\tilde{v}$		$\tilde{E}$			
Lagrangian	(3) Phase II–Implicit			u^L	v^L	p^L				
Lagrangian	(4) Phase I–Second Part	x^L	y^L				E^L			ρ^L
Rezone	(5) Phase III–Rezone	x^{n+1}	y^{n+1}	u^{n+1}	v^{n+1}		E^{n+1}	M^{n+1}		ρ^{n+1}
	(6) Auxiliary					p^{n+1}			I^{n+1}	

For advancing velocities it is necessary to assign a mass to each vertex. In this technique it is assumed that the mass in each cell is equally shared between its four corner vertices, so that vertex 4 in Fig. 2, for example, is given the mass

$$M_4 = \tfrac{1}{4}(M_A + M_B + M_C + M_D). \tag{9}$$

To ensure energy conservation the total specific energy, E, is directly advanced in time rather than the internal energy I. The relation between them for the typical cell A in Fig. 2 is

$$E_A = I_A + \tfrac{1}{8}(u_1^2 + u_2^2 + u_3^2 + u_4^2 + v_1^2 + v_2^2 + v_3^2 + v_4^2). \tag{10}$$

At the beginning of a calculation, E is computed from the input values of I, u, and v. Thereafter, this relation is used to recover I from E for use in the equation of state pressure, Eq. (4).

2. *Phase One, First Part.* In this step velocities are advanced explicitly in time using pressure gradients and body forces computed from the currently available pressures and mesh coordinates. If viscous, elastic, or other stresses are desired, they may be included at this stage as well (see Ref. [6]). The total energy of each cell is also advanced in time to account for the work done by the body forces and other stresses, except those of pressure. Pressure work terms are included only after the implicit pressure calculation in phase two. This delay permits time-advanced pressures to be used in computing the work and ensures consistency with the velocities coming out of phase two.

The difference equations used to advance the velocity components of vertex 4 in Fig. 2 are

$$\tilde{u}_4 = u_4 + \frac{\delta t}{2M_4}\{r_4[p_A(y_1 - y_3) + p_B(y_3 - y_6) + p_c(y_6 - y_8) + p_D(y_8 - y_1)]\} + \delta t g_x, \tag{11a}$$

where the tilde over u_4 on the left side signifies the temporary new value of u_4, and similarly

$$\tilde{v}_4 = v_4 + \frac{\delta t}{4M_4}\{p_A(r_1 + r_3)(x_3 - x_1) + p_B(r_3 + r_6)(x_6 - x_3) + p_C(r_6 + r_8)(x_8 - x_6) + p_D(r_8 + r_1)(x_1 - x_8)]\} + \delta t g_y. \tag{11b}$$

These expressions were obtained from the integral form of the equations of motion, Eqs. (6). A mass of $2M_4$ has been assumed to lie within the integration volume, because it includes approximately 1/2 the mass in each cell surrounding vertex 4, while M_4 contains only 1/4 of each surrounding cell mass. It would be possible to use the actual mass contained in the integration volume, but it is not necessarily true that this would produce more accurate results. For example, in an incompressible flow the integration volume may not be constant even though the individual volumes of the cells are constant, and therefore the vertex mass computed before coordinates are moved would not be the same as that computed afterward. In any case, the prescription used here has been successfully used by many other investigators; see, for example, Ref. [9].

For the initial time advancement of energy E, an auxiliary quantity, Q, is computed for each vertex. This quantity represents the work done on fluid in the integration volume

by all stresses, except pressure. In the present case, again referring to Fig. 2,

$$\begin{aligned} Q_4 = \frac{\delta t}{8M_4}\{&q_A[(u_1+u_3)(y_1-y_3)\\ &+(v_1+v_3)(x_3-x_1)](r_1+r_3)\\ &+q_B[(u_3+u_6)(y_3-y_6)\\ &+(v_3+v_6)(x_6-x_3)](r_3+r_6)\\ &+q_C[(u_6+u_8)(y_6-y_8)\\ &+(v_6+v_8)(x_8-x_6)](r_6+r_8)\\ &+q_D[(u_8+u_1)(y_8-y_1)\\ &+(v_8+v_1)(x_1-x_8)](r_8+r_1)\}\\ &+\delta t[g_x u_4+g_y v_4]. \end{aligned} \tag{12}$$

After all the vertex Q values have been computed, the total specific energy for each cell is adjusted to

$$\tilde{E}_A = E_A + \tfrac{1}{4}(Q_1+Q_2+Q_3+Q_4). \tag{13}$$

In other words, the total energy change at each vertex is assigned to each neighboring cell in proportion to the mass that each cell contributed to the vertex. Total energy is conserved in this process.

3. *Phase Two, Implicit.* When an explicit calculation is wanted, this step can be omitted. The object of phase two is to obtain new velocities that have been accelerated with time-advanced pressure gradients. Since the time-advanced pressures depend on the densities and energies obtained when vertices are moved with their new velocities, which in turn are functions of the new pressures, these pressures are defined implicitly and must be determined by iteration. The implicit problem can be formulated as follows: Let a superscript L denote time-advanced values and a superscript n denote values at the beginning of a cycle. The desired pressure, p_A^L, of cell A will be the solution of the equation

$$p_A^L - f(\rho_A^L, I_A^L) = 0, \tag{14}$$

where the new cell density and energy can be approximated in terms of their initial values as

$$\begin{aligned} \rho_A^L &= \rho_A^n \frac{V_A}{V^*}\\ I_A^L &= I_A^n + \frac{p_A^n}{\rho_A^n}\left(1-\frac{V^*}{V_A}\right), \end{aligned} \tag{15}$$

where V_A is the current volume of cell A and V^* is the volume the cell would have if its vertices were moved according to, e.g.,

$$x_4^* = x_4^n + u_4\,\delta t, \quad y_4^* = y_4 + v_4\,\delta t, \quad \text{etc.} \tag{16}$$

It is important that V^* be computed in terms of coordinates shifted with velocities accelerated with the new pressures through formulae like Eqs. (11), in which p^n is replaced by p^L.

