CONF-8504150--1

RECEIVED BY OSTI JUN 07 1985

LA-UR--85-1882

DE85 012762

TITLE COMPRESSIBLE LAGRANGIAN HYDRODYNAMICS WITHOUT LAGRANGIAN CELLS

AUTHOR(S) Robert A. Clark

SUBMITTED TO A. M. Crawley, ICFD Secretary, Department of Mathematics
University of Reading, P. O. Box 220, Whiteknights Reading
RG6 2AX, UNITED KINGDOM, for proceedings, April 1 through 4, 1985

DISCLAIMER

MASTER

Los Alamos Los Alamos National Laboratory
Los Alamos, New Mexico 87545

FORM NO 836 R4
ST NO 2629 5/81

COMPRESSIBLE LAGRANGIAN HYDRODYNAMICS WITHOUT LAGRANGIAN CELLS

Robert A. Clark

Computational Physics

Group X-7, MS B257

Los Alamos National Laboratory

Los Alamos, New Mexico 89745

I. INTRODUCTION

Traditional Lagrangian hydrodynamic codes for time dependent, compressible, multimaterial problems in two dimensions use the same general method. A Lagrangian mesh is defined, which moves with the fluid and this mesh defines a set of Lagrangian cells. The mass in each cell remains fixed and the motion of the mesh determines the volume and hence the density of each cell. These methods work well until the mesh becomes distorted due to shear or turbulence. Large distortions cause computer codes to quickly grind to a halt.

The usual solution to distortion is to "rezone" the mesh. Here we move the mesh points artificially so as to reduce distortions and then map the quantities from the old mesh to the new. This results in unwanted diffusion of mass, momentum and energy throughout the mesh. Even with rezoning, few Lagrangian codes can handle more than limited distortions. Recently, what we call "Free-Lagrangian" codes have been developed specifically to handle large distortions. These codes, in addition to adjusting the mesh points, can reconnect mesh points, thus creating new cells. While Free-Lagrangian codes can handle virtually any distortion, they are even more diffusive than rezoners.

We are trying a different approach to the problem. We abandon the idea of Lagrangian cells entirely. In the next section we will discuss how the conservation equations can be solved directly without resorting to Lagrangian cells. Next we will give some examples of calculations using this method. Finally, we will give details of the calculational method presently being used.

II. SOLVING THE CONSERVATION EQUATIONS

The equations we are trying to solve can be written

$$\frac{D}{Dt}\rho = -\rho \vec{\nabla} \cdot \vec{U} \qquad [2.1]$$

$$\frac{D}{Dt}\vec{U} = -\frac{1}{\rho}\vec{\nabla} P \qquad [2.2]$$

$$\frac{De}{Dt} = -\frac{P}{\rho}\vec{\nabla} \cdot \vec{U} \qquad [2.3]$$

$$P = P(\rho, e) \qquad [2.4]$$

where $\vec{U}$ represents the vector velocity, ρ the density, e the specific internal energy and P the pressure of the fluid. Equation [2.1] expresses conservation of mass, [2.2] conservation of momentum and [2.3] conservation of energy. The Lagrangian time derivative, i.e., the derivative following the fluid, is indicated by $\frac{D}{Dt}$.

In a standard Lagrangian calculation only Eq. [2.2], the momentum equation is solved directly. The procedure is to integrate [2.2] over some region of space to arrive at the acceleration of each mesh point. The mesh points are then moved and the new cell volumes along with the fixed cell mass determine the new density, hence, indirectly solving Eq. [2.1]. The associated PdV work term updates the cell energy and indirectly solves Eq. [2.3] and the new pressure is obtained from the equation of state [2.4].

We propose the following: Instead of Lagrangian cells, we think of a set of Lagrangian points which are embedded in and move with the fluid. There is no mass associated with these points. They are just moving tracer points at which we will attempt to keep track of the velocity, density, energy, and pressure of the fluid. In our later example calculations we will show point positions at various times in the calculation. At each of these points, we know the density,

energy and velocity of the fluid, but we do not associate any particular mass with the point.

Looking now at Eq. [2.1], we note that to approximate the time integral of the density change from time t to time t + δt we need an approximation to $\vec{\nabla} \cdot \vec{U}$ at that point. To solve Eq. [2.2], we need an approximation for $\vec{\nabla}P$ and for [2.3] we again need $\vec{\nabla} \cdot \vec{U}$. To obtain these, we select a set of "representative" neighbors. We then make a finite difference approximation to $\vec{\nabla}P$ and $\vec{\nabla} \cdot \vec{U}$, using these neighbors, and update ρ, $\vec{U}$ and e at each point. Each point is then moved the distance $\vec{U}$ δt and one time step is completed.

