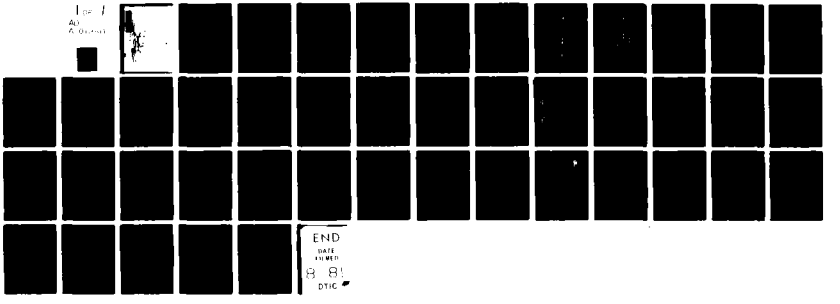
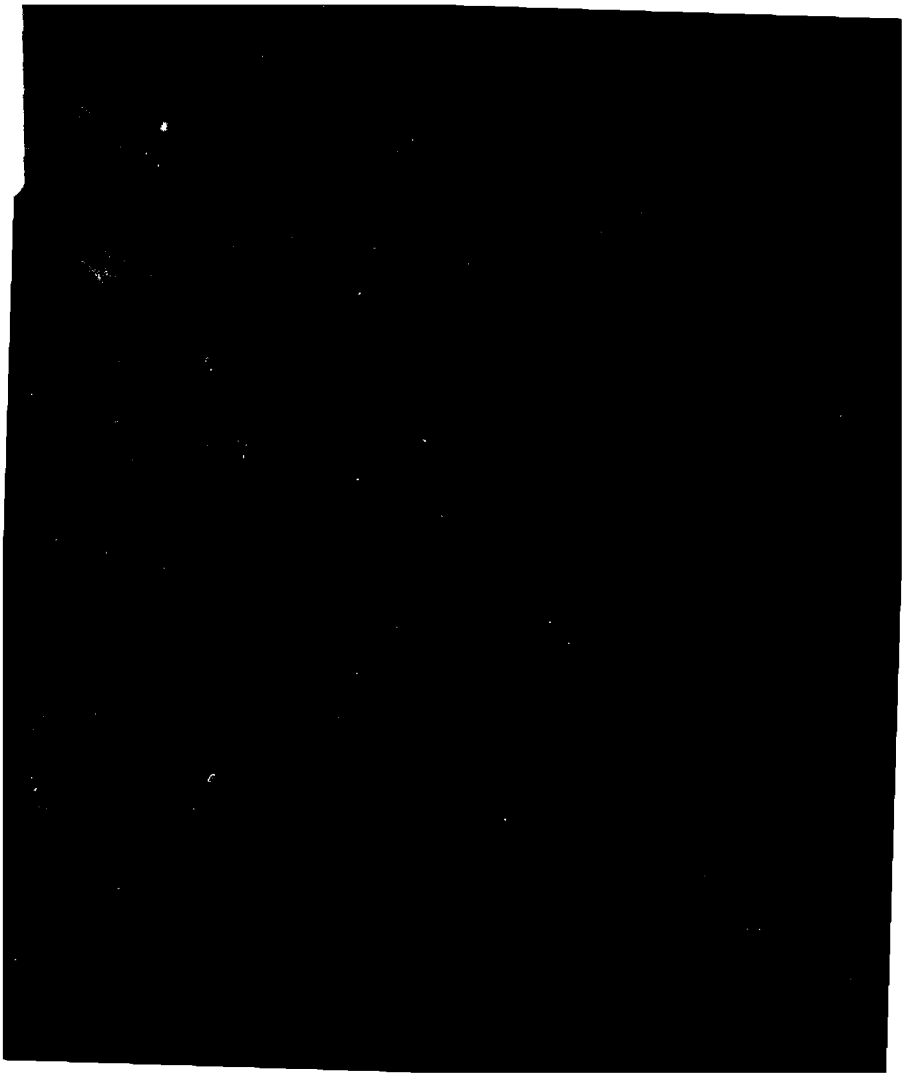
AD-A101 980 NAVAL RESEARCH LAB WASHINGTON DC F/G 21/2
LAGRANGIAN FLUID DYNAMICS FOR COMBUSTION MODELLING.(U)
JUL 81 M J FRITTS, E S ORAN, J P BORIS
UNCLASSIFIED NRL-MR-4570 NL





SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: NRL Memorandum Report 4570 — 2. GOVT ACCESSION NO.: AD-A101 980	3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): LAGRANGIAN FLUID DYNAMICS FOR COMBUSTION MODELLING	5. TYPE OF REPORT & PERIOD COVERED: Interim report on a continuing NRL problem.
	6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): M. J. Fritts, E. S. Oran, and J. P. Boris	8. CONTRACT OR GRANT NUMBER(s):
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Naval Research Laboratory, Washington, DC 20375	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61153N; RR011-09-41; 44-0574-0-1
11. CONTROLLING OFFICE NAME AND ADDRESS: Office of Naval Research, Washington, DC	12. REPORT DATE: July 21, 1981
	13. NUMBER OF PAGES: 44
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Lagrangian hydrodynamics
Combustion
Flame modelling
Triangular gridding

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Recent flow visualization experiments have shown the need to follow the behavior of dynamically interacting coherent structures in both cold flows and flames. Since these structures move with the fluid, a Lagrangian approach is especially useful in theoretical calculations because we can observe the interaction of a particular fluid element with its changing environment as the flow evolves. One-dimensional Lagrangian flame models are successful because they also minimize the effects of numerical diffusion which is the bane

(Continues)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20 ABSTRACT (Continued)

of laminar flame calculations. However, most two- and three-dimensional flame models are Eulerian because of problems in standard Lagrangian formulations for multi-dimensional models and because phenomenological turbulent diffusion terms are usually added which mask the numerical diffusion. The purpose of this paper is to describe one- and multi-dimensional Lagrangian algorithms which eliminate many of the problems previously associated with this approach. An example of a one-dimensional flame calculation which incorporates the new ideas will be given. Finally, examples will be given of the two-dimensional Lagrangian triangular gridding technique and it will be indicated how this may be applied to multi-phase combustion problem.

CONTENTS

LAGRANGIAN FLUID DYNAMICS FOR COMBUSTION MODELLING

I. Introduction

Flow visualization experiments are a powerful way to improve our understanding of transient fluid flows. This is true even though the method provides relatively little quantitative information relative to that which can be obtained from stationary probe experiments. Although very precise probe data can be collected at any particular station in a flow field, it is difficult to correlate that data with that of other stations without some idea of the overall flow patterns linking them. Flow visualization has helped provide a qualitative link by delineating recirculation patterns, boundary layer development, vortex shedding and wake formation. Recent flow visualization experiments (1, 2, 3) have shown the dynamic interaction of coherent structures in shear flows, jets and flames.

The theoretical counterpart to following the evolution of material interfaces and coherent flow structures is the Lagrangian formulation of the equations of motion. In numerical hydrodynamics, the mesh points used to discretize these equations move with the flow, which gives the method several important advantages. Most important is the ability to isolate and follow the interactions of any particular fluid element with its changing environment, a direct analogue of the flow visualization procedure. Since material interfaces move with the flow, numerical simulations of flames and heterogeneous flows, for example, do not need expensive additional computations to track the interfaces. Lagrangian calculations, by definition, have no need to approximate the advective terms in the fluid equations. The numerical diffusion which results from approximating these terms in an Eulerian formulation is therefore eliminated. Similarly, boundary conditions are more easily and accurately specified.

Manuscript submitted May 21, 1981.

However, the advection of the mesh points has also proven to be the greatest obstacle to implementing general Lagrangian calculations. In all but the simplest flows, the mesh soon distorts and becomes tangled, resulting in deteriorating accuracy in the calculation. Reducing this distortion by using any one of a number of Eulerian rezoning techniques reintroduces the problem of numerical diffusion which the Lagrangian method is supposed to minimize. For strong shear flows, for example, the grid may become fixed at some maximum distortion. The rezoning phase of the calculation then accounts for all the transport. The calculation is often less accurate at this point than if it were performed by an Eulerian formulation from the beginning.

Because of this basic grid-distortion problem, Lagrangian methods have not been widely used for complex, transient flows although these are the flows in which we are particularly interested for combustion systems. Numerical analyses and techniques have also lagged in parallel areas, so that general Lagrangian solution methods, error estimates and physical approximations are less well understood than the corresponding Eulerian cases. The first part of this paper discusses mesh distortion and proposes non-diffusive solutions to it. Questions of accuracy and solution procedures are addressed in the second section where calculations are shown which use the algorithms presented in the first section.

II. Distortion of Lagrangian Meshes

A. One-dimensional algorithms

Mesh distortion must lead to decreased accuracy for either the Eulerian or Lagrangian formulations because the location of grid points determines the accuracy of the numerical approximation. Figure 1 illustrates a one-dimensional mesh containing an interface at the i-th grid point. We can expand in both the forward and backward directions about the point i to obtain a Taylor series for the fluid variable f:

$$f_{i+1} = f_i + \frac{\partial f}{\partial x_i}\Delta x + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}\Delta x^2 + \frac{1}{6}\frac{\partial^3 f}{\partial x^3}\Delta x^3 + O(\Delta x^4) \tag{1}$$

and

$$f_{i-1} = f_i - \frac{\partial f}{\partial x_i}\Delta x' + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}\Delta x'^2 - \frac{1}{6}\frac{\partial^3 f}{\partial x^3}\Delta x'^3 + O(\Delta x'^4), \tag{2}$$

where $\Delta x = x_{i+1} - x_i$ and $\Delta x' = x_i - x_{i-1}$.

Subtracting Eq. (2) from Eq. (1) and rearranging gives the centered difference approximation for the derivative of f with respect to x,

$$\frac{\partial f}{\partial x_i} = \frac{f_{i+1}-f_{i-1}}{(\Delta x+\Delta x')} - \frac{1}{2}\frac{\partial^2 f}{\partial x^2}\frac{(\Delta x^2 - \Delta x'^2)}{(\Delta x + \Delta x')} + O[\max(\Delta x^2, \Delta x'^2)]. \tag{3}$$

The local average grid spacing is $\overline{\Delta x} = \frac{1}{2}(\Delta x + \Delta x')$ as shown in Fig. 2. Substituting

$$\Delta x = \overline{\Delta x} + \delta x \tag{4}$$

and

$$\Delta x' = \overline{\Delta x} - \delta x \tag{5}$$

into Eq. (3) yields

$$\frac{\partial f}{\partial x_i} = \frac{f_{i+1}-f_{i-1}}{2\overline{\Delta x}} - \frac{1}{2}\frac{\partial^2 f}{\partial x^2}\frac{(4\overline{\Delta x}\,\delta x)}{2\overline{\Delta x}} + O[\Delta x^2] \tag{6}$$

or

$$\frac{\partial f}{\partial x_i} = \frac{f_{i+1}-f_{i-1}}{2\overline{\Delta x}} + O[\max(\delta x, \Delta x^2)]\ , \tag{7}$$