A solution for p_A^L can be obtained by applying a Newton–Raphson iteration to Eq. (14) considered as an implicit equation for p_A^L through Eqs. (11), (15), and (16). The velocities $(\tilde{u}, \tilde{v})$ obtained in the previous step are used as initial guesses for the iteration. The iteration proceeds by sweeping through the mesh and applying the following adjustments to each cell, once each sweep:

(a) Compute V^* using the most updated values for (u, v).

(b) Compute new guesses for ρ_A^L and I_A^L from Eqs. (15).

(c) Compute a pressure change, δp_A, according to

$$\delta p_A = -\frac{p_A^L - f(\rho_A^L, I_A^L)}{S_A} \tag{17}$$

where the most updated value is used for p_A^L on the right side and S_A is a relaxation factor to be described.

(d) Adjust the current guess for cell pressure, p_A^L, by adding δp_A to it, and adjust the velocities at the corners of the cell to reflect this pressure change:

$$\begin{aligned} u_1 &\to u_1 + \frac{\delta t}{2M_1} r_1 (y_2 - y_4)\,\delta p_A\\ v_1 &\to v_1 + \frac{\delta t}{4M_1} (r_2 + r_4)(x_4 - x_2)\,\delta p_A\\ u_2 &\to u_2 + \frac{\delta t}{2M_2} r_2 (y_3 - y_1)\,\delta p_A\\ v_2 &\to v_2 + \frac{\delta t}{4M_2} (r_1 + r_3)(x_1 - x_3)\,\delta p_A\\ u_3 &\to u_3 + \frac{\delta t}{2M_3} r_3 (y_4 - y_2)\,\delta p_A\\ v_3 &\to v_3 + \frac{\delta t}{4M_3} (r_2 + r_4)(x_2 - x_4)\,\delta p_A\\ u_4 &\to u_4 + \frac{\delta t}{2M_4} r_4 (y_1 - y_3)\,\delta p_A\\ v_4 &\to v_4 + \frac{\delta t}{4M_4} (r_3 + r_1)(x_3 - x_1)\,\delta p_A. \end{aligned} \tag{18}$$

The mesh is repeatedly swept and calculations (a–d) are performed once for each cell each sweep, until no cell exhibits a pressure change violating the inequality

$$\left|\frac{\delta p}{p_{\max}}\right| < \varepsilon, \tag{19}$$

where $p_{\max}$ is the actual or an estimated maximum pressure in the mesh and ε is a suitably chosen small number. Typically, ε is of order 10^{-3}.

The relaxation number S_A, used in Eq. (17) for δp_A, must be chosen to keep the pressure changes in bound and progressing in the right direction, but its exact value is not crucial. In the ordinary Newton–Raphson procedure S_A is the derivative of the function whose root is sought with respect to the iteration variable. That is, S_A is the rate at which the quantity $p_A - f(\rho_A, I_A)$ changes as the variable p_A changes. This rate must be computed using the implicit relations given by Eqs. (11), (15), and (16). The difference equations can be manipulated [6] to yield an algebraic expression for S_A, but it is easier to compute the rate of change numerically. For this purpose a small pressure change, δp_T, is chosen, which is usually of order $\varepsilon p_{\max}$. Then the velocity *changes* induced by this pressure change in a cell are used to compute the volume change and corresponding density and energy changes according to Eqs. (15). Finally, S_A for the cell is set equal to the quotient of the difference between $p - f(\rho, I)$, evaluated after and before the change in pressure, and δp_T. These S_A values are computed and stored for each cell before the phase two iteration is started. It is unnecessary to update them during an iteration.

For a given ε, the number of iterations necessary to obtain convergence can be roughly estimated through the following argument. Assume N iterations are necessary for convergence. Since the iteration acts something like an explicit calculation, its effective time step must be $\delta t/N$. In a flow with Mach number M, pressure variations satisfy the approximate inequality

$$\left|\frac{\delta p}{p}\right| \lesssim M^2. \tag{20}$$

Therefore, the iteration has an effective Mach number, $M^2 \approx \varepsilon$, and hence an effective sound speed, C_{eff}, such that

$$C_{\text{eff}}^2 = \frac{u^2}{\varepsilon}, \tag{21}$$

where u is a typical fluid speed. Since the Courant number for an explicit calculation must be less than unity for stability, this suggests the iteration will be stable only if its effective Courant number is less than unity,

$$\frac{C_{\text{eff}}}{\delta x}\left(\frac{\delta t}{N}\right) \lesssim 1, \tag{22}$$

where δx is a typical cell dimension. If it is assumed that the iteration has been designed to proceed at maximum speed, corresponding to an equality in the above expression, then the number of iterations necessary for convergence will be of order

$$N \approx \left(\frac{u\,\delta t}{\delta x}\right)\frac{1}{\varepsilon^{1/2}}. \tag{23}$$

According to this, the iteration number increases as either δt increases or ε decreases, but is independent of the actual material sound speed. Thus, once a tolerable level of pressure error, ε, has been chosen, the implicit scheme converges in a finite number of iterations, regardless of the actual material sound speed. It is this feature of the ICE method [4] that makes it superior to an ordinary explicit method whose time step must be continually reduced as the sound speed is increased.

4. *Phase-Two, Second Part.* The final values of u^L, v^L, and p^L from the iteration in the previous step are the new Lagrangian values for the cycle. To complete the Lagrangian portion of a cycle, the cell energies must now be adjusted for the pressure work terms omitted in step 2, and vertices must be moved with the fluid to their new positions.

The energy for cell A in Fig. 2 is changed according to

$$\begin{aligned} E_A^L = \tilde{E}_A + \frac{\delta t}{4M_A}\{ & p_{12}(r_1 + r_2)[(u_1 + u_2)(y_1 - y_2) \\ & + (v_1 + v_2)(x_2 - x_1)] \\ & + p_{23}(r_2 + r_3)[(u_2 + u_3)(y_2 - y_3) \\ & + (v_2 + v_3)(x_3 - x_2)] \\ & + p_{34}(r_3 + r_4)[(u_3 + u_4)(y_3 - y_4) \\ & + (v_3 + v_4)(x_4 - x_3)] \\ & + p_{41}(r_4 + r_1)[(u_4 + u_1)(y_4 - y_1) \\ & + (v_4 + v_1)(x_1 - x_4)]\}, \end{aligned} \tag{24}$$

where the velocities and pressures are the final values of u^L, v^L, and p^L. Cell-edge pressures are required; p_{34}, for

example, is the pressure along the left edge of cell A between vertices 3 and 4. For this boundary pressure a mass weighting scheme is used,

$$p_{34} = \frac{M_B p_A^L + M_A p_B^L}{M_A + M_B}, \tag{25}$$

and similarly for the other edges of cell A. The mass weighted average (25) was recommended by Fromm [10] and has led to good results in a variety of test cases.