At the next time step the selection of a set of "representative" neighbors may change, but this does not require any sort of re-mapping of variables. It only means that a different set of points will be used in the next finite difference approximation to $\vec{\nabla} \cdot \vec{U}$ and $\vec{\nabla}P$. Large distortions in the flow will produce frequent changes in neighbor selection, but since there are no cells to distort and no re-mapping to be done the calculation proceeds from cycle to cycle with no difficulty.

III. SOME EXAMPLE CALCULATIONS

3.1 Here we will give three examples of calculations performed by the code HCRO using the free Lagrangian method described herein. In the first test problem, the initial condition is a sphere of perfect gas with a gamma of 5/3. The gas is divided into four regions as seen in Fig. 3.1. Pressures are in megabars, density in gm/cc and dimension in cm.

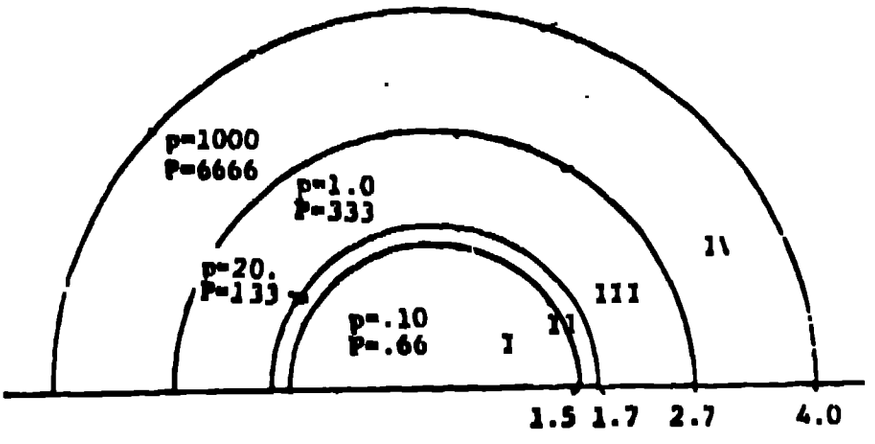


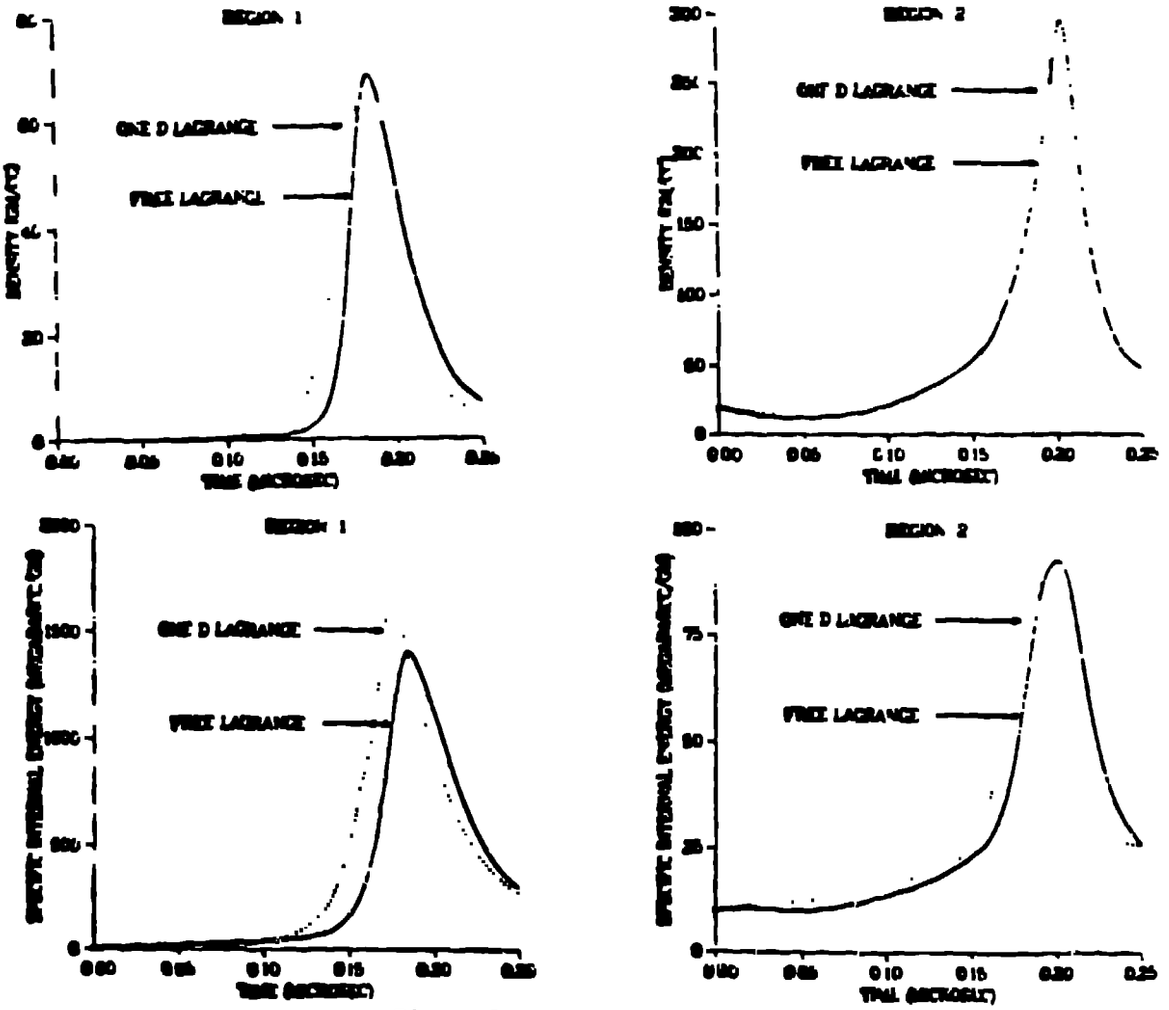
Fig. 3.1

The high pressure in region IV will drive a spherical implosion which will greatly compress region III, II, and particularly I. There are two challenges to this problem, the first is to maintain a spherical ball while running the calculation in cylindrical (r,Z) geometry. Six snapshots of region II are shown in Fig. 3.2. Region I is interior to region II. The minimum volume of region I occurs in the fifth snapshot after which region I begins to expand. We ran 1000 calculational cycles with 73 points in the radial direction and 64 points covering 180° of angle. The left half of the snapshot is a reflection of the right half which was calculated.

Fig. 3.2