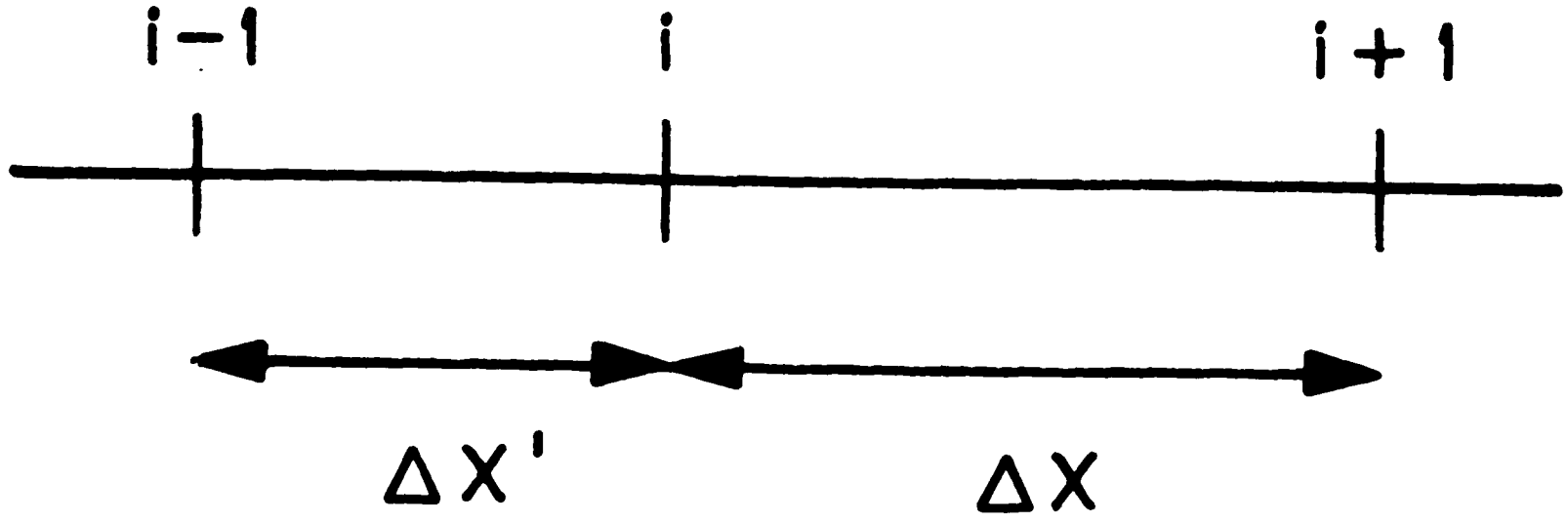


Fig. 1 — One-dimensional mesh with an interface at the ith grid point

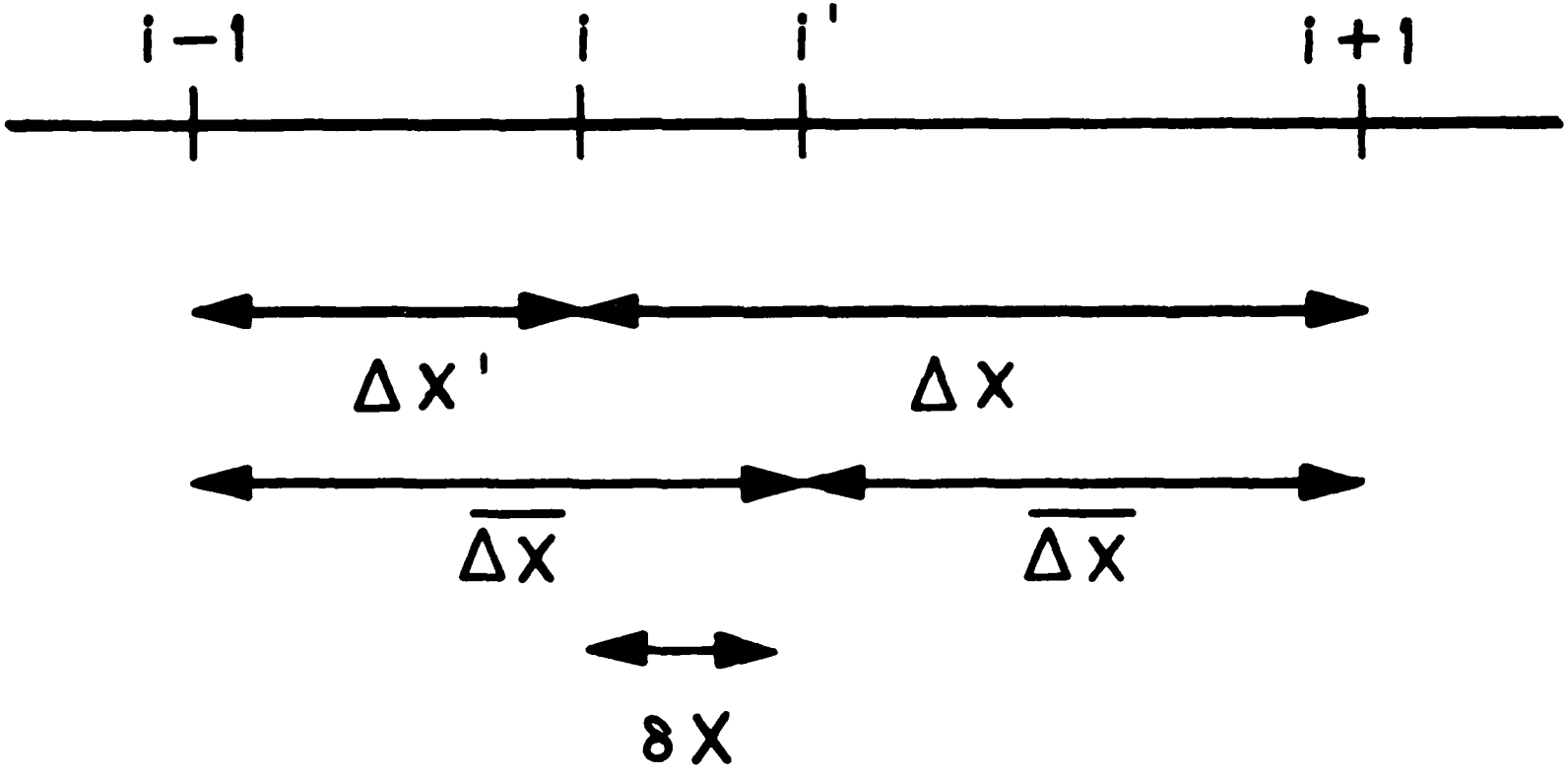


Fig. 2 — One-dimensional mesh showing grid spacings for centered approximations of the derivative of a function of x

since $\Delta x \geq \Delta x'$ by definition. The centered difference approximation is therefore fully second-order accurate only if $\delta x = 0$, $(\Delta x = \Delta x')$ or δx errors are smaller than those associated with Δx^2. Formally, δx must be of the order of Δx^2 to retain accuracy. Of course, for a given physical problem, the constraint in Eq. (7) may be more or less severe depending on the magnitudes of the higher-order derivatives.

The problem in either Eulerian or Lagrangian representations is to derive adaptive gridding techniques which resolve boundaries and interfaces correctly and yet retain the required accuracy. The fundamental differences in the methodology of adaptive gridding in the two representations can be illustrated by considering a hypothetical shock calculation which contains an embedded interface. For an Eulerian calculation both the shock and the interface must either be finely resolved or tracked by some ancillary technique such as marker particles. Otherwise, the numerical diffusion needed to maintain stability will spread the shock over an unacceptably large spatial extent. In a sliding rezone Eularian calculation, these surfaces are kept finely resolved by sliding in zones from nearby regions where enhanced resolution is not required. A transition zone of slowly varying cell size is maintained to accommodate the constraint of Eq. (7). When marker particles are used instead, a high-order scheme must be used to track the surface. In either case, when the shock interacts with the imbedded surface, complicated logic is required to keep zones from migrating across surfaces, to distinguish different marker particles and to introduce new mechanisms to track both the reflected and transmitted shocks.

Conventional Lagrangian codes generally use an Eulerian rezoning phase when the cell size variations are large enough to affect the accuracy of the

approximation. This approach reintroduces the numerical diffusion which the technique sought to eliminate, and therefore will not be discussed further here. Some form of adaptive gridding is still necessary, however, but the resulting procedure is more complicated than in the Eulerian case. Cell interfaces must be inserted and removed to alter the resolution to that desired without losing the non-diffusive advantages of the Lagrangian formulation throughout the bulk of the mesh. In the shock interface problem discussed above, the fully Lagrangian solution would be maintained by injecting mesh points as needed in front of the advancing shock and to remove them in its wake. The interface would require no special treatment other than assuring that the interface point is never deleted.

The Lagrangian rezoning is accomplished by injecting new cell boundaries into the interior of existing cells and by removing unnecessary boundaries between two existing cells (Fig. 3). The number of cells changes during these "split" and "merge" operations, but only local restructuring occurs this way. To maintain the Lagrangian nature of the calculation, material and energy do not flow across cell boundaries, as in global rezone procedures, and no numerical diffusion is introduced at any of the already existing cell boundaries. The regridding criteria in a computer code are programmed so that this process automatically adapts to the resolution needed by the evolving solution. The bookkeeping for this method is complicated, since the number of grid points changes and the location in computer memory of data referring to a given physical point in space also changes as cells are subdivided or removed. The governing physical variable gradients generally determine whether better resolution is required in the vicinity of each cell boundary or whether larger cells would be permissible. This method is currently being used in transient flame simulations (4, 5), and will be discussed in more detail in section III below.

DISCONTINUOUS LAGRANGIAN REZONE

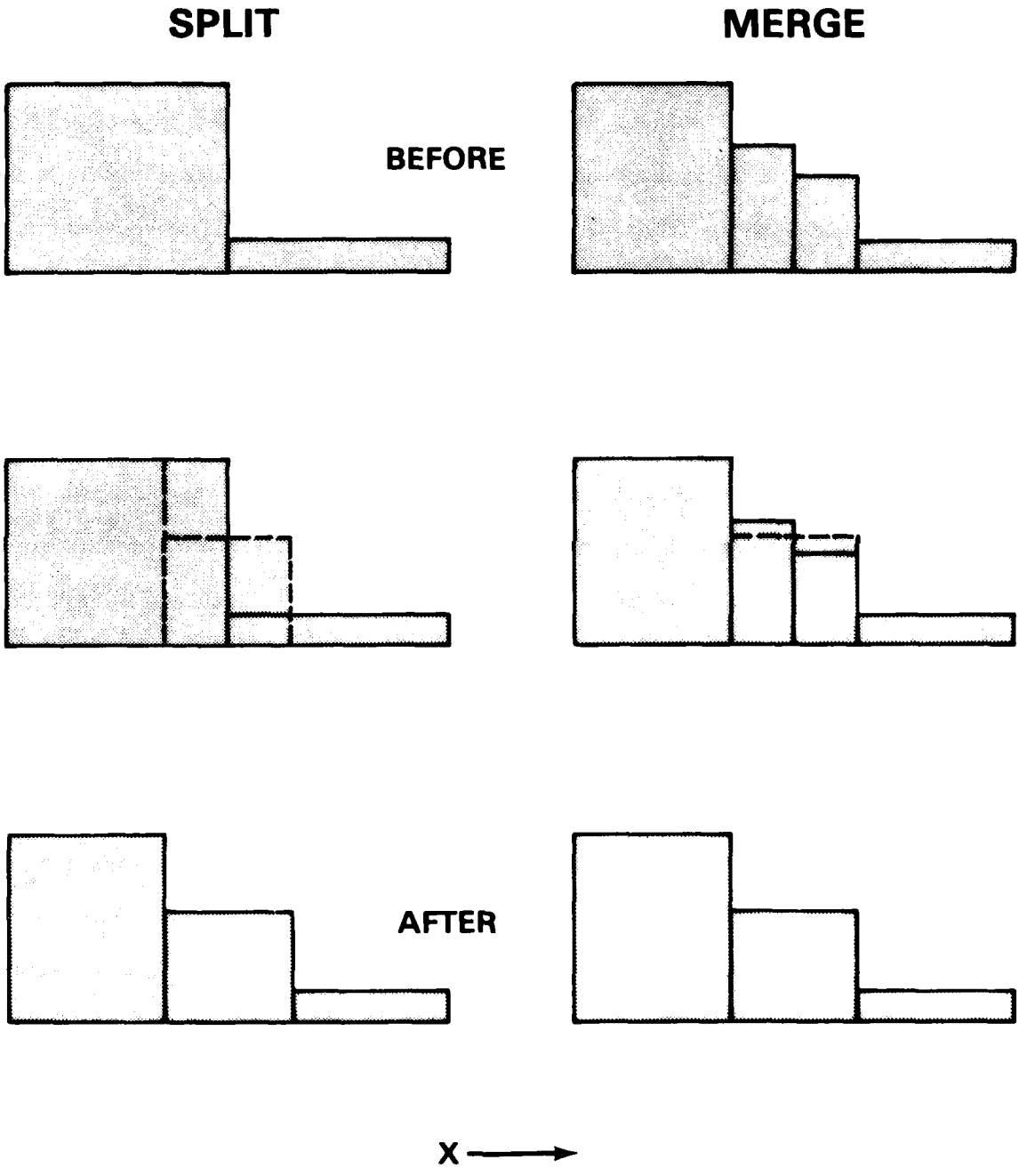


Fig. 3 — Lagrangian rezoning showing how cells are split and merged

B. Two-dimensional algorithms.

The mesh points commonly used to evaluate gradients and Laplacians are shown in Fig. 4a for a regular two-dimensional grid. Figure 4b illustrates a simple grid distortion produced by shear flow. A well-formulated Lagrangian finite-difference algorithm will properly account for the angles between grid lines and the variable mesh spacing produced by this distortion. Nevertheless, numerical approximations based on this mesh can still be grossly in error because differences no longer involve neighboring vertices. Mesh points which are now closer to the central vertex do not enter into the approximation, while those farther away do.