Mesh vertices are moved with the fluid to their new locations,

$$\begin{aligned} x_4^L &= x_4^n + \delta t\, u_4^L \\ y_4^L &= y_4^n + \delta t\, v_4^L. \end{aligned} \tag{26}$$

The new velocities are used to move vertices since this makes the explicit Lagrangian portion of a cycle second-order accurate in time. The implicit calculation, however, is only first-order accurate.

After the vertices have been moved, new densities, ρ^L, are computed for each cell as the quotient of cell mass divided by the new cell volume.

The phase one and phase two calculations contained in the previous steps comprise an implicit Lagrangian method that is stable for any Courant number, $C\ \delta t/\delta x$, where C is the fluid speed of sound. When the sound speed becomes very much larger than the fluid speed, this method approaches a variant of the incompressible Lagrangian method described in Ref. [11].

5. *Phase Three, Rezone.* As is well known, Lagrangian cell methods are not adequate for describing flows undergoing large distortions. In the present method, the devastating effects of large distortions are eliminated by moving the mesh vertices with respect to the fluid so as to maintain a reasonable mesh structure. Whenever a vertex is moved relative to the fluid, however, there must be an exchange of material among the cells surrounding the vertex. This exchange, which can be interpreted as a convective flux, is expressed by the second terms in Eqs. (6).

Either the convective flux adjustments can be performed for the entire mesh at one time using only values of the fluid variables coming out of the Lagrangian portions of the calculation to compute the new values after rezoning, or each vertex can be separately adjusted, with the values arising from each adjustment used in subsequent calculations for other vertices. The former method requires extra storage for quantities needed both before and after adjusting. The latter method requires no extra storage and has the additional advantage that individual vertices can be rezoned repeatedly if necessary. In fact, a simple scheme has been devised that automatically limits the distance a vertex can move in any one shift and forces as many repeats of the rezone calculations as are necessary for those vertices that exceed the limit. In this way the rezone calculations are always stable, even when a gross change in the mesh configuration is called for.

Both schemes have been used in connection with the ICED-ALE formulation, but only the latter will be described here, while the former is detailed in Ref. [6]. Several prescriptions for choosing vertex rezone velocities are described in Section IV-D. For purposes of this section, they are assumed given.

Before the rezone calculations are started all vertex velocities are converted to momenta and all cell specific energies are converted to total energies so that the rezone calculations will be rigorously conservative of mass, momentum, and energy.

The adjustments associated with a shift in the position of a typical vertex, say 4 in Fig. 2, proceed as follows. First the vertex is moved to its new location,

$$\begin{aligned} x_4^{n+1} &= x_4^L + \delta t\, U_4 \\ y_4^{n+1} &= y_4^L + \delta t\, V_4, \end{aligned} \tag{27}$$

where U_4, V_4 are the rezone velocities specified for the vertex.

When vertex 4 is moved, the lines connecting it to its neighbors 1, 3, 6, and 8 sweep out volumes containing mass and total energy that must be exchanged between the adjacent cells. For example, if vertex 4 moves to the right, the grid line connecting 4 to 3 sweeps out of cell A and adds to cell B a volume equal to

$$\delta V = \frac{\delta t}{3}(2r_4 + r_3)[U_4(y_3 - y_4) + V_4(x_4 - x_3)]. \tag{28}$$

Associated with this volume exchange there will also be a mass and total energy exchange between the cells. The mass or energy per unit volume assigned to this volume can be computed in various ways. It is well known that use of a simple average of the quantities on either side of the line leads to a computational instability, but that a stable calculation can be obtained by weighting the average in favor of the value in the cell from which the quantity is subtracted. This is the upstream or donor cell convective flux approximation. Thus, the mass subtracted from cell A and added to cell B is

$$\delta M = \frac{1}{2}(\delta V + \alpha\,|\,\delta V\,|)\frac{M_A}{V_A} + \frac{1}{2}(\delta V - \alpha\,|\,\delta V\,|)\frac{M_B}{V_B}, \tag{29}$$

where α is the donor cell weighting factor. When $\alpha = 0$ the flux is centered and when $\alpha = 1$ the flux is full donor cell. The best choice for α is discussed in Section IV-B.

The corresponding total energy subtracted from cell A and added to cell B is

$$\delta(ME) = \frac{1}{2}(\delta V + \alpha|\delta V|)\frac{M_A E_A}{V_A} + \frac{1}{2}(\delta V - \alpha|\delta V|)\frac{M_B E_B}{V_B}. \tag{30}$$

Similar formulae are used for the exchanges of mass and energy between the other pairs of cells surrounding the vertex.

A shift in vertex 4 is also accompanied by a momentum exchange between vertices 1, 3, 6, and 8, because 4 is a corner of the control volumes for these vertices. For example, when vertex 4 is moved, the surface connecting vertices 4 and 2 sweeps out a volume,

$$\delta V = \frac{\delta t}{3}(2r_4 + r_2)[U_4(y_2 - y_4) + V_4(x_4 - x_2)]. \tag{31}$$

The mass in this volume is $(M_A/V_A)\ \delta V$ and the u-momentum it contains is approximated as

$$\delta(Mu) = \frac{1}{2}\frac{M_A}{V_A}[(\delta V - \alpha|\delta V|)\,u_3 + (\delta V + \alpha|\delta V|)\,u_1]. \tag{32}$$

This momentum change must be subtracted from vertex 1 and added to vertex 3. Similar exchanges are computed for the vertex pairs (3, 6), (6, 8), and (8, 1). The v-momentum is handled in the same way, with v replacing u in the above formula. The scalar α is again chosen as zero for a centered momentum flux and as unity for an upstream or donor cell flux.