The second challenge is the accuracy of the solution. For comparison purposes we ran a standard one-dimension Lagrangian code using 800 zones, 100 zones in each region. In Figs. 3.3.a, b, c, and d. We have plotted the average density and average specific internal energy in regions I and II as calculated by HOBO with 73 points in the radial direction and the one-dimension Lagrangian calculation with 800 points. We feel the agreement to be quite good. One notable difference is the time at which minimum volume is reached. HOBO is slow by about .0075 μsec or 4% of the problem time at that point. Since average density and energy are integral quantities we have plotted one of the variables as a function of radius in Fig. 3.4. We chose radial velocity, but the agreement in all other variables is very similar. The plots are from slightly different times to compensate for the time shift just mentioned. The 1D Lagrange plot is at 2.125 μsec and the HOBO plot is from 2.25 μsec. Apart from the inability of the more coarsely zoned HOBO to resolve the shock front at the radius 1.2 cm we feel the

agreement is excellent. The time chosen for the plot is late in the calculation when region II has expanded almost back to its original volume.



Figs. 3.3.a, b, c, and d

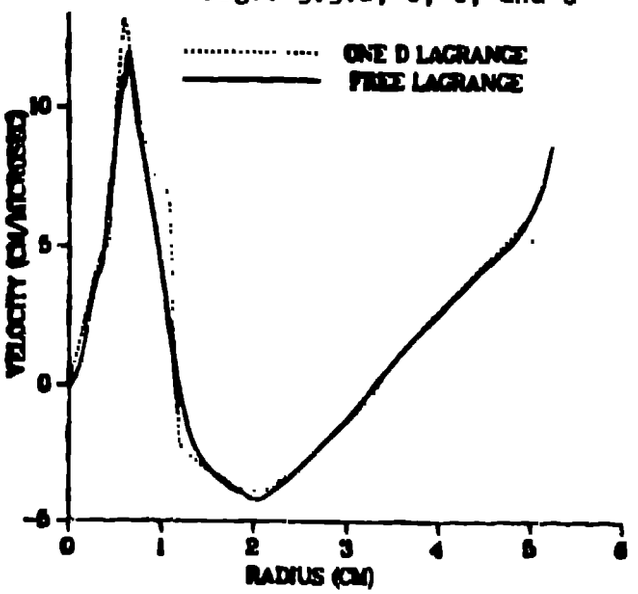


Fig. 3.4

3.2. For our second test problem we have chosen a Meshkov instability based on the geometry used in one of Meshkov's experiments. The initial conditions are shown in Fig. 3.5. A piston driven shock is driven through a region of air and then helium. The air to helium density ratio is just over 7. There is an initial perturbation in the air -- He interface which grows with time after the shock passes through the interface. In Fig. 3.6 we plot several snapshots of the Lagrangian point positions in the air (the He is not plotted). For comparison purposes we ran the same problem in a two-dimensional Eulerian code with the cell size similar to the point separation used in HOBO. In Figs. 3.7.a and b, we compare the size of the perturbation as it grows in time. In 3.7.a the initial perturbation, δ, is .2 cm and in 3.7.b it is .4 cm in width. The agreement between the two codes is excellent.