Higher-order approximations may lead to even greater error. Fig. 5a shows the vertices commonly used in higher-order approximations. Fig. 5b shows how these approximations on a distorted mesh may include vertices which are even further removed from the central vertex, while neglecting other vertices which lie closer. In other words, the distorted mesh cannot be used to self-consistently improve the accuracy of the approximation. The problem can be resolved only by ensuring that differencing is carried out over the appropriate vertices. The effective mesh distortion must be reduced.

As in one dimension, the conventional solution to maintaining a regular two-dimensional Lagrangian mesh is an Eulerian rezone phase. And again, this rezoning procedure reintroduces numerical diffusion in an essentially uncontrollable manner. The solution we present preserves the Lagrangian nature of the calculation while greatly reducing the errors associated with the two-dimensional analog of Eq. (7).

Rezoning attempts to solve the problem of grid deformation by using vertex motion to rectify the distortions. The alternate solution is illustrated in Fig. 6. A section of a quadrilateral mesh about a shear layer is shown in

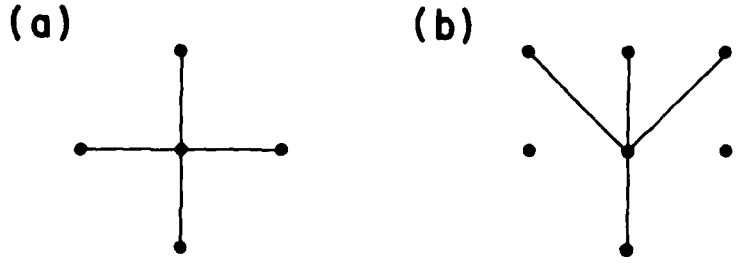


Fig. 4 — (a) Mesh points used to calculated gradients and Laplacians for a two-dimensional grid at the beginning of a calculation. (b) Distortion due to shear flow results in approximations evaluated among points which are no longer nearest neighbors.

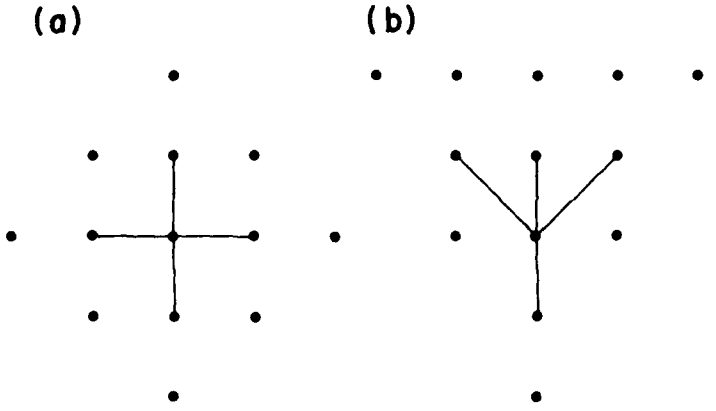


Fig. 5 — (a) Gridpoints used for higher order approximations may become even more distorted (b) due to shear flows

Fig. 6a. A Lagrangian calculation quickly leads to the mesh shown in Fig. 6b, in which mesh connections about the layer no longer join neighboring vertices. In this figure one grid line has been relocated to show a connection which is now appropriate. For a periodic system all stretched grid lines could be reconnected, thereby restoring the mesh to its original configuration without relocating any vertices. In general, some reconnections either are inappropriate or cannot be made due to boundaries, so that one triangular and one pentagonal cell remain (shown in Fig. 6b). Therefore, reconnection on a quadrilateral grid cannot totally resolve the problem of distorted grids.

On a triangular mesh, however, there are no such complications. As shown in Fig. 7, a reconnected grid line on a triangular mesh still results in two triangular cells. This technique, first used in a Lagrangian hydrodynamics code by Crowley (6), represents a very attractive alternative to rezoning since there is no diffusive vertex motion.

Reconnecting grid lines, therefore, can solve the problem of grid distortion due to shear, but the problem of a variable resolution requirement still must be addressed in two dimensions. Reconnection helps. The number of grid lines meeting at any vertex can be reduced to three by reconnections, with the three neighboring vertices forming a triangle which surrounds only one vertex. If the fluid is accumulating too many vertices locally, the central vertex and its three grid lines can be eliminated, leaving only the surrounding triangle. This produces the desired decrease in resolution and avoids the formation of long, narrow triangles near the point of converging flow. Conversely, a vertex may be added inside any triangle or along any line by providing the necessary grid lines to other vertices within the affected triangles. Subsequent reconnections will link the added vertices to other nearer neighbors. In this way the transition to multiply-connected regions and the flow near stagnation

points can be handled smoothly by decreasing or increasing resolution where appropriate. The combination of grid line reconnection with vertex addition and deletion therefore provides a means of smoothly restructuring the grid without recourse to diffusive vertex movement. Since the number of triangles meeting at a vertex is variable, increased accuracy in one region of the flow does not force unnecessary resolution in other areas of the flow.

As in the one-dimensional case, the rezoning criteria can be based on local fluid behavior. Conservation equations are used here also to ensure proper physical behavior in a Lagrangian sense. This flexibility is gained at the expense of increased calculational overhead because of the varying length of grid lines, the arbitrary number of connections to a vertex and the necessary logic involved in labelling of interface and boundary sides. In addition, the lack of global ordering leads to more dependence on iterative solution procedures, since standard fast algorithms are generally inapplicable. Illustrations of the use of these algorithms will be given in section III.

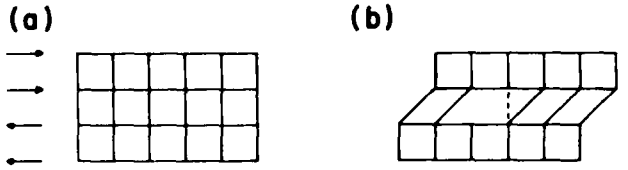


Fig. 6 — (a) Section of quadrilateral mesh at the beginning of a calculation. Arrows indicate direction of fluid motion. (b) Grid is distorted due to shear flow so that points are no longer corrected to nearest neighbors. Reconnection leads to a grid with a mixture of triangles, quadrilaterals, and other polygons.

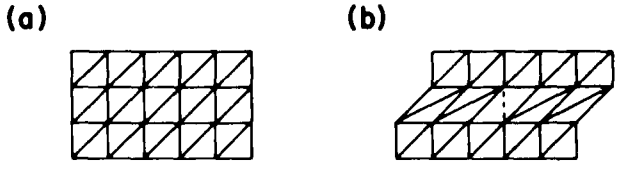


Fig. 7 — Reconnecting a grid tessellated by triangles (a) leads to a grid which contains only triangles

C. Three-dimensional algorithms

The three-dimensional analogue of the triangular grid is a tetrahedral grid in which surfaces are tessalated by triangles. As expected, the addition of a further dimension introduces a further complication, appearing this time as a more complicated reconnection algorithm. As shown in Fig. 8, connecting two previously unconnected points may not preserve the total number of tetrahedra. Here a five-vertex configuration, which originally encompassed two tetrahedra, now must include three tetrahedra because an additional vertex connection is required. This ambiguity exists primarily because of the additional line. Fig. 9 shows a six-vertex configuration for which reconnection must keep the total number of tetrahedra constant.

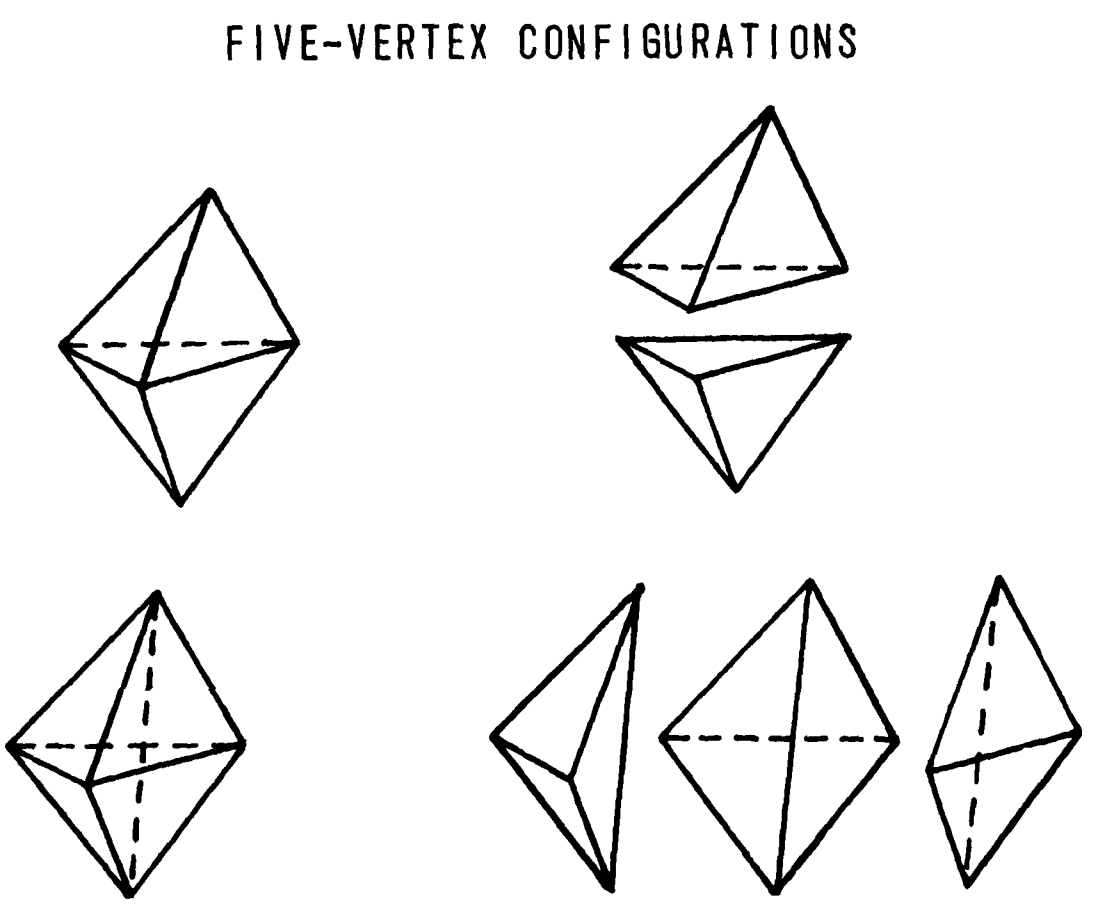


Fig. 8 — Reconnections in a three-dimensional grid of tetrahedra does not preserve the total number of tetrahedra