When these exchanges among the cells and vertices surrounding the moved vertex have been completed, new volumes are computed for the cells (A, B, C, and D) and the mesh is ready to have any other, or even the same, vertex moved to a new location.

After all vertices have been moved to the positions desired, the total cell masses and energies are converted back to densities and specific energies, and vertex momenta are converted back to velocities.

6. *Auxiliary Calculations.* New specific internal energies, I, are computed from E by subtracting the average cell kinetic energy according to Eq. (10). Finally, new cell pressures may be computed from the defining equation of state in terms of the new values of ρ and I.

B. *Boundary Conditions*

Many kinds of boundary conditions are possible. In this section are given the prescriptions for rigid boundaries, inflow and outflow boundaries, and free boundaries. Also, rigid boundaries may be classified as free-slip or no-slip and may be given prescribed motions. Combinations of these conditions can be used to simulate a great variety of problem situations.

In nearly all cases, the setting of boundary conditions is accomplished by making adjustments to the velocities of the boundary vertices. These adjustments must be performed before and after the phase-one calculations and after each iteration in phase two.

Consider a vertex located on the top or bottom boundary of the mesh, with coordinates (x_c, y_c). Coordinates of the vertex to the left will be denoted by (x_L, y_L) and those to the right by (x_R, y_R). For a vertex on the left or right sides of the mesh, the following discussion will apply provided "below" is read for "left" and "above" is read for "right."

The simplest boundary condition to impose is for a rigid no-slip wall on which the fluid velocity is set equal to the prescribed wall velocity.

A rigid free-slip boundary is more difficult to handle, since it is only the fluid velocity normal to the boundary that is constrained. If the normal direction to the boundary at vertex (x_c, y_c) is defined as the direction normal to the line connecting (x_L, y_L) with (x_R, y_R), then the correct boundary condition is achieved by replacing the velocity at the vertex by

$$\begin{aligned} u_c' &= u_n \sin\theta + u_c \cos^2\theta + v_c \sin\theta\cos\theta \\ v_c' &= -u_n \cos\theta + u_c \cos\theta \sin\theta + v_c \sin^2\theta, \end{aligned} \tag{33}$$

where u_n is the prescribed boundary velocity in the normal direction positive when directed to the right of the vector pointing from (x_L, y_L) to (x_R, y_R). The angle θ is determined from

$$\cos\theta = \frac{x_R - x_L}{[(x_R - x_L)^2 + (y_R - y_L)^2]^{1/2}}.$$

This transformation leaves the tangential fluid velocity unchanged while replacing the normal fluid velocity by u_n. If the boundary vertices are to move, as in a Lagrangian calculation, then further refinement of this boundary condition will be needed to keep the vertices on the boundary when it is curved, since Eq. (33) only keeps a vertex on the local tangent to the boundary.

Prescribed inflow and outflow boundaries are imposed by setting fluid velocities at the boundary vertices to the desired values.

At a free surface the tangential and normal stresses are zero, and no special conditions are required in this case. However, since free surface vertices receive accelerations from only one side, some caution must be exercised when the tangential accelerations vary significantly in a direction

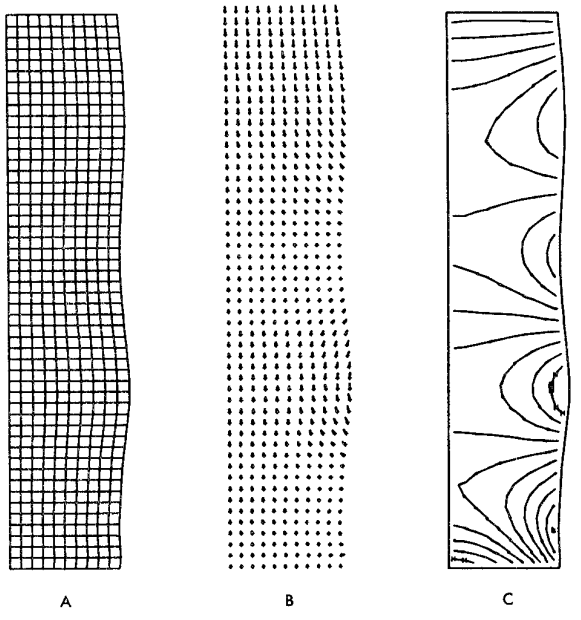

FIG. 3. Calculation simulating the pulsating flow of fluid pumped through an elastic tube. Plots shown are the computing grid, velocity vectors, and isobars after 275 calculational cycles, or at $t = 2.641$.

normal to the free surface. The one-sided calculations can then be a poor approximation, and it may be necessary to extrapolate the acceleration from within the fluid out to the free surface.

Continuative outflow boundaries are always troublesome for low speed flows, since influences from these boundaries can be felt upstream. The goal of a continuative boundary is to permit outflow with a minimum of upstream disturbance. A prescription that has worked in ALE is to set the velocities of the boundary vertices equal to the velocities located at vertices immediately inside the boundary. This replacement should be made before and after phase one, but *not* after each iteration in phase two. During phase two the continuative boundary velocities are permitted to adjust to whatever pressure changes occur during the iteration. The use of this prescription in a low speed application, however, must be carefully checked in each case to be sure it is not causing unwanted upstream influences.

For phase-three rezoning of ρ and E, values of these quantities must be specified in cells outside the boundary if flow is to take place across the boundary.

A calculational example that illustrates the use of several kinds of boundary conditions is shown in Fig. 3. This figure illustrates the result of a calculation of fluid pumped periodically through an elastic tube. The bottom edge of the mesh is an inflow boundary with an assigned periodic inflow velocity of the form

$$v = a \sin^2 \omega t,$$

with $a = 0.5$ and $\omega = 0.8\pi$. The tube has unit radius and the segment computed is initially 5 units long. The time step was 0.01. The top of the mesh is a continuative outflow boundary, while the left edge is an axis of cylindrical symmetry (i.e., a rigid free-slip wall). The right edge of the mesh is a free surface, except that an additional force is imposed on these vertices to represent the stress that would be generated in an elastic confining membrane. The fluid is assumed to be incompressible with density 1.0.