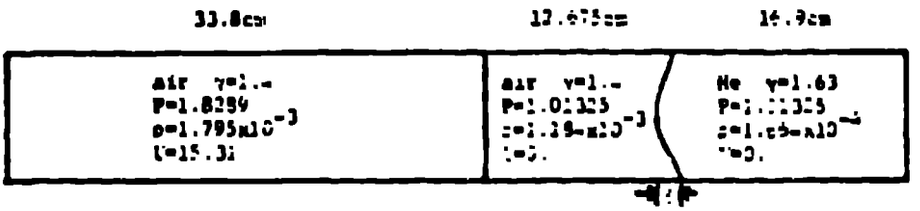


Fig. 3.5

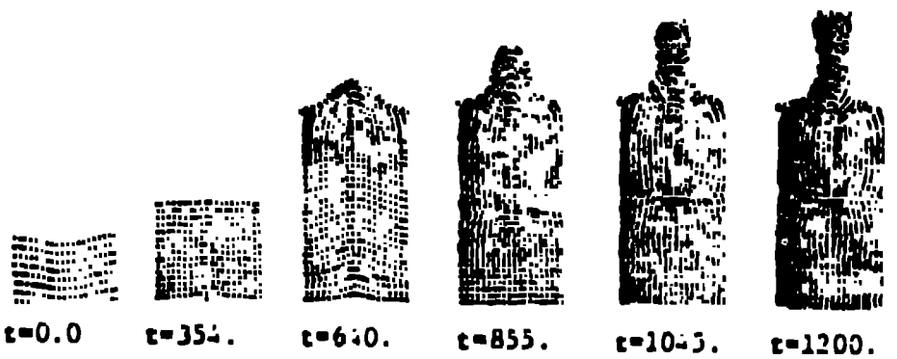
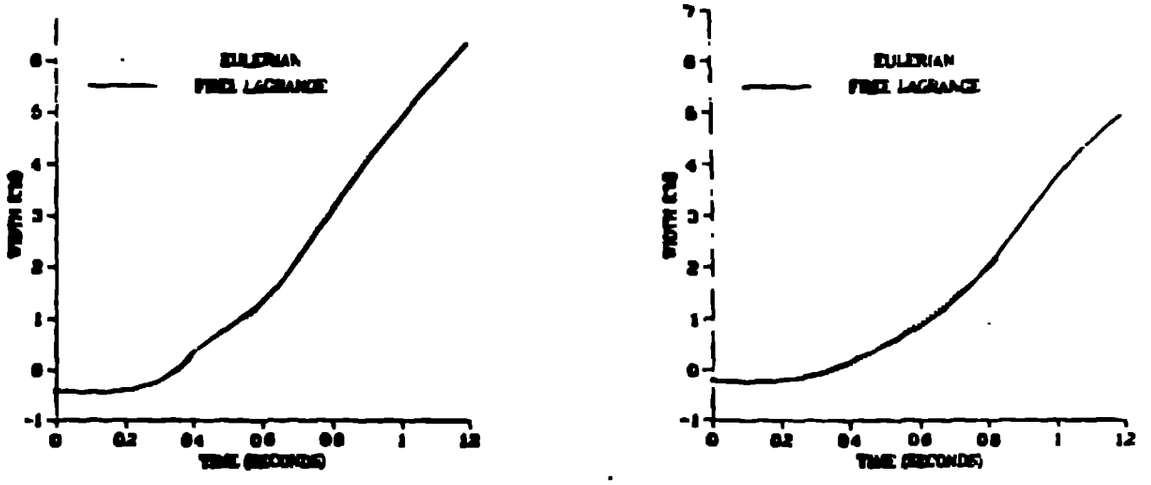


Fig. 3.6



Figs. 3.7.a and b

3.3. Our third test problem is the penetration of a concrete plate by a steel rod moving at an initial velocity of .2134 cm/μsec. The rod is 9.066 cm in diameter and 45 cm in length. The concrete is 50 cm thick. In Fig. 3.8 we show six snapshots of the rod penetrating the concrete. Incompressible theory predicts a constant time rate of change in the length of the steel rod. The sound speed in the rod is .4545 cm/μsec and $(v/c)^2$ = .22, so this problem should not be too far from the incompressible solution. As is shown in Fig. 3.9, the rod length as a function of time matches the incompressible theory very well. Calculations with a two-dimensional Eulerian code produced an almost identical result.