SIX-VERTEX CONFIGURATIONS

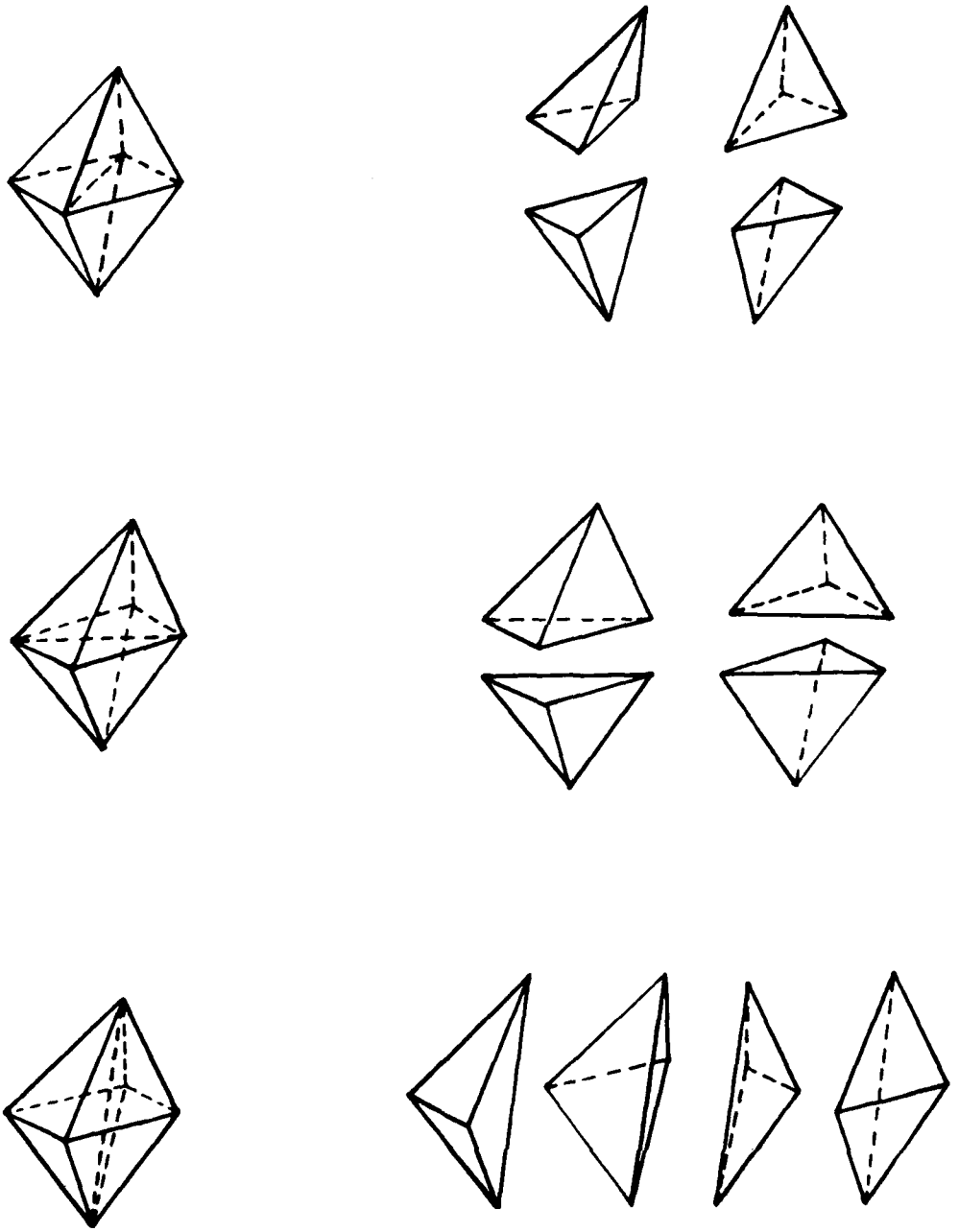


Fig. 9 — Six-vertex configurations of tetrahedra for which reconnection keeps the total number of tetrahedra constant

Despite this additional complication, the bulk of what was learned from the two-dimensional case can be carried over intact to three dimensions. Vertices can still be deleted by successive reconnections to isolate the vertex within a single tetrahedron. At that point, the vertex and its four lines can be eliminated from the grid. Vertices can be added within tetrahedra, in the plane of a triangle and on lines without major modification of the techniques used to ensure conservation of physical quantities in two-dimensions. Therefore, the same representational flexibility can be achieved in three dimensions as in one- and two-dimensions while maintaining the resolution.

The criteria used for reconnection, addition and deletion of cells in three-dimensions usually only involves either extending integrals to one higher dimension, or measuring an angle between planes rather than lines. Similarly, the hydrodynamics finite-difference algorithms are logical extensions of the two-dimensional algorithms.

III. Calculations and Results

A. One-dimensional case

The one-dimensional techniques discussed in Section II have been incorporated in the algorithm ADINC, developed by Boris (7). An implicit momentum equation solver is also included in the algorithm, permitting time-steps to exceed the Courant condition. ADINC solves the equations of mass and momentum transport in the form

$$\frac{d\rho}{dt} = - \rho \nabla \cdot \underset{\sim}{v} \tag{8}$$

$$\rho \frac{d\underset{\sim}{v}}{dt} = - \nabla p. \tag{9}$$

The energy evolution equation is eliminated by using an adiabatic equation of state in which the local entropy s(r) is assumed constant throughout the numerical timestep. Nonadiabatic processes such as external heating, thermal conduction, and chemical energy release can be added to Eqs. (8) and (9) using timestep-splitting, provided sufficiently short timesteps are used to make the splitting procedure accurate. In the version of ADINC given by Boris (7) the equation of state of the fluid in each computational cell is

$$\rho(p, s. \ldots) = \rho_c + (p/s)^{1/\gamma} \tag{10}$$

When $\rho_c = 0$, Eq. (10) describes adiabatic compression and expansion of an ideal gas. When $\rho_c \neq 0$, Eq. (10) represents a mildly compressible liquid.

During an ADINC timestep ρ_c, γ, and s are treated as constants; only the variation of ρ with p is considered. This equation of state density is compared to the density derived from the fluid dynamics through Eq. (8). The difference is iterated to zero using a quadratically convergent implicit solution of Eq. (9)which gives an improved pressure approximation. During this

iteration the analytic derivative $\frac{d\Lambda}{dp}$ is used, where Λ is the volume of a computational cell. Thus,

$$\frac{1}{\Lambda}\frac{d\Lambda}{dp} = -\frac{1}{\gamma\rho p}(p/s)^{1/\gamma} \tag{11}$$

for the particular equation of state Eq.(10).

Because finite errors in pressure and density are expected during the iteration process, ADINC uses the equation of state in the form $\rho(p, s, \ldots)$. Rather than finding the pressure as a function of ρ, ADINC calculates the fluid density from an approximation to the pressure. For liquids and solids the density is a weak function of the pressure. In the other form $p(\rho, s, \ldots)$, errors in density ρ would appear as large pressure fluctuations. For gases and plasma the two forms are basically of the same accuracy. There is a second related reason for using the equation of state in the form $\rho(p, s, \ldots)$. The ADINC algorithm is especially designed to deal with discontinuities in zone size and density. When a gas-solid interface is encountered, the pressure is continuous, but the density need not be. Therefore, finite differences in the pressure are bound to be more accurate than if they are calculated from differences in the density according to $\Delta p = \frac{\partial p}{\partial \rho}\Delta\rho$.

Figure 10 shows a schematic diagram of the computational region treated by ADINC. There are N cells of volume $\Lambda_i (i = 2, 3, \ldots, N + 1)$ bounded by $N + 1$ interfaces of area $A_i (i = 1, \ldots, N + 1)$. The interfaces are located at $r_i (i = 1, \ldots, N + 1)$, so $A_i = A(r_i)$.

The cell interface positions $\{r_i\}$ satisfy

$$\frac{dr_i}{dt} = v_i, \tag{12}$$

which has a straightforward discretization

$$r_i^n = r_i^o + \delta t[\epsilon_r v_i^o + (1 - \epsilon_r)\, v_i^n]. \tag{13}$$

ADINC GRID STRUCTURE AND VARIABLE PLACEMENT

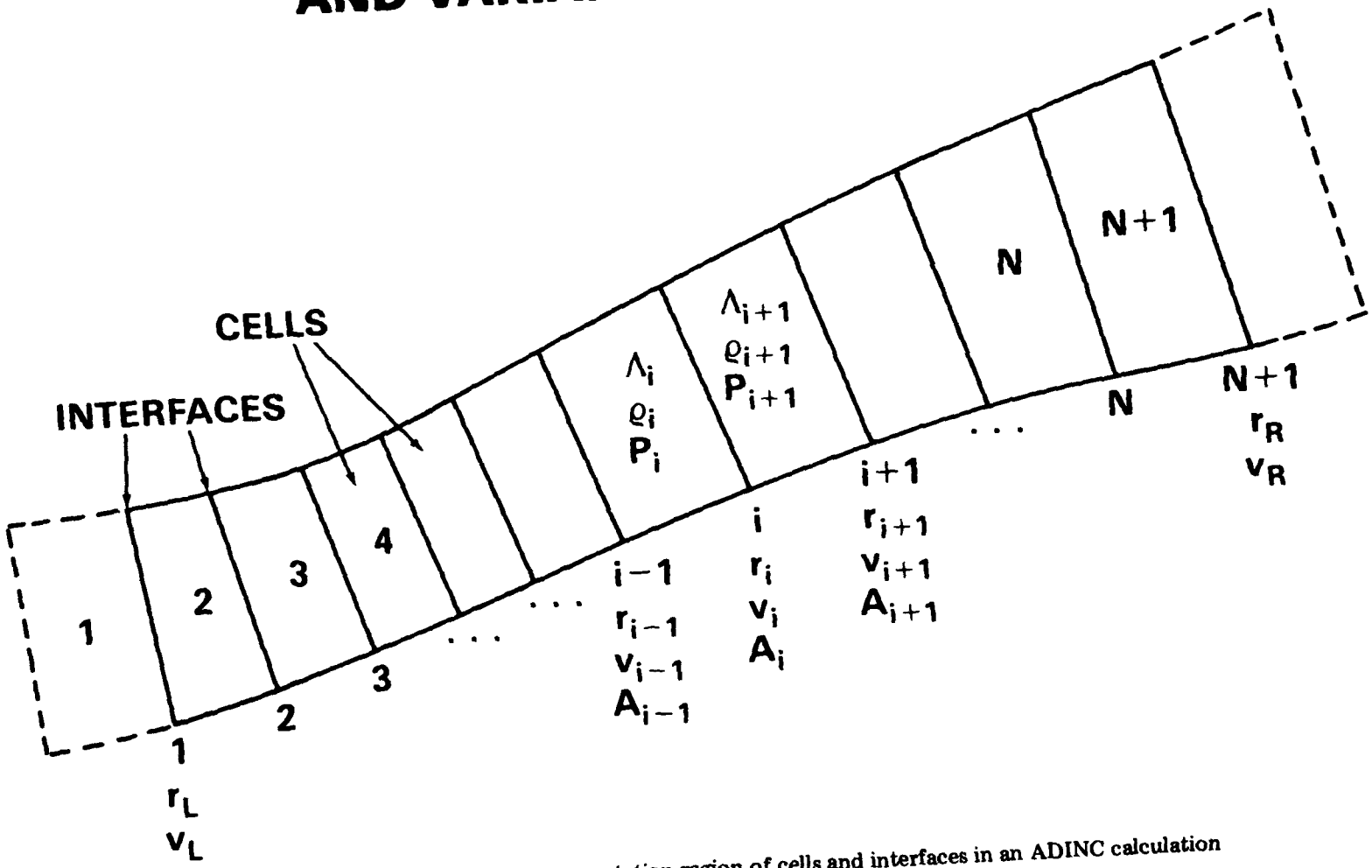


Fig. 10 — Schematic diagram of a computation region of cells and interfaces in an ADINC calculation

In Eq.(13), the superscript "n" indicates variables at the time $t + \delta t$ while superscript "o" indicates variables at the old time t. The quantity ε_r is the explicitness parameter for the interface position, $0 \leq \varepsilon_r \leq 1$. When $\varepsilon_r < 1$, the method is at least partially implicit. When $\varepsilon_r = 1/2$, the method is centered and nominally most accurate. If long timesteps are contemplated, $\varepsilon_r \leq 1/2$ is required for Courant stability, with strict inequality usually required to deal with nonlinear effects. When $\varepsilon_r = 0$, the calculation is fully implicit, i.e., fully forward-differenced. This choice is most stable but is only first-order accurate. Currently ADINC uses the same value of ε_r at every interface, varying it from cycle to cycle.