Three kinds of data are included in Fig. 3. In Fig. 3a is shown the computing grid after 275 calculational cycles. Bulges of successive pressure pulses are evident along the outer tube boundary. In Figs. 3b and 3c are the corresponding velocity vector and pressure contour plots. A region of high pressure is located under each radial bulge and a low pressure under each depression.

In this example the mesh was continuously rezoned to keep the radial grid lines fixed, while the axial grid lines were adjusted to be equally spaced along each radial line.

No detailed comparisons with theoretical or experimental data have been attempted with this problem, since it is only presented here as a qualitative example. Some detailed calculations illustrating the accuracy of the ICED-ALE technique are presented in the next section.

IV. GENERAL REMARKS

To make the ICED-ALE technique presented in this paper a useful tool, it is necessary to give consideration to such matters as computational stability, accuracy, prescriptions for rezoning, automatic timestep control, marker particle techniques, etc. These topics are discussed in the following paragraphs.

A. *Test Calculations*

A useful problem to test a compressible flow code is the shock tube, in which a long straight cylinder is divided into two compartments by a central diaphragm. On one side of the diaphragm there is a gas, say, of density $\rho = 0.2$ and internal energy $I = 0.18$, while on the other side the gas has density $\rho = 0.1$ at the same energy. A calculation begins by removing the diaphragm with the gases at rest. We assume γ-law gases with $\gamma = 5/3$. The pressure difference in the two gases drives a shock into the less dense gas, while a rarefaction moves into the denser gas. A contact surface trails behind the shock.

Both a Lagrangian and an Eulerian calculation have been made for this problem, using 60 zones of size $\delta z = 0.333$ and time step $\delta t = 0.1$ (nondimensional units are used throughout). The artificial viscosity coefficient used in both cases was $\lambda = 0.04$. In Figs. 4a and 4b, the density and velocity profiles are shown at $t = 10.0$ for each calculation in comparison with the theoretical predictions. These

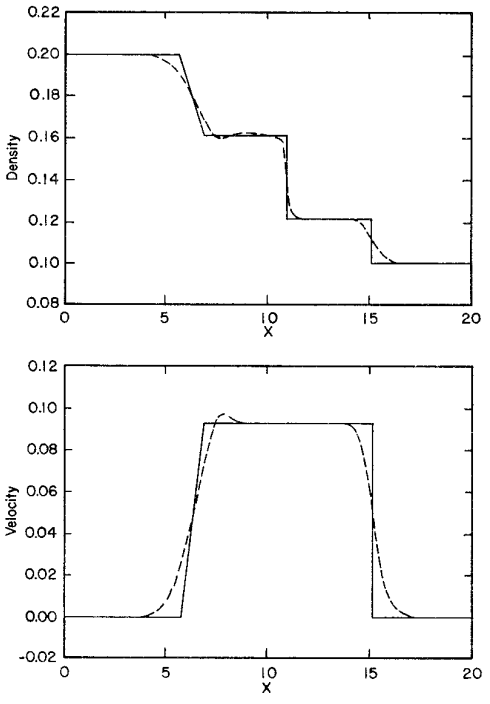

FIG. 4a. Density and velocity profiles for a Lagrangian calculation of the shock tube problem at $t = 10.0$. The solid line is the theoretical prediction.

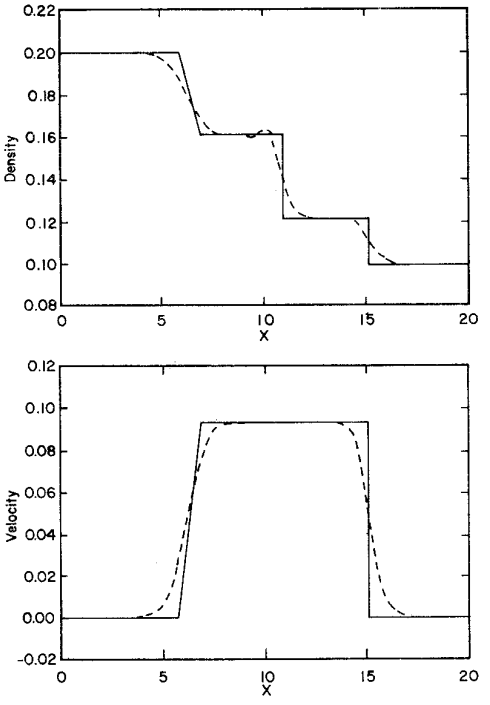

FIG. 4b. Density and velocity profiles for a Eulerian calculation of the shock tube problem at $t = 10.0$. The solid line is the theoretical prediction.

results are typical for standard finite difference calculations of similar problems.

A more significant test of the ICED-ALE method is illustrated in Fig. 5 where the velocity profiles are shown for three calculations of the same problem in which the time step was successively $\delta t = 0.1$, 1.333, and 3.333. In all cases the calculations are stable. For the largest time step only three cycles are needed to reach $t = 10.0$, and the shock has moved approximately 15 zones in this time.

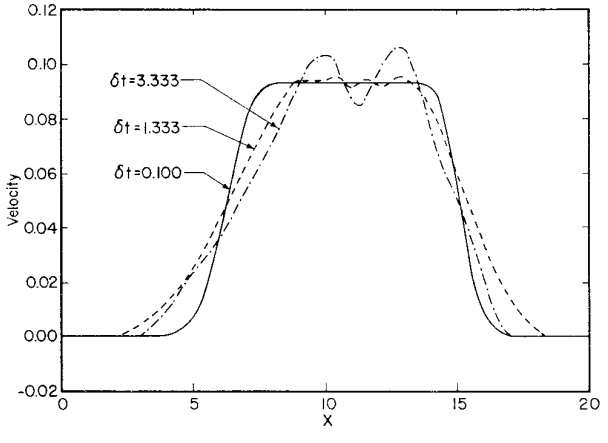

FIG. 5. Comparison of shock tube velocity profiles obtained using three different time increments. The profiles obtained with $\delta t = 0.1$ and $\delta t = 3.333$ are shown at $t = 10.0$, the profile obtained with $\delta t = 1.333$ is shown at $t = 10.66$.