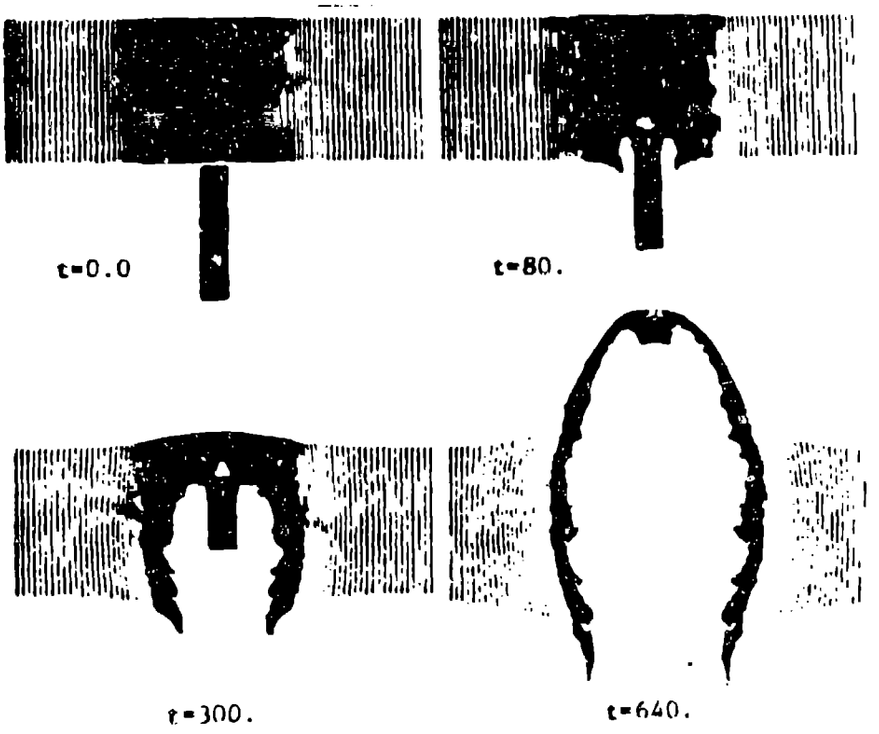

t=0.0
t=80.
t=300.
t=640.

Fig. 3.8

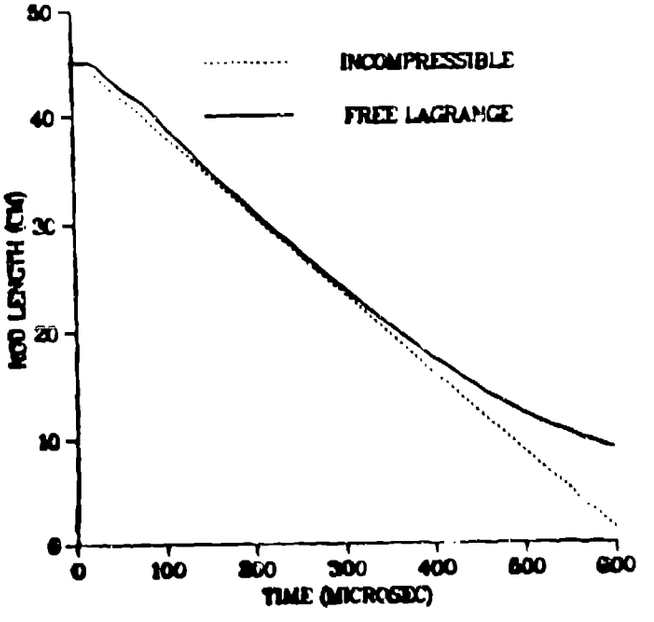

INCOMPRESSIBLE
FREE LAGRANGE
ROD LENGTH (CM)
TIME (MICROSEC)
0
10
20
30
40
50
0
100
200
300
400
500
600

Fig. 3.9

IV. THE FINITE DIFFERENCE SCHEME

4.1 The pressure gradient

We want to approximate $\vec{\nabla}P$ at the point k where neighbors are the points k_1, k_2 ... k_{nmax}. Our neighbor selection guarantees at least three neighbors for each point, the average is six and there is no maximum number. Clearly there are many methods that could be used to approximate $\vec{\nabla}P$. The following was arrived at through much trial and error and appears to work very well.