The momentum equation Eq.(9) yields the acceleration of interface i,

$$\frac{dv_i}{dt} = \frac{-1}{\rho_i} \left.\frac{\partial p}{\partial r}\right|_i , \tag{14}$$

where the density and pressure gradient are also needed at cell interfaces. The discretization used in ADINC is

$$v_i^n = v_i^o - \frac{\delta t \varepsilon_v}{\langle \rho \delta r \rangle_{i+\frac{1}{2}}} (p_{i+1}^o - p_i^o) - \frac{\delta t (1-\varepsilon_v)}{\langle \rho \delta r \rangle_{i+\frac{1}{2}}} (p_{i+1}^n - p_i^n). \tag{15}$$

Here ε_v is the explicitness parameter for the interface velocity and has the same properties described above for ε_r. The quantities ε_r and ε_v are distinct in ADINC, but no reason has been uncovered to date for using different values in an actual calculation. The interface average indicated as $< \rho \delta r >_{i+1/2}$ is both a spatial and temporal average. Physical considerations are used to define $< \rho \delta r >_{i+1/2}$, so the discretization in Eq.(15) is insensitive to numerical errors arising from large density discontinuities at the interfaces.

The new pressures $\{p_i^n\}$ are found iteratively through a tridiagonal equation derived as follows. The momentum equation, Eq. (15), can be simplified to

$$v_i^n = a_i - b_i(p_{i+1}^n - p_i^n) \tag{16}$$

for i = 2, ..., N, where

$$\begin{aligned} a_i &= v_i^o - \frac{\delta t \varepsilon_v}{\langle \rho \delta r \rangle_{i+\frac{1}{2}}} (p_{i+1}^o - p_i^o), \\ b_i &= \delta t(1 - \varepsilon_v)/\langle \rho \delta r \rangle_{i+\frac{1}{2}}. \end{aligned} \tag{17}$$

The equation of state is introduced by requiring that the cell volume Λ_i^{eos} computed from the equation of state using the new time value of pressure equal the new cell volume computed from the fluid dynamics, Λ_i^{fd}. At any iteration the difference is

$$\delta\Lambda_i = \Lambda_i^{eos}(p_i, s_i, \cdots) - \Lambda_i^{fd}(\{r_i\}). \tag{18}$$

Iteration should be continued until $\delta\Lambda_i$ vanishes. Changing p_i^p to p_i^n varies both terms in Eq. (18) where the superscript "p" denotes a provisional value. In the fluid dynamics contribution r_i^p converges to r_i^n as a function of the pressure through Eqs. (16) and (13). We use a Newton-Raphson approach to obtain a quadratically convergent iteration to the desired solution at time t + δt,

$$\Lambda_i^{eos}(p_i^n, s_i^n, \cdots) = \Lambda_i^{fd}(\{r_i^n\}).$$

When integrating the fluid dynamic equations, ADINC assumes that each interface moves in a fully Lagrangian manner according to Eq.(12). The change in density from one timestep to the next in a cell is therefore given simply by the change in cell volume according to the mass conservation equation

$$\rho_i^n \Lambda_i^n = \Delta M_i = \rho_i^o \Lambda_i^o, \tag{19}$$

When individual species number densities must be followed, they are also advanced by Eq. (19).

The flame ignition studies of Oran and Boris (4,5), Indritz et al (8) and Kailasanath et al (9) are examples of combustion calculations carried out using ADINC. The computational model consists of a solution of the time-dependent conservation equations for multispecies reactive flow. Here ADINC, which is used to do the convective transport, is coupled to algorithms which calculate the effects of chemical energy release and diffusive transport of particles and heat.

Results from a typical flame calculation are shown in Figs. 11,12 and 13. Figure 11 shows a typical temperature profile for a flame initiated by a quadratic temperature distribution at the onset of the calculation. Also drawn on the same figure is a plot of the cell size Δx as a function of position, showing that the resolution decreases as the absolute magnitude of temperature gradient increases. It is clear that the considerable variation in cell size introduces no related variation in the temperature, as would occur if the results of the convective transport depended strongly on the local value of Δx. Species profiles are shown in Fig. 12 and details of the flame front in Fig. 13. We note that the intermediate species densities and the temperature gradients do not all peak at the same location, and the structure of the flame front is clearly distinguishable.

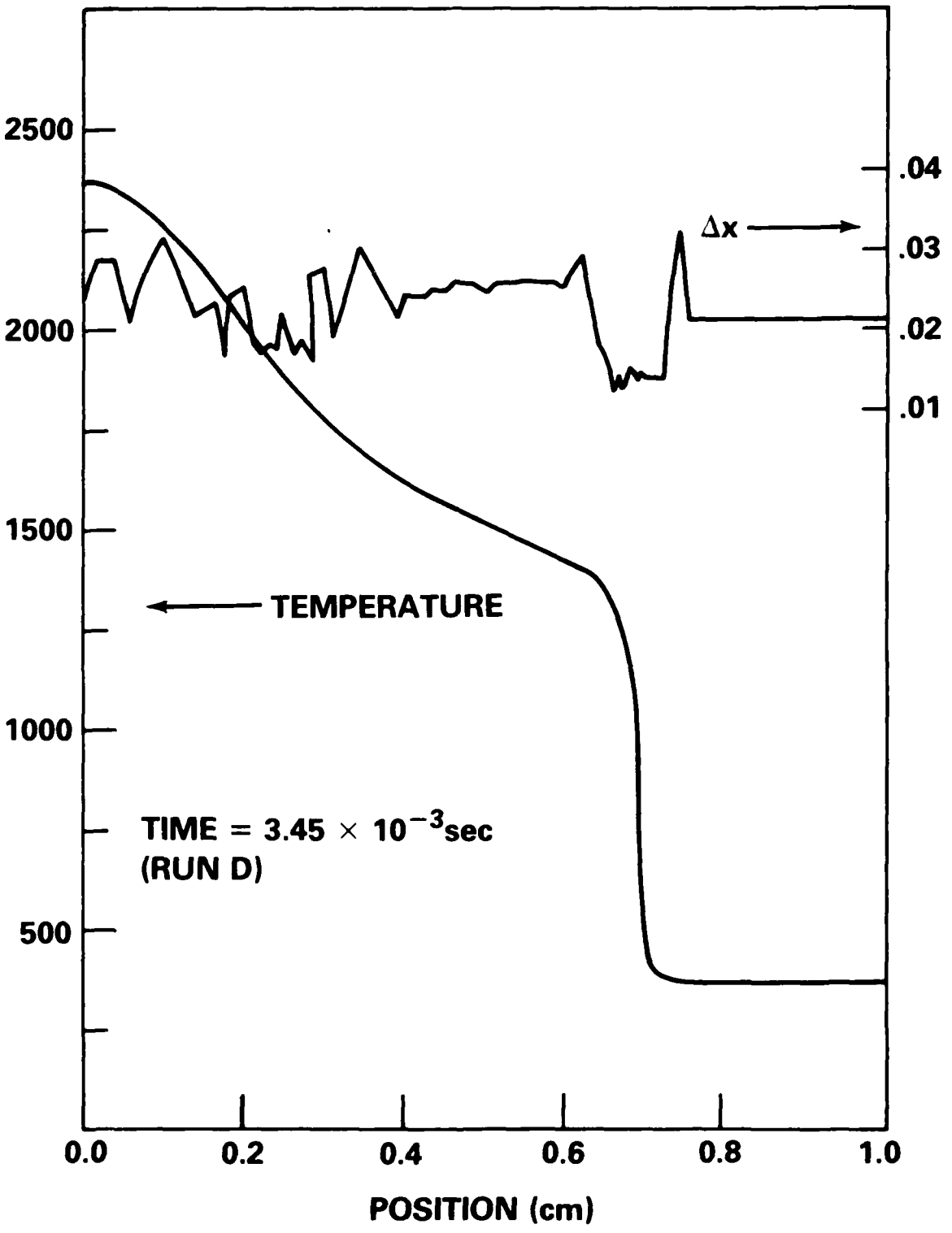


Fig. 11 — Temperature and cell size as a function of position in a typical flame propagation

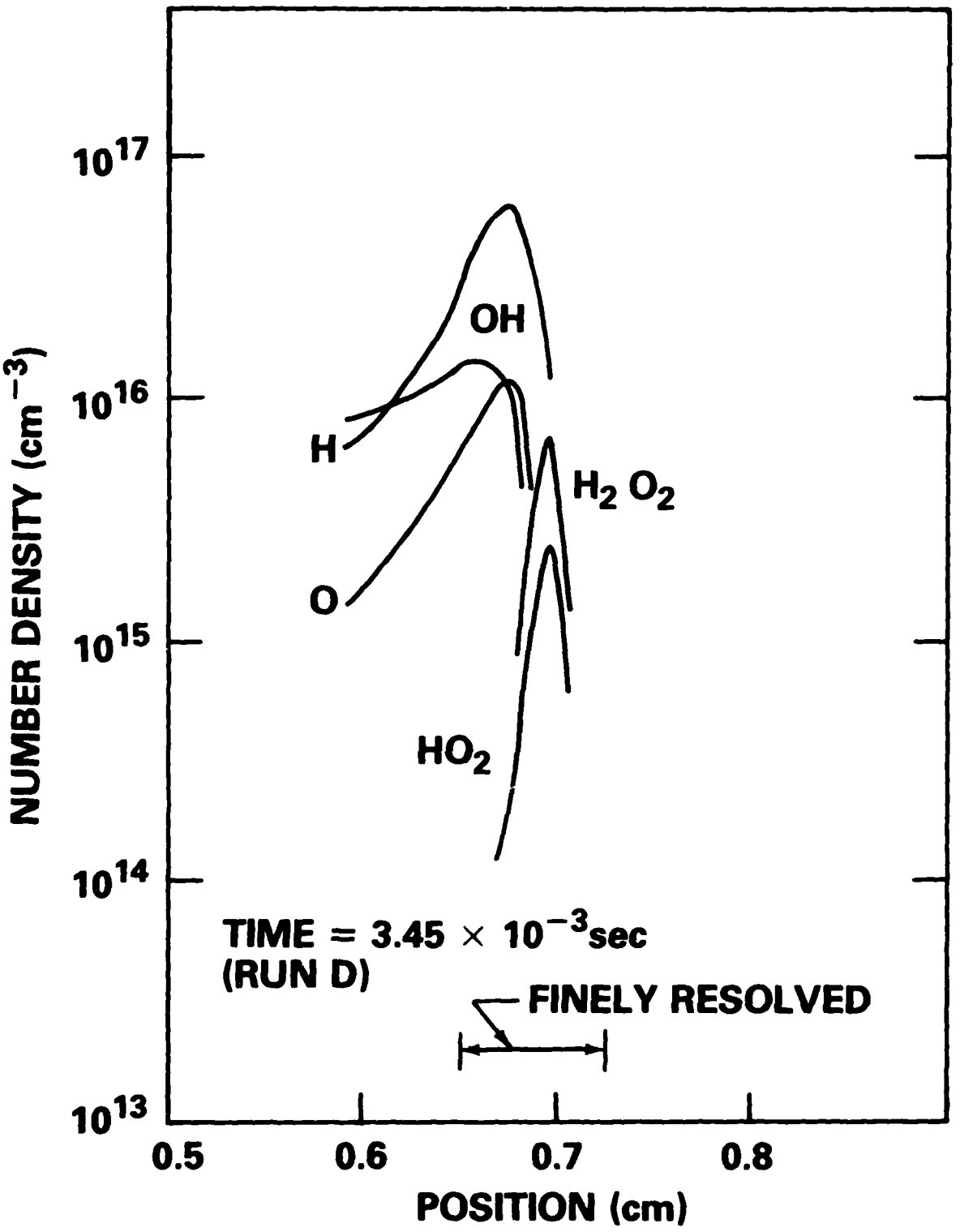


Fig. 12 — Detail of the flame front in a one-dimensional flame propagation calculation