Some accuracy has been lost at the largest δt, but only in exceptional cases would one choose a time step that allowed the shock to move five cells each cycle. In general, the time step should be chosen to give reasonable resolution for the time scales of interest. If the shock is of primary interest, a time step would be used in which the shock traverses no more than one cell each step. By contrast, in an incompressible or very low speed flow the time step should be chosen to have fluid particles moving one cell every few cycles, but in this limit compression waves should travel across many cells each time step.

A simple example of an incompressible fluid calculation can be obtained by repeating the shock tube calculation described above using gases with a very large sound speed and with gravity added and directed along the confining cylinder axis. A hydrostatic pressure should be established in the tube when its ends are closed, to balance the gravitational acceleration, and the fluid should remain at rest. This is indeed the case as can be seen in Fig. 6 where the pressure obtained after 145 iterations is plotted as a function of height, x. The equation of state was chosen in this case to be $p = a^2(\rho - \rho_0(x))$ where the sound speed is $a = 10^5$ and $\rho_0(x)$ is the initial density distribution. The time step was $\delta t = 0.01$, the gravity acceleration was 3.0, and the convergence criterion was $\varepsilon = 10^{-4}$.

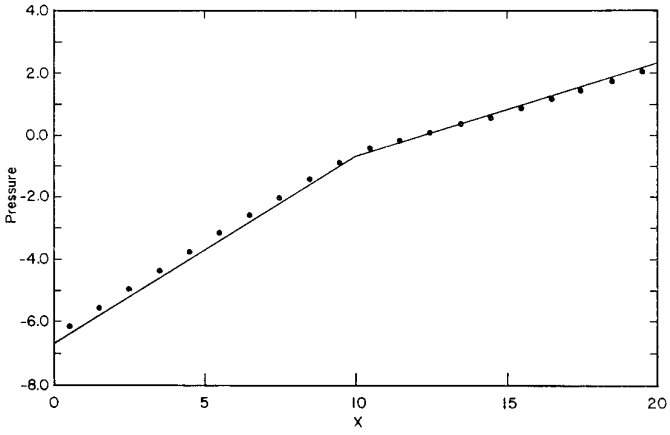

FIG. 6. Hydrostatic pressure profile after 145 iterations in the first cycle (dots) compared with the theoretical profile (solid line). After three cycles the calculated results lie on the solid line.

These examples show that the ICED-ALE method is stable for arbitrary Courant numbers and is accurate for both high speed and low speed problems, although some accuracy is lost in processes with time scales not well resolved by the chosen time step.

B. *Computational Stability*

A rigorous stability analysis cannot be performed for the ICED-ALE technique presented here, but good estimates can be made based on analogies with simpler schemes for linear equations with constant cell sizes [12].

When phase one is followed by phase two there is no stability restriction on the distance a sound wave may propagate in a time step. However, when viscous effects are included in the phase one calculations, as in Ref. [6], for example, there is a stability condition that limits the distance over which momentum can diffuse in one time step to be less than one cell width. Violation of this condition results in a rapidly growing and oscillating instability. A good *estimate* for the restriction on the time step, δt, needed in this case is

$$\delta t < \left[\frac{2(2\mu + \lambda)}{\rho}\left(\frac{1}{\delta x^2} + \frac{1}{\delta y^2}\right)\right]^{-1}, \tag{34}$$

where μ and λ are the first and second coefficients of viscosity and where δx and δy are the effective cell sizes defined as

$$\delta x = \tfrac{1}{2}(x_1 + x_2 - x_3 - x_4),$$
$$\delta y = \tfrac{1}{2}(y_2 + y_3 - y_4 - y_1).$$

The most important stability considerations are associated with phase three. It is well known that forward time and centered space differencing ($\alpha = 0$) for the convection in phase three is unstable. Stability can be achieved by increasing the magnitude of α. However, to prevent unnecessary numerical smoothing the magnitude of α should be kept as small as possible. An optimal choice might be developed along the lines of an idea by Boris [13], but this has not yet been done. As a rule of thumb, α should not be less than $\delta V/V$ where δV is either given by Eq. (28) and V is the average volume of cells on either side of the mass flux boundary, or δV is given by Eq. (31) and V is the volume of the cell containing the momentum flux boundary.

In addition to the choice of α there is a more fundamental stability and accuracy requirement inherent in phase three. Material cannot be fluxed through more than one cell in one time step, because the flux approximations have been based on the implicit assumption of exchanges only between neighboring cells or vertices. Thus, the flux volume to cell volume ratio, $\delta V/V$, must never be allowed to exceed unity. Since the δV in Eqs. (28) or (31) is proportional to δt, this limitation is really a limitation on the time step.

In practice only one of the forms, Eqs. (28) or (31), is needed for limiting δt, and coupled with Eq. (34) these restrictions can be used to automatically control the time step in a program. In Ref. [6], the automatic control of δt is coupled with the option of an automatic determination of the viscosity coefficients, λ and μ, to optimize stability and efficiency.

C. *Coupling Alternate Mesh Vertices*

Accelerations computed at a vertex (i, j) with the difference equations presented in Section III are independent of the position of the vertex within the integration region outlined in Fig. 2. Intuitively it is expected that the most accurate results will be obtained when the vertex is located at the center of the integration region, a condition that can often be arranged with proper rezoning of the mesh. However, this insensitivity to the location of the central vertex is symptomatic of a common problem in finite difference methods in which the shortest resolvable wave lengths $(2\delta x)$ are not sufficiently damped.