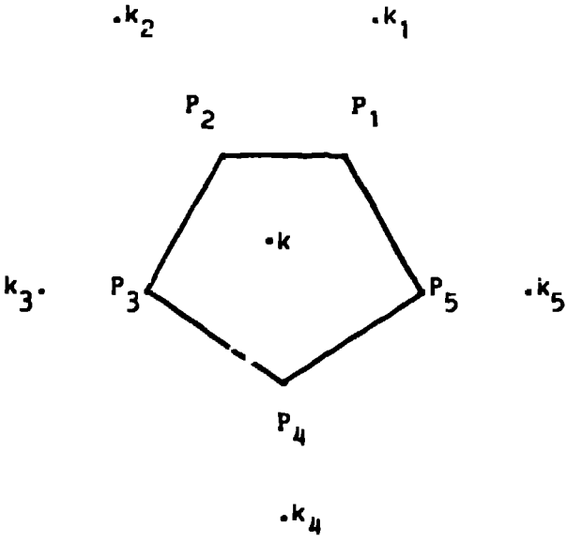


Fig. 4.1

Consider point k in Fig. 4.1 that has five neighbors. We construct a polygon with vertices midway between the point and each of its neighbors. The position of the nth vertex is $\vec{x}_n = 1/2\ [\vec{x}(k) + \vec{x}(k_n)]$ and the vector from $\vec{x}_k$ to $\vec{x}_n$ is denoted by $o\vec{x}_n = \vec{x}_n - \vec{x}_k$. The pressure at the nth vertex, P_n, is a weighted average of P(k) and $P(k_n)$ (to be described in section 4.3). We assume a linear pressure distribution along each edge of the polygon and integrate the pressure over the surface to get a force $\vec{F}$. We assume a constant density ρ_k over the polygon to calculate a mass M. Then we have $\frac{D}{Dt}\vec{U} = \frac{\vec{F}}{M}$. Now let $\vec{x}'_{n2} = \vec{x}_R + \epsilon\partial\vec{x}_n$ and the pressure at the new vertex is $P'_n = P_k + \epsilon(F_n - P_k)$. Now F and M are functions of ϵ and we compute

$$\lim_{\varepsilon \to 0} \frac{F(\varepsilon)}{M(\varepsilon)}.$$

The resulting expression for the pressure gradient is

$$\vec{\nabla} P_k = \frac{\hat{x} \sum_n P_n \left(\delta y_{n-1} - \delta y_{n+1}\right) + \hat{y} \sum_n P_n \left(\delta x_{n+1} - \delta x_{n-1}\right)}{\sum_n \left(\delta x_{n+1}\, \delta y_n - \delta y_{n+1}\, \delta x_n\right)} \qquad [4.1]$$

where $\hat{x}$ and $\hat{y}$ are respectively the unit vectors in the x and y directions and $\delta\vec{x}_n = \delta x_n \hat{x} + \delta y_n \hat{y}$.

If the preceeding is done in cylindrical geometry, the result is identical for $\vec{\nabla}P$ with x and y replaced by r and z. It is of interest to note that if the $\lim_{\varepsilon \to 0}$ is not taken, the result does not give a spherically symmetric pressure gradient in a spherically symmetric problem using cylindrical coordinates.

There is an easier way to arrive at Eq. [4.1] although the method just described is how we originally derived it. Since it takes only three points to describe a plane surface, each consecutive pair of neighbors along with the point k defines a pressure plane to first order. If we assign a weight to each of these approximations we have an approximation for $\vec{\nabla}P$. If the weighting function is the area of the triangle formed by the three points, the result is the same as Eq. [4.1]. We have tried other weighting function, θ and $\sin\theta$ where θ is the angle between $\delta\vec{x}_n$ and $\delta\vec{x}_{n+1}$ both work fairly well, but area weighting appears to be best at this time.

4.2 The divergence of the velocity field

In cartesean coordinates we represent the velocity at the point k by $\vec{U}_k = u_k \hat{x} + v_k \hat{y}$. The divergence of the velocity field can be expressed as $\vec{\nabla} \cdot \vec{U} = \frac{1}{V} \frac{\partial V}{\partial t}$

where V is the specific volume of the fluid. Referring back to Fig. 4.1 the specific volume of the constructed polygon is proportion to the area of the polygon given by

$$A = 1/2 \sum_n (X_{n+1} + X_n)(y_{n+1} - y_n)$$

Hence we can write

$$\vec{\nabla} \cdot \vec{U} = \frac{1}{A} \frac{\partial A}{\partial t} \frac{\sum_n (u_{n+1} + u_n)(y_{n+1} - y_n) + (x_{n+1} + x_n)(v_{n+1} - v_n)}{\sum_n x_{n+1} y_n - y_{n+1} x_n} \qquad [4.2]$$

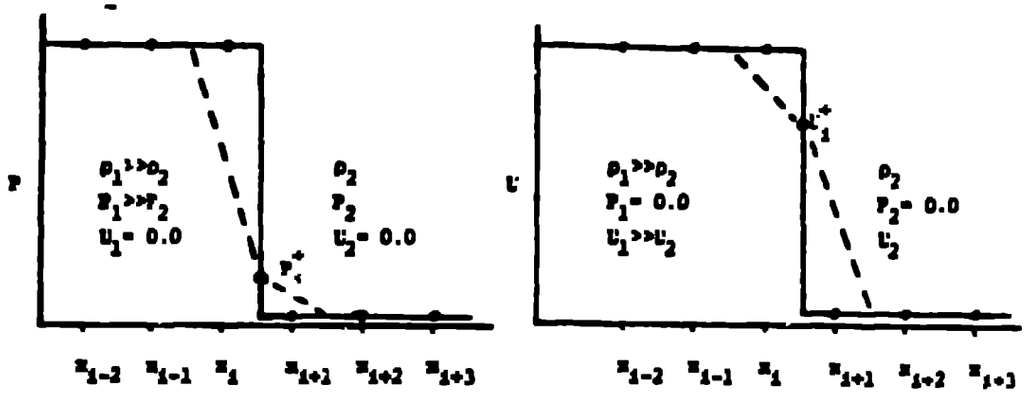
Equation 4.2 can be derived directly from Eq. 4.1 by noting that 4.1 implies a definition for the operators $\frac{\partial}{\partial x}$ and $\frac{\partial}{\partial y}$ and when these are applied to $\vec{\nabla} \cdot \vec{U} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$ Eq. [4.2] is obtained. Thus, we have in effect three ways of deriving the same finite difference approximation to the operators $\frac{\partial}{\partial x}$ and $\frac{\partial}{\partial y}$. In cylindrical coordinates we express the divergence of the velocity field as