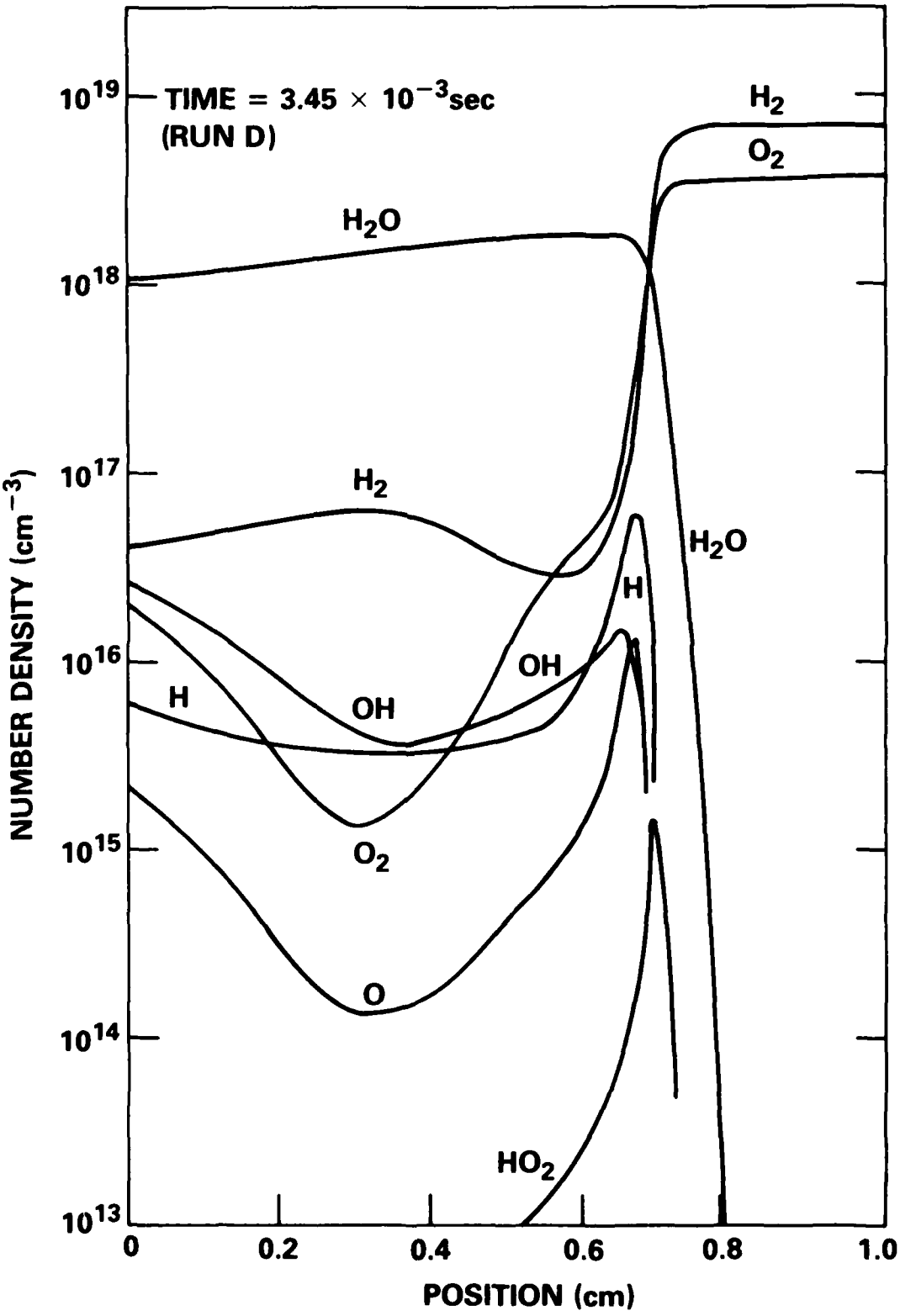


Fig. 13 — Number density as a function of position in a one-dimensional calculation of a flame propagating through a hydrogen-oxygen mixture.

B. Two-dimensional case

Figure 14 shows a section of a triangular mesh representation with an interface between two fluids of types I and II. The basic elements involved in the construction of a triangular mesh are shown. In Fig. 14a, a particular triangle j is shown in heavy lines and the various components of the triangle are labeled. Three vertices, V_1, V_2 and V_3, are connected consecutively by sides S_1, S_2 and S_3. The direction of labeling around each triangle is counterclockwise and the z axis is directed out of the page. Since the mesh can be irregularly connected, an arbitrary number of triangles can meet at each vertex.

Figure 14b illustrates several important properties of triangles which are used in constructing finite-difference algorithms. It is convenient to define a cell surrounding a vertex, as shown by the shaded region surrounding V_3. The borders of such vertex-centered cells are determined by constructing all of the side bisectors for each triangle. Since the three side bisectors all intersect at a point, as shown for triangle j, there is no ambiguity in constructing the vertex cells as indicated.

A vertex centered cell must be defined because the cell size should be preserved in exactly the same manner as indicated for the one-dimensional case. On a quadrilateral mesh, there is a one-to-one correspondence between quadrilaterals and vertices, except at boundaries. It is therefore possible to try to conserve quadrilateral areas by specifying pressures at the center of each quadrilateral to maintain its size. If diagonals are drawn through each quadrilateral, a triangular mesh results with roughly twice as many triangles as vertices. Therefore, the pressures specify twice as many triangular cell areas to be conserved, but with exactly the same number of vertices to be moved. For this reason vertex cell volumes are defined and used instead to maintain the proper counting between pressures and positions, both of which are specified at the vertices.

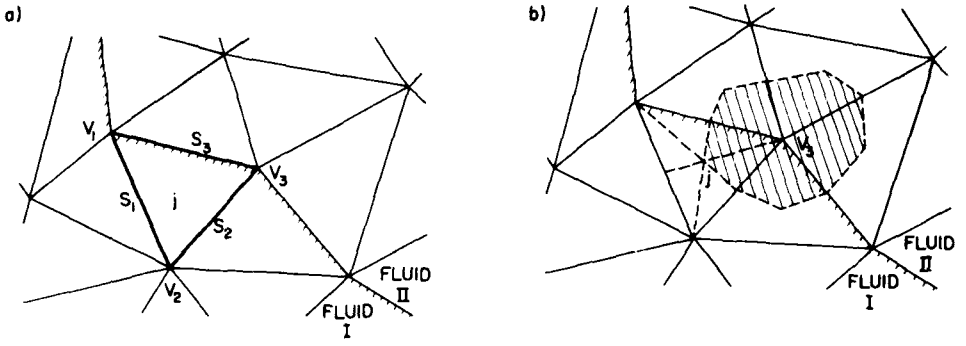


Fig. 14 — Section of a triagular mesh with an interface between fluids of types I and II. (a) The "j" labels a triangle surrounded by sides S_1, S_2, and S_3 and verticles V_1, V_2, and V_3. (b) Construction of a cell around vertex V_3.

The use of pressures defined on vertices permits a definition of pressure gradients in direct analogy with Eq. (3). We can rewrite Eq. (3) using Eqs. (1) (2) for the forward and backward approximations:

$$\frac{\partial p}{\partial x_i} = \frac{\left(\frac{\partial p}{\partial x_i}\right)^{+}\Delta x_i + \left(\frac{\partial p}{\partial x_i}\right)^{-}\Delta x_i'}{\Delta x_i + \Delta x_i'} . \tag{20}$$

That is, the central difference can be obtained by an area-weighted sum of the forward and backward differences.

This result carries over directly to general triangular grids in two dimensions. We can define the pressure gradient there as

$$\nabla p = \sum_{i=1}^{3} p_i \frac{\hat{z} \times (\underline{r}_{i-1} - \underline{r}_{i+1})}{2A} \tag{21}$$

The right-hand side of Eq. (21) is the first-order accurate finite difference approximation to the gradient, evaluated at the triangle centroid. Here A is the triangle area, $\hat{z}$ is the unit vector in the direction of the ignorable coordinate, and the sum extends over the three triangle vertices. The analog of Eq. (20) is

$$\nabla p_i = \frac{\sum_{j=1}^{3} A_j \nabla p_j}{\Sigma A_j} , \tag{22}$$

where the index j labels triangle-centered quantities and i labels vertices. For special geometries Eq. (22) is second-order accurate or higher. In most situations it is less accurate, but it yields accuracy reasonably close to second order for a general mesh, provided that the triangle areas are nearly equal. In that case the error is determined by a formula similar to Eq. (7).

The worst that can be achieved is first-order accuracy, which occurs only in the degenerate case of a zero-area triangle.

In a cell-centered scheme, the pressures are located half a cell away, and boundary conditions, particularly at free surfaces, are more difficult to implement. Accuracy is diminished primarily by narrow triangles in the interior of the fluid. This restriction is not too serious for a reconnecting grid, since the grid can be made to reconnect to preserve regular triangles. Where this is not possible (near interfaces, for example), the addition or deletion of vertices can be used to regularize the mesh.

To date most of the simulations done with the triangular gridding method have been of inviscid, incompressible fluids. However, the conservative integral approach and definitions of divergence used allow a natural extension to compressible flow, as discussed at the end of this section. The basic equations of the system presented here are:

$$\frac{d\rho}{dt} = 0 \tag{23}$$

$$\nabla \cdot V = 0 \tag{24}$$

$$\rho \frac{dV}{dt} + \nabla p = F_e \tag{25}$$

The fluid density ρ, pressure p, and velocity v are assumed to vary only with x and y. Equation (24), the condition for incompressibility, removes the sound waves. We will assume that p is a constant along free surfaces.

With pressures specified at the vertices, ∇p is evaluated over triangles, and Eq. (25) can easily be advanced implicitly or explicitly if velocities are considered to be triangle-centered. In what follows, we

will continue to use the subscript i to denote a vertex-centered quantity and j to denote a triangle-centered quantity. The integration of velocities uses a split-step algorithm in which the velocities are advanced one-half timestep, the grid is advanced a full timestep and then the velocities advanced forward the other half timestep:

$$\underset{\sim}{v}_j^{\frac{1}{2}} = \underset{\sim}{v}_j^{o} - \frac{\delta t}{2\rho_j} (\nabla p)_j^{o} - \frac{\delta t}{2} g\hat{\underset{\sim}{v}} \; ; \tag{26}$$

$$\underset{\sim}{v}_i^{\frac{1}{2}} = \frac{1}{2}(\underset{\sim}{v}_i^{o} + \underset{\sim}{v}_i^{n}) \; ; \tag{27}$$

$$\underset{\sim}{x}_i^{n} = \underset{\sim}{x}_i^{o} + \delta t \, \underset{\sim}{v}_i^{\frac{1}{2}} ; \tag{28}$$

$$\underset{\sim}{v}_j^{\frac{1}{2}} = \underset{\approx}{R}([\underset{\sim}{x}_i^{o}], [\underset{\sim}{x}_i^{n}]) \cdot \underset{\sim}{v}_j^{\frac{1}{2}}) ; \tag{29}$$

$$\underset{\sim}{v}_j^{n} = \underset{\sim}{v}_j^{\frac{1}{2}} - \frac{\delta t}{2\rho_j} (\nabla p)_j^{n} - \frac{\delta t}{2} \underset{\sim}{g\hat{v}} \; . \tag{30}$$

The vertex velocity $\underset{\sim}{v}_i^{n}$ appearing in Eq. (27) is obtained from the area-weighted $\underset{\sim}{v}_j^{n}$ from the previous iteration,

$$\underset{\sim}{v}_i^{n} = \frac{\sum_j \underset{\sim}{v}_j^{n} A_j}{\Sigma A_j} \; . \tag{31}$$

The advantage of using triangle-centered velocities is the ease in understanding and expressing conservation laws. Because of the paucity of experience in formulating algorithms over a general triangular grid, we have employed the control volume approach which uses an integral formulation to derive the difference algorithms. Equation 29 is a consequence of this approach. It reflects numerically the fact that the triangle velocities must rotate and

stretch as the grid rotates and stretches. The transformation $\underset{\approx}{R}$ is derived by considering the circulation about each vertex. Since the triangle velocities are constant over the triangle, the circulation taken about the boundary of the vertex cell is straightforward to calculate. The conservation of vorticity then takes the form of the operator $\underset{\approx}{R}$ which preserves the value of the circulation about each vertex while maintaining the proper divergence at each cell. This transformation ensures that the vorticity integral calculated about any interior vertex is invariant under the advancement of the grid. It is easy to show that the ∇p and gravity terms cannot alter the vorticity either, since $\nabla \times \nabla p = 0$ identically and gravity is a constant. Only the $\nabla p_j/\rho_i$ term can change vorticity, exactly as dictated by the physics. Since the transformation $\underset{\approx}{R}$ is time-reversible, Eqs.(26-30) are also. Hence the entire algorithm advances vertex positions and velocities reversibly while evolving the correct vorticity about every interior vertex.