An example of the problems that may arise is contained in the velocity vector plot shown in Fig. 7a. The problem consists of an incompressible fluid flow directed from bottom to top with unit speed entering the bottom of the mesh and passing around a rectangular block. The computing mesh was treated as Eulerian with square cells throughout. Vectors are drawn from each mesh vertex. Rigid free-slip conditions were imposed on the boundaries of the block. The region of difficulty extends off the trailing edge

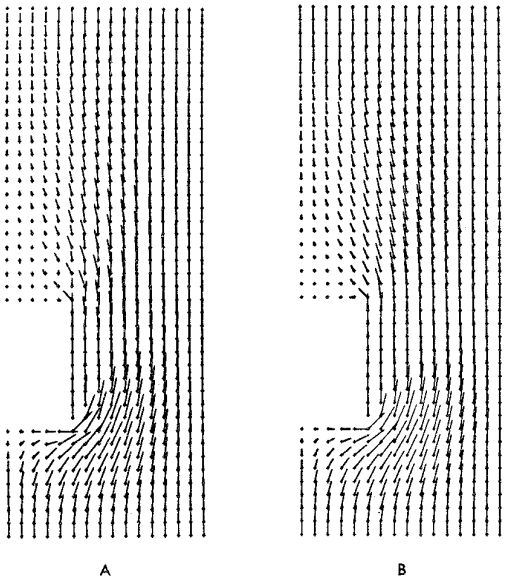

FIG. 7. Velocity vectors for flow around a rectangular block. These plots, taken at the same time in two different calculations, show the appearance before and after coupling of the alternate mesh vertices.

(top) of the block where the velocity vectors are seen to alternate direction on adjacent vertices.

An effective means of eliminating this undesirable feature was developed for an early version of the ALE method [14]. The idea is to introduce a small restoring force on each vertex to keep it more in line with neighboring vertices. For vertex 4 in Fig. 2 the restoring acceleration is

$$\frac{1}{a_{nc}\,\delta t}\left[\frac{1}{4}(\mathbf{u}_1 + \mathbf{u}_3 + \mathbf{u}_6 + \mathbf{u}_8) - \mathbf{u}_4\right]. \tag{35}$$

This acceleration is treated as arising from a body force and is added to **g** for purposes of computing the work done in Eq. (12). The coefficient a_{nc} implies that this relaxation in the velocity field has a characteristic time of a_{nc} time steps. Note, however, that if $a_{nc} = 1$, the technique becomes identical to a procedure introduced by Lax many years ago [15]. To avoid the difficulty of that procedure as $\delta t \to 0$, it would be better to define $a_{nc} = a'_{nc}/\delta t$, in which a'_{nc} is the actual relaxation time, rather than the number of cycles for relaxation.

The effect of using expression (35) can be seen in Fig. 7b, where the alternating velocities have been removed. This calculation, with the vertex coupler, agrees very well with a calculation of the same problem using the Marker-and-Cell method [5].

D. *Zoning and Rezoning*

Many choices are available for the construction of suitable meshes. For example, a variety of options have been previously described for the calculation of fluid sloshing in a rectangular tank [2]. In Ref. [16] a simple technique is presented for the automatic construction of grids that follow curved boundaries, have increased resolution in selected regions, etc.

When considering a problem involving more than one material, no special techniques are needed if interfaces between the different materials coincide with mesh lines. Of course these interface lines must be moved with the fluid in Lagrangian fashion, which means that severe distortions of interfaces cannot be allowed. Unfortunately this imposes a limitation on some multimaterial applications.

The kind of difficulties that may be encountered are illustrated in Fig. 8. In this example, a heavy, incompressible fluid of nondimensional density 2 was above a lighter fluid of unit density. The calculational mesh initially consisted of rectangular cells having equal masses. A unit gravitational acceleration was directed downward, producing an unstable situation. A half cosine velocity perturbation was applied to the interface at the start of the calculation. The fluid configuration (mesh) is shown at times 0.085 in Fig. 8a and 0.123 in Fig. 8b. Arrows along the side of the mesh mark the interface intersections. The mesh on either side of the interface was continuously rezoned to be approximately orthogonal, but no rezoning prescription is likely to be found that will permit the calculation to proceed significantly further than shown in Fig. 8b. Permitting slip tangentially along the interface would help, but even with this the rolling up of the interface becomes progressively harder to define with the initial rectangular

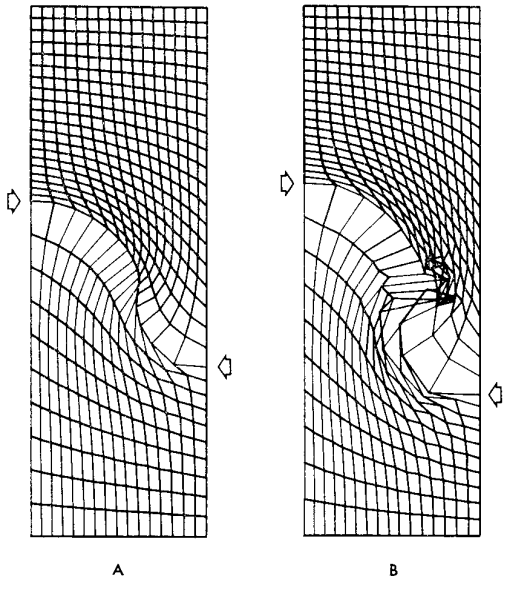

FIG. 8. Calculation of an interface instability flow. Arrows indicate the Lagrangian interface. Even with continuous rezoning of the interior vertices, severe grid distortion will soon terminate the calculation.

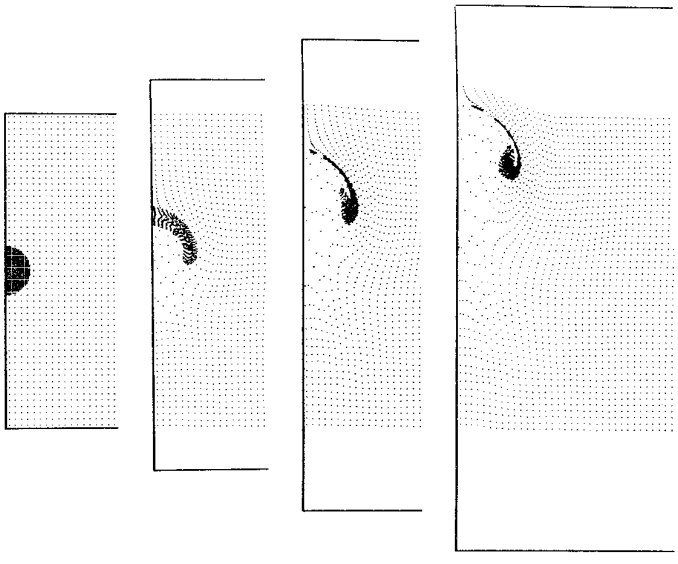

FIG. 9. An ICED-ALE calculation of an intense explosion in the atmosphere, showing the marker particle configuration at 0.5, 10.0, 20.0, and 30.0 s. The expansion of the mesh is indicated by the growing frame, and the rise of the hot bubble through the ambient atmosphere is evident in the relative particle motion.

array of cells. Nevertheless, this calculation agrees remarkably well, up to the time shown in the last figure, with a calculation performed by Daly using a two-fluid Marker-and-Cell method [5].