$$\vec{\nabla} \cdot \vec{U} = \frac{1}{r} \frac{\partial}{\partial r}(ru) + \frac{\partial v}{\partial z} = \frac{u}{r} + \frac{\partial u}{\partial r} + \frac{\partial v}{\partial z}$$

where $\frac{\partial u}{\partial r} + \frac{\partial v}{\partial z}$ is calculated by Eq. [4.2] with x,y replaced by r,z.

4.3 The midpoint pressure and velocity

In 4.1 we use a pressure P_n which is midway between points k and k_n. This is not a numerical average. Consider the one-dimensional problem depicted in Fig. 4.2.a.



$$P_i^+ = \frac{\rho_{i+1}P_i + \rho_i P_{i+1}}{\rho_i + \rho_{i+1}}$$

Fig. 4.2.a

$$U_i^+ = \frac{(\rho c^2)_i U_i + (\rho c^2)_{i+1} U_{i+1}}{(\rho c^2)_i + (\rho c^2)_{i+1}}$$

Fig. 4.2.b

What pressure should we use for $P_i^+ = P_{i+1}^-$? If we use the average, 1/2 (P_1 + P_2) the acceleration at i+1 will be much greater than at i. However, we know that the velocity should be continuous across the discontinuity. Given equal zoning the boundary pressure which gives equal accelerations to points i and i+1 is $P_i^+ = (P_i\,\rho_{i+1} + P_{i+1}\rho_i)/(\rho_i + \rho_{i+1})$.

It can be shown that the resulting finite difference approximation $\rho_x = (P_i^+ - P_i^-)/\delta x$ is second order accurate when the density is continuous.

Now consider the problem depicted in 4.2.b. Here we have a heavy material on the left moving into a very light material on the right. What should we use for $U_i^+ = U_{i+1}^-$? If we use the average, 1/2 $(U_i + U_{i+1})$, there will be a very large rate of compression in region 2 which is incorrect because region 1 is moving into a near vacuum. The quantity that should be continuous is pressure. The velocity which causes equal pressure increases at points i and i+1 is $U^+ = [(\rho c^2)_i\, u_i + (\rho c^2)_{i+1}\, u_{i+1}]/[(\rho c^2)_i + (\rho c^2)_{i+1}]$. This assumes the sound speed c is a constant. Again it can be shown that the resultant finite difference approximation to U_x is second order accurate if ρc^2 is continuous.

The midpoint pressure used in Eq. [4.1] are inverse density weighted and the midpoint velocities in Eq. [4.2] are ρc^2 weighted.

4.4. The artificial viscosity

An artificial viscosity, q, is added to the midpoint pressure in Eq. [4.1]. It is quadratic inform. Let U_c be the closing rate between points k and k_n, i.e.

$$U_c = (\vec{U}_k - \vec{U}_{k_n}) \cdot \frac{(\vec{X}_r - \vec{X}_{k_n})}{|\vec{X}_k - \vec{X}_{k_n}|}$$

Then let $q_k = a^2 \rho_k U_c^2$ and $q_{k_n} = a^2 \rho_{k_n} U_c^2$. In the spirit of paragraph 4.3, we inverse density weight the two to get our expression for the midpoint q, i. e.,

$$q_n = 2a^2 U_c^2 / (1/\rho_k + 1/\rho_{k_n}) \qquad [4.3]$$

In all of our example calculations in section 2 we used $a^2 = 5.76$. Now we must fold q into the internal energy equation in which we need to evaluate $(P + q)\vec{\nabla}\cdot\vec{U}$. Our approximation for $\vec{\nabla}\cdot\vec{U}$ is given by Eq. [4.2]. The q term is brought inside the summation so that

$$(\rho + q)\vec{\nabla}\cdot\vec{U} = \frac{\sum_n (P_k - q_n) u_n (y_{n-1} - y_{n+1}) + \sum_n (P_k + q_n) v_n (x_{n+1} - x_{n-1})}{\sum_n x_{n+1} y_n - y_{n+1} x_n} \qquad [4.4]$$

4.5. Prevention of density striations

The method so far described has one remaining difficulty. By having all of the variables centered in space it becomes impossible to detect a sawtooth type wave as depicted in one dimension in Fig. 4.3.