This technique is unique for Lagrangian codes, which usually either ignore vorticity conservation completely or conserve vorticity through an iteration simultaneously with the pressure iteration. In this technique the vorticity is conserved exactly regardless of whether the pressures have iterated to their final values.

The pressures $\{p_i^n\}$ in Eq. (30) are derived from the condition that the new velocities $\{v_j^n\}$ should be divergence-free at the new timestep, satisfying Eq. (24). The pressure Poisson equation is derived from Eq. (30) by setting $(\nabla \cdot V_j^n) = 0$ to obtain a pressure p_i^n such that

$$\frac{\delta t}{2}\left(\nabla \cdot \frac{1}{\rho_j}(\nabla p)_j\right)_i = (\nabla \cdot v_j^{n/2})_i \qquad (32)$$

Both terms in Eq. (32) are simple to evaluate, since the divergence is taken over triangle-centered quantities. Three features of the Poisson equation

Eq. (32) are noteworthy. First, it is derived from $\nabla^2\phi = \nabla \cdot \nabla\phi$, as in the continuum case. Second, the left-hand side results in the more familiar second order accurate templates (such as the five-point formula) for the Laplacians found in connection with homogeneous fluids and regular mesh geometries. Finally, it is directly analagous to the one-dimensional case in that pressures are found iteratively and correspond to the divergences generated by advancing the grid according to the fluid dynamic equations.

The derivation of the reconnection and vertex addition and deletion algorithms are also done through the control volume approach and the use of triangle velocities. For all of the algorithms used, the divergence and curl taken about each vertex are both identically conserved for grid reconnections and vertex addition. When a vertex is deleted, the vorticity and divergence carried by the deleted vertex are reapportioned by an area-weighted algorithm. Mass and momentum are also identically conserved.

A finite-difference code SPLISH (10) which used the algorithms discussed above has been tested on a series of physical problems. These problems were designed specifically to exercise particular algorithms as they were added, and to test as well as overall accuracy of the code against well-known analytic and experimental solutions. The basic hydrodynamic algorithms were tested by simulations of nonlinear free-surface waves. Figure 15 shows a plot of the wave natural period versus grid size for four different resolution calculations of the same physical problem. The second-order accuracy over long times is demonstrated by the parabolic form of the curve. Going to smaller stepsize gives improved accuracy, as expected. An added advantage accrues from using triangular cells. The nonlinear mesh-separated instabilities which plague low-dissipation rectangular cell techniques seem to be absent when triangular cells are used.

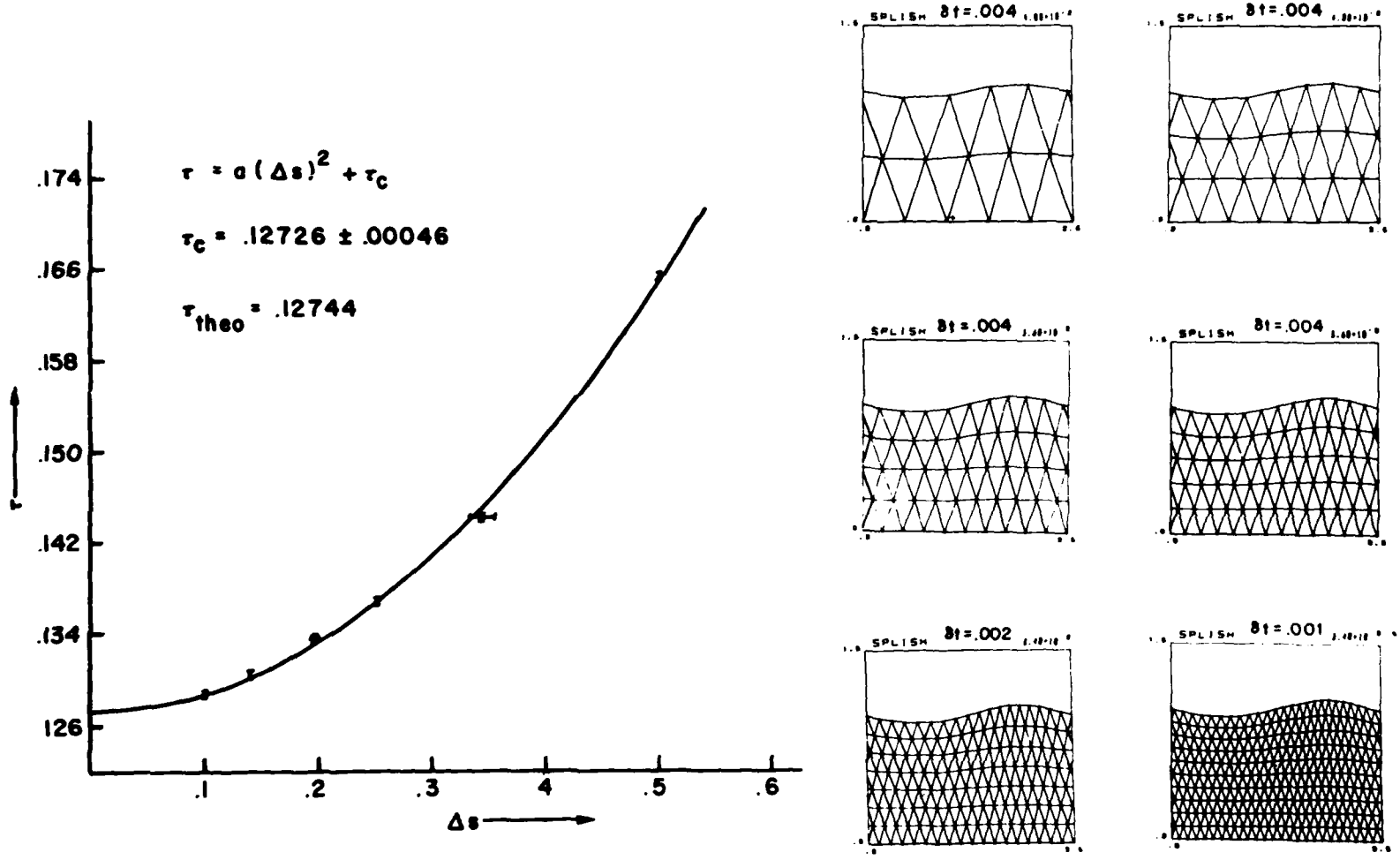


Fig. 15 — Wave period as a function of grid size for six different resolution calculations. Insets show grid structures for the six calculations.

A further test of the hydrodynamic algorithms was made by calculating the pressures predicted numerically for waves incident on a submerged half-cylinder (12). The numerical results were compared directly with fifth-order theory and with experiment. A half-cylinder of radius "a" in submerged in a fluid whose free surface was at a height $h = 2a$ above the otherwise flat bottom. A progressive wave train with $\lambda = 5a$ was incident on the cylinder. Figure 16 illustrates the numerical results when the wave crest is directly over the cylinder, as compared with both theory and experiment.

It was found that to within experimental error, all of the observed discrepancies could be explained by two factors. The first factor is that the model did not exactly describe the physical situation in the experiment: the wave tank experiments had a single cylinder whereas the calculation is for a series of cylinders due to periodic boundary conditions. The second factor was the surprising result that the roughly 5% reflected wave from the wave tank significantly affected the experimental results due to modifications in the dynamic pressure fluctuations. In this instance a detailed examination of the model and experimental results has indicated that an experimental effect thought to be small could in fact cause noticeable deviations in the data measured.

The reconnection algorithms have been tested through a shear flow calculation of the Kelvin-Helmholtz instability in both the linear and nonlinear regimes (14). The linear growth rates were found to agree well with theory, and the simulations reproduced the physical behavior of subsequent roll-up into billows.

An even more difficult test of a Lagrangian code has been carried out with an unstable density gradient. The Rayleigh-Taylor calculation for two fluids of density ratio 2:1 is patterned on the situation found in a simple laboratory demonstration and displays all of the linear and nonlinear features observed experimentally (Fig. 17).

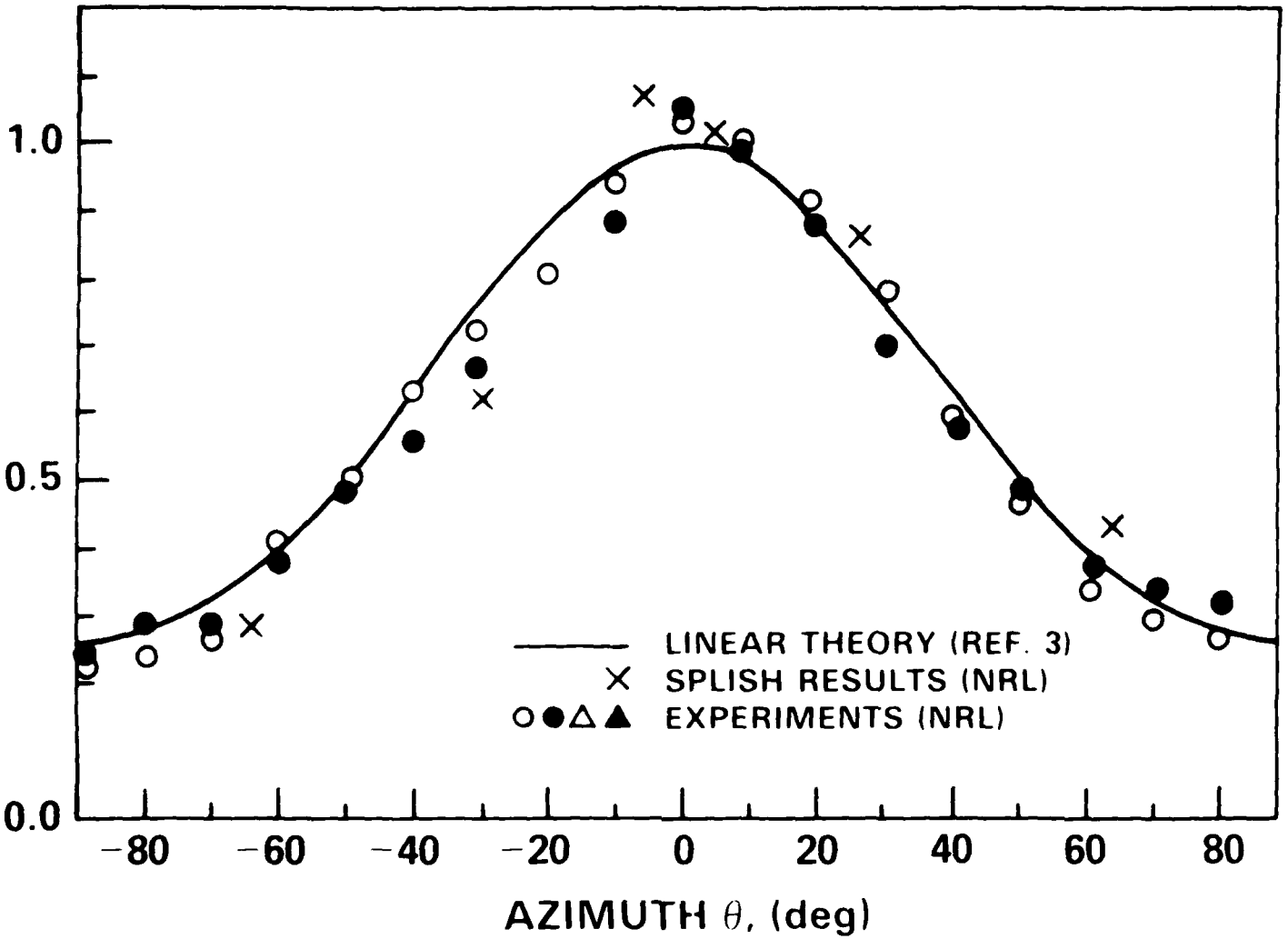


Fig. 16 — Comparison of numerical simulations using SPLISH, linear theory, and experiments for the pressure over the surface of a submerged half-cylinder