E. *Incompressible Flow*

For an incompressible flow it is not desirable to compute the pressure for phase one from the equation of state Eq. (4). If a calculation is attempted with the sound speed significantly larger than $u/\varepsilon^{1/2}$, where u is a typical fluid speed, then small density variations possibly remaining after the phase-three rezoning may be magnified into large pressure variations when used in the equation of state for the start of the next cycle. These pressures can then generate velocity fluctuations that may be impossible to eliminate in the following phase-two iteration. The problem may be avoided by omitting the equation of state calculation in phase one and using instead the pressures remaining from the previous phase-two iteration as a first guess for the next cycle. In practice, of course, the iteration can be started with any reasonable guess for the initial pressure, but the better the first guess the sooner convergence is obtained. Thus, the equation of state calculation is omitted before phase one when Mach numbers are less than approximately $\varepsilon^{1/2}$, but retained otherwise.

F. *Marker Particles*

In some problems it is convenient to use Lagrangian marker particles to aid in the visualization of complicated fluid motions. Figure 9, for example, shows a sequence of marker particle configurations obtained in the course of a calculation of an intense explosion in the atmosphere. The left edge of the computation region is an axis of cylindrical symmetry. Initially particles were densely, but uniformly, distributed in a semicircular region about the center of the explosion. Additional particles with a much smaller particle density were placed in the remainder of the computing region. As the problem proceeded, the mesh was continuously enlarged to approximately double its original size, leaving a region without particles around the outer edges of the mesh. The particle configurations show the subsequent collapse of the initial hot bubble and the formation of a buoyant vortex ring. Although it is not evident in Fig. 9, the mesh was continuously rezoned to translate upward with the hot material. Many calculations of this type have been performed and have been shown to yield more accurate results than are obtainable with standard Eulerian techniques [17].

Marker particles are moved with the local fluid velocity each time step. In previous particle techniques [5], the local velocity at each particle is computed as a linear interpolation in both x and y directions among the nearest four vertex velocities. In the present method, however, the usual interpolation technique is difficult to apply directly, because the mesh consists of arbitrary quadrilateral cells. With arbitrary cells it is even difficult to determine which cell a particle is located in. To overcome these problems, an auxiliary rectangular mesh of uniform zones is superimposed over the general ALE mesh. Each cycle, the auxiliary mesh is assigned a velocity field linearly interpolated from the general mesh. Particles can then be moved in the usual way [5] with respect to the auxiliary mesh. To interpolate from the ALE to the rectangular mesh, a sweep is made through the vertices of the ALE mesh. For each vertex, its location in the rectangular mesh is determined and then its momentum and mass are distributed linearly to the four nearest rectangular mesh vertices. When all ALE vertices have been swept, the total momentum accumulated at each rectangular vertex is divided by the total mass accumulated there, resulting in a velocity field that can be used to move the particles. Boundary conditions must be set in the auxiliary mesh as appropriate.

ACKNOWLEDGMENTS

The authors are pleased to acknowledge contributions from several members of Group T-3 at the Los Alamos Scientific Laboratory, and in particular thank J. U. Brackbill, T. D. Butler, R. A. Gentry, F. H. Harlow, and H. M. Ruppel.

REFERENCES

1. F. H. Harlow, *Numerical Methods for Fluid Dynamics, an Annotated Bibliography,* Report LA-4281, Los Alamos Scientific Laboratory, Los Alamos, NM, 1969.

2. C. W. Hirt, *Proceedings of the Second International Conference on Numerical Methods in Fluid Dynamics, Berkeley, 1970.*
3. J. G. Trulio, Report AFWL-TR-66-19, Air Force Weapons Laboratory, Kirtland Air Force Base, 1966.
4. F. H. Harlow and A. A. Amsden, *J. Comput. Phys.* **8,** 197 (1971).
5. F. H. Harlow and J. E. Welch, *Phys. Fluids* **8,** 2182 (1965); J. E. Welch, F. H. Harlow, J. P. Shannon, and B. J. Daly, Report LA-3425, Los Alamos Scientific Laboratory, 1966; A. J. Chorin, *Math. Comput.* **22,** 745 (1968).
6. A. A. Amsden and C. W. Hirt, *Yaqui: An Arbitrary Lagrangian–Eulerian Computer Program for Fluid Flows at All Speeds,* Report LA-5100, Los Alamos Scientific Laboratory, 1973.
7. G. K. Batchelor, *An Introduction to Fluid Dynamics,* Cambridge Univ. Press, Cambridge, 1967.
8. F. H. Harlow, Report LAMS-1956, Los Alamos Scientific Laboratory, 1955; A. A. Amsden, Report LA-3466, Los Alamos Scientific Laboratory, 1966.
9. M. L. Wilkins, Report UCRL-7322, Rev. 1, Lawrence Radiation Laboratory, Livermore, CA, 1969.
10. J. E. Fromm, Report LA-2535, Los Alamos Scientific Laboratory (1961).
11. C. W. Hirt, J. L. Cook, and T. D. Butler, *J. Comput. Phys.* **5,** 103 (1970).
12. C. W. Hirt, *J. Comput. Phys.* **2,** 339 (1968).
13. J. P. Boris and D. L. Book, *J. Comput. Phys.* **11,** 38 (1973).
14. T. D. Butler, *Proceedings of the Second International Conference on Numerical Methods in Fluid Dynamics, Berkeley, 1970.*
15. P. D. Lax, *Comm. Pure Appl. Math.* **7,** 159 (1954).
16. A. A. Amsden and C. W. Hirt, *J. Comput. Phys.* **11,** 348 (1973).
17. C. W. Hirt and J. L. Cook, *Proc. Atomic Effects Symposium, April 1973.*