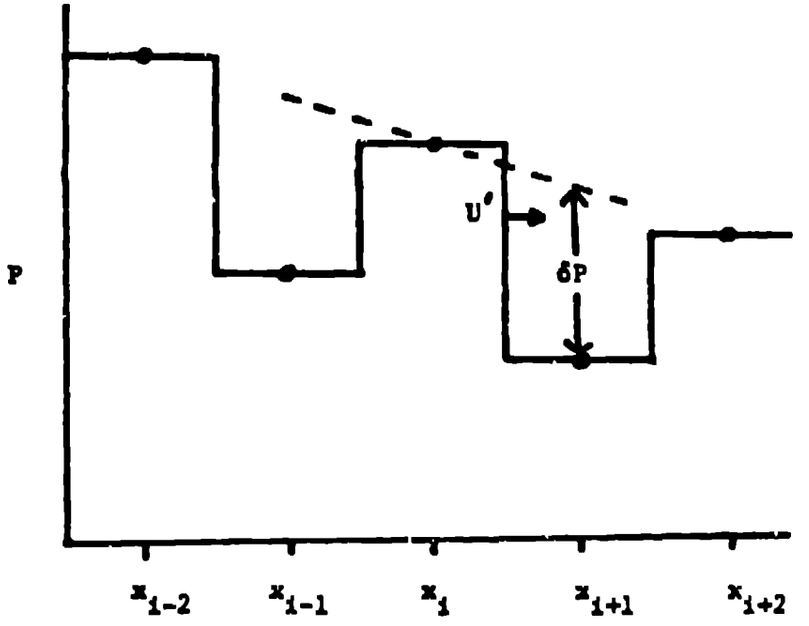


Fig. 4.3

If such a wave develops it cannot be detected by a centered difference scheme. To correct for this, we define an artificial velocity u' as depicted in Fig. 4.3.b. We use our calculated $\vec{\nabla} P$ to extrapolate from point k to point k_n giving $P^{ext}_{k_n} = P_k + (\vec{x}_{k_n} - \vec{x}_k) \cdot \vec{\nabla} P_k$. If the pressure field is linear then $P^{ext}_{k_n} = P_{k_n}$. If they are not equal, there is a second derivative in the pressure field which we attempt to reduce. Physically what should happen is a velocity would be produced at the midpoint as indicated in 4.3.b., which would decompress point i and compress point i+1. This velocity must be proportioned to $\delta P = P^{ext}_{k_n} - P_{k_n}$. We chose to use $u' = b^2 \delta P/\rho c$. We then use ρc^2 weighting between points k and k_n to arrive at

$$u'_n = \frac{b^2 \partial P \,(c_k + c_{k_n})}{\rho_k c_k^2 + \rho_{k_n} c_{k_n}^2} \qquad [4.4]$$

u'_n is added to u_n in calculating $\vec{\nabla}\cdot\vec{U}$.

In our present calculations $b^2 = 1.44$. We further limit $|u'_n|$ to be less than 20% of the maximum of (c_k, c_{k_n}). In practice, u' is a very small term, but an absolutely necessary one. For example, in test problem 1, density striations of around 50% will occur without using u'. We note also that δP is proportional to δP is $\delta x^2 P_{xx}$ and thus is quadratic in nature. The similarity between q and u' is striking. The q is an artificial pressure which smooths the velocity field while u' is an artificial velocity which smooths the pressure field.

4.6 Neighbor selection

The method requires a good selection of representative neighbors at each point in time. We have found out that the neighbors whose bisectors form the Vornoi polygon around the point k are an excellent choice. The k_{th} Vornoi polygon is defined as that region of space which is nearer point k than any other point.

V. SUMMARY

The partial differential Eqs [2.1, 2.2, and 2.3], along with the equation of state 2.4, which describe the time evolution of compressible fluid flow can be solved without the use of a Lagrangian mesh. The method follows embedded fluid points and uses finite difference approximations to $\vec{\nabla}P$ and $\vec{\nabla}\cdot\vec{u}$ to update ρ, $\vec{u}$ and e. We have demonstrating that the method can accurately calculate highly distorted flows without difficulty. The finite difference approximations are not unique, improvements may be found in the near future. The neighbor selection is

not unique, but the one being used at present appears to do an excellent job. The method could be directly tended to three dimensions. One drawback to the method is it's failure to explicitly conserve mass, momentum and energy. In fact, at any given time, the mass is not defined. We must perform and auxially calculation by integrating the density field over space to obtain mass, energy and momentum. However, in all cases where we have done this, we have found the draft in these quantities to be no more than a few present.

References

1. JETP, Vol. 44, No. 2, p.s 424, (1976)

2. G. Birkhoff, E. H. Farantonello, "Jets, Wakes and Cavities," Academic Press (1957)

3. G. Vornoi, J. Reine Angew, Math., 134, 198 (1908)