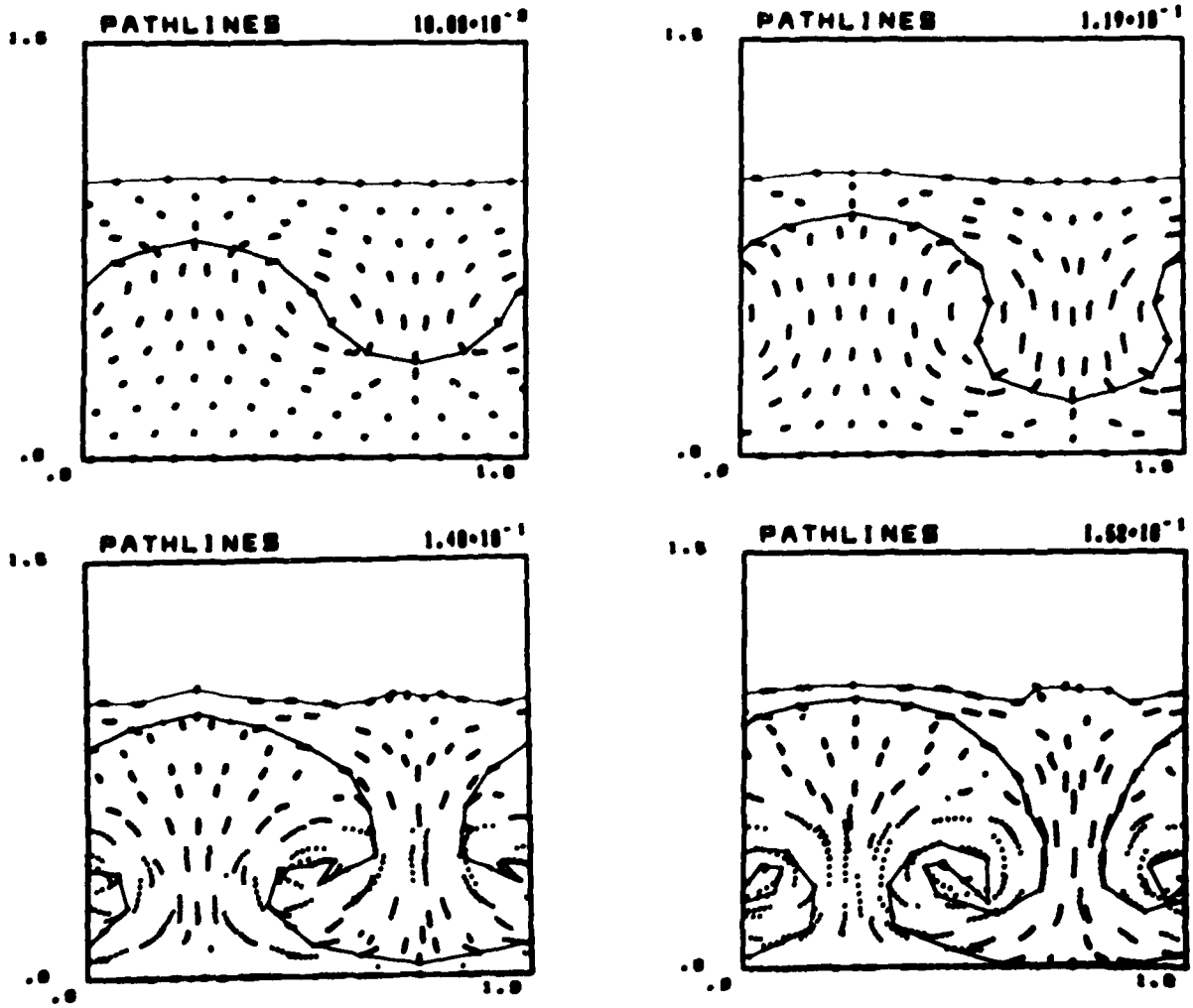


Fig. 17 — Numerical simulations using SPLISH of a Rayleigh-Taylor instability near a free surface

An initial grid of triangles was established with a small sinusoidal perturbation of the interface at $t = 0$. The initial velocities were all taken to be zero. The flow field is followed as the instability enters its nonlinear phase. The vertex locations are plotted at successive timesteps to give streaks whose lengths are proportional to the local fluid velocity. Shear flow is established as the interface steepens, giving rise to vortices due to the onset of the Kelvin-Helmholtz instability (15). The two fluids are observed to mix in the vortices, entraining enough lighter fluid to form "bubbles," which are about to be completely enveloped by the heavier fluid.

Near the end of the calculation most of the grid carried by the lighter fluid has flowed out from below the downward-jetting heavier fluid. Similarly, vertices in the heavier fluid have been deleted above the upswelling lighter fluid. A reasonable resolution has been maintained by adding other vertices, particularly along the interface. Originally the interface was resolved with 10 vertices, but by the end of the run forty-six were required.

The results of these tests indicated that the Lagrangian grid rezoning algorithms do indeed permit accurate calculations to proceed in a consistent manner without a diffusive Eulerian rezone phase. The case of an incompressible fluid was used for these calculations, but the extension to compressible fluids can be done in two ways. The simplest way is to introduce the equation of state in exactly the same manner as in the one-dimensional ADINC code. The cell volume computed from the equation of state using the new pressures is iterated to the cell volume computed from the fluid dynamic variables (Eq.18). This technique would require an additionalterm on the right-hand side of Eq. 32 to represent the volume difference. The term does in fact already exist for the incompressible case, since a residual error correction term has been added to the code to account for incomplete convergence. This residual

error term has exactly the form of a volume difference which is to be iterated to zero. A scheme which would be capable of following strong shocks would have to include an equation for energy transport as well. From the new energies and pressures, a predicted density for each cell can be computed using the equation of state. This density can be compared with the density found from mass conservation using the new physical variables from the fluid dynamics. Any difference in the two densities is then iterated to zero. Chemical reactions and diffusive transport effects may also be added by the same time-step splitting method used in the one-dimensional model.

IV. Conclusion

The application of Lagrangian methods to one-dimensional problems in combustion (4.16) and plasma physics (17,18) has given us confidence in the basic ideas underpinning the Lagrangian adaptive gridding method of splitting and merging cells. The fact that these same ideas have been successfully used in two-dimensional hydrodynamics calculations has shown that it is indeed possible to do non-diffusive multi-dimensional Lagrangian calculations.

The next step is to modify the two-dimensional algorithm so that it will handle compressible flows. Several methods have been proposed, ranging from a version which will be adequate for most combustion problems where characteristic flow velocities are subsonic to a fully compressible version capable of treating strong shocks. These algorithms are currently being tested and implemented in one version of the SPLISH code. The final step in development will be to add the chemical and diffusive transport terms required for a simulation of a combustion system.

Current plans include application of the model to a set of problems covering a large range of time and space scales. These include studies of large scale structures due to voriticity generation, vortex pairing and transport which dominate the fluid dynamics of shear layers. Also of particular interest is the calculation of the properties of liquid fuel droplets in both the evaporating and burning stages. A two-or three-dimensional model which can maintain material interfaces is ideal for studying the circulation pattern around and in droplets as they evaporate and burn.

Acknowledgments

This work has been supported by the Office of Naval Research, the Naval Research Laboratory, the Naval Material Command, and the Department of Energy.

REFERENCES

1. A. Roshko, "Structure of Turbulent Shear Flows: A New Look" AIAA Journal, 14, 1349-1357, 1976; A. Roshko, "Progress and Problems in Understanding Turbulent Shear Flows," (Ed. S.N.B. Murthy), 295-311, Plenum, 1975.
2. F. K. Browand and P. D. Weidman, "Large Scales in the Developing Mixing-Layer," J. Fluid Mech., 76, 127-144, 1976; F. K. Browand and J. Laufer, "The Role of Large Scale Structures in the Initial Development of Circular Jets," Proc. 4th Biennial Symp. Turbulence in Liquids, Univ. Missouri-Rolla, 333-334, Science Press, 1975.
3. N. A. Chigier and A. J. Yule, "The Structure of Eddies in Turbulent Flames-I," Technical Report, Project Squid, Purdue University, 1979; and N. A. Chigier, and A. J. Yule, "The Physical Structure of Turbulent Flames," AIAA Paper 79-0217, New York, 1979.
4. E. S. Oran and J. P. Boris, "Theoretical and Computational Approach to Flame Ignition," to appear in Proc. International Colloquium on Gas Dynamics of Explosions and Reactive Systems, Göttingen, 1978.
5. E. S. Oran and J. P. Boris, "Detailed Modelling of Combustion Systems," Prog. Energy Combust. Sci. 7, 1, 1981.
6. W. P. Crowley, "FLAG: A Free-Lagrange Method for Numerical Simulating Hydrodynamic Flows in Two Dimensions," Proceedings of the 2nd International Conference on Numerical Methods in Fluid Dynamics, Springer-Verlag, New York, 1971.
7. J. P. Boris, "ADINC: An Implicit Lagrangian Hydrodynamics Code," NRL Memorandum Report 4022, Naval Research Laboratory, Washington, D.C. 20375, 1979.
8. D. Indritz, J. Boris, H. Carhart, E. Oran, R. Sheinson, F. Williams, and T. Young, Computation of Hydrogen-Oxygen Flammability Litmits, submitted to Combustion and Flame.

9. K. Kailasnath, E. Oran and J. Boris, A Theoretical Study of the Ignition of Pre-Mixed Gases, submitted to Combustion and Flame.

10. M. J. Fritts, and J. P. Boris, "The Lagrangian Solution of Transient Problems in Hydrodynamics Using a Triangular Mesh," J. Comp. Phys. 31, 173, 1979.

11. M. J. Fritts, "Transient Free Surface Hydrodynamics," NRL Memorandum Report 3651, 1977.

12. M. J. Fritts, E. W. Miner and O. M. Griffin, "Numerical Calculation of Wave Sturcture Interaction," Computer Methods in Fluids, Pentech Press, London, 1, 1980.

13. E. W. Miner, O. M. Griffin, S. E. Ramberg and M. J. Fritts, "Numerical Calculations of Surface Wave Effects on Marine Structures," NRL Memorandum Report 4395, 1980.

14. M. J. Fritts, Science Applications, Inc. Report, SAI-76-632-WA, 1976.

15. J. Orens, D. Book, J. Boris, M. Emery, M. Fritts, J. Gardner and P. Moffa, "Rayleigh-Taylor and Related Fluid Instabilities in Inertial Confinement Fusion," Eighth International Conference on Plasma Physics and Controlled Nuclear Fusion Research, Brussels, Belgium, 1-10 July 1980.

16. C. M. Lund, "HCT - A General Computer Program for Calculating Time-Dependent Phenomena Involving One-Dimensional Hydrodynamics, Transport, and Detailed Chemical Kinetics," UCRL-52504, Lawrence Livermore Laboratories, Livermore, CA, 1978.

17. A. L. Cooper, J. M. Pierre, P. J. Turchi, J. P. Boris and R. L. Burton, Modeling of Linus-Type Stabilized Liner Implosions," Megagauss Physics and Technology, p. 447, Plenum Press, New York, 1980.

18. G. B. Zimmerman, "Numerical Simulation of the High Density Approach to Laser Fusion," Lawrence Livermore Laboratory, UCRL 74811, 1973.


DA1
FILM