JOURNAL OF COMPUTATIONAL PHYSICS 73, 349–363 (1987)

A Convective Flux Limiter for Non-Lagrangian Computational Fluid Dynamics

LAWRENCE D. CLOUTMAN*

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

Received January 22, 1986; revised October 10, 1986

The rezoning phase of the ALE method models the advection terms in the Navier–Stokes equations. Consequently, this method suffers from all the numerical difficulties associated with these terms, except when it is used for pure Lagrangian calculations. In particular, the use of any of the differencing options presently available in the CONCHAS-SPRAY computer program other than pure donor cell can create artificial oscillations in the solution through the dispersive truncation errors. This paper describes a flux-limiting procedure that suppresses these oscillations by locally adding enough donor-cell differencing to prevent nonphysical growth of local maxima and minima. Numerical solutions from CONCHAS-SPRAY illustrate the effectiveness of this algorithm. © 1987 Academic Press, Inc

I. INTRODUCTION

This paper describes a convective flux limiter that improves the accuracy and robustness of non-Lagrangian computational fluid dynamics methods, especially the arbitrary Lagrangian–Eulerian (ALE) technique (for example, [1, 2] and references therein), whose roots go back at least as far as the work of Trulio and Trigger [3]. The ALE technique splits each computational time step into a Lagrangian phase and an optional rezoning phase which models advection as the grid is moved in an arbitrary fashion with respect to the fluid. Lagrangian and Eulerian calculations are two special cases in ALE: omission of the rezoning phase results in a Lagrangian calculation, and moving the grid back to its original position results in an Eulerian calculation.

Non-Lagrangian fluid dynamics solutions can suffer from two kinds of truncation errors: diffusive and dispersive. The former artificially smear out a solution, causing a loss of accuracy. Some diffusion is often tolerated, however, because diffusive schemes also tend to be stable. Donor-cell, or upwind, differencing is an example of a highly diffusive scheme. Dispersive errors appear as oscillations in the solution with a wavelength of several cells, and they tend to be worst in low-diffusion

* Now at L-35, Lawrence Livermore National Laboratory, Livermore, CA 94550. Work supported by U.S. Dept. of Energy under contract W-7405-ENG-36. The U.S. Government's right to retain a non-exclusive royalty-free license in and to the copyright covering this paper, for governmental purposes, is acknowledged.

0021-9991/87 $3.00
Copyright © 1987 by Academic Press, Inc.
All rights of reproduction in any form reserved

calculations containing steep gradients, such as interpolated donor cell simulations of shock waves or deflagrations. These errors, too, can cause an unacceptable loss of accuracy. A number of techniques have been developed for reducing these errors, and we now briefly discuss some of these techniques to motivate the development of the new flux limiter.

In principle these errors may be reduced by using a Lagrangian technique. However, this is not possible for most problems of practical interest. Even in one dimension, cells can become so compressed or stretched that accuracy is lost. In two or three dimensions, vortical motions tangle the mesh within a short time. This difficulty has been overcome by Horak *et al.* [4], who run in the Lagrangian mode until the mesh becomes somwhat distorted. Then they introduce a large number of marker particles which are used in a simple counting procedure to map the physical variables from the old distorted mesh to a new regular mesh while conserving mass, momentum, and energy. If it were used every time step, this procedure would be equivalent to donor cell differencing. However, it is used typically every 50 or 100 time steps, so the calculation is almost Lagrangian. One disadvantage of this technique is the "graininess" introduced into the solution by the finite number of particles, which can be overcome only by using a very large number of particles, which in turn requires large computational resources. A second disadvantage is that the method as presently implemented requires the new mesh to consist only of rectangular cells. Otherwise, the method becomes more complex and expensive. Promising continuum versions of this approach have been developed [5, 6], however.

Several other approaches have been tried to improve the accuracy of non-Lagrangian algorithms. Some codes allow the user to select a mixture of differencing techniques. For example, the CONCHAS-SPRAY ALE program [1] allows an arbitrary mixture of centered, donor cell, and interpolated donor cell differencing. The fraction of each can be adjusted on a problem-by-problem basis to obtain optimal accuracy within the constraint that these fractions are constant in space and time. We try to run with pure interpolated donor cell whenever possible because it is second-order accurate in space and time on a uniform mesh. In many cases dispersion errors require the addition of some donor cell to reduce dispersive oscillations to an acceptable level. This approach is simple and more effective than, say, pure donor cell, but it often introduces more diffusion in large parts of the mesh than is actually needed or desired.

A more sophisticated approach was developed by Rivard *et al.* [7] (see also [8, 9]). Called the "truncation error cancellation" technique, the algorithm is based on the use of a weakly unstable scheme, such as centered differencing, with just enough diffusion added locally to provide stability by cancelling the lowest order truncation errors with negative diffusivities. The local diffusion coefficient is determined from the Taylor series expansion of the difference equations, keeping only the lowest order diffusional truncation errors. In regions where the errors provide a positive diffusivity, no diffusion is added. In regions where the numerical diffusivity is negative (thereby causing instability), it is multiplied by -2 and added to the

physical diffusivity. This procedure avoids the global diffusiveness of, for example, donor cell, but it is tedious to implement and sometimes produces significant dispersive errors.

Another approach is the "flux-corrected transport" (FCT) algorithm (for example, [10] and references therein). The philosophy behind the original formulation is in a sense opposite that of the truncation error cancellation algorithm: begin with a diffusive algorithm and then use an antidiffusion stage to remove locally the artificial smoothing without introducing spurious extrema. The original version works well on some one-dimensional test problems with piecewise constant solutions, but is subject to problems with "clipping" and "terracing" errors, especially in two and three dimensions. Van Albada *et al.* [11] show a simple one-dimensional example where the original FCT scheme fails completely. Zalesak's newer version [10] seems to mitigate some of these difficulties, and it is now written in the form of a spatially and temporally dependent mixture of a high order accurate scheme and a low order scheme.

Chapman [12] has developed a somewhat different difference method, which he calls the "FRAM filter." It is based on the use of a high order scheme everywhere except in regions where dispersion errors drive extrema to unphysical values as derived from simple Lagrangian estimates. In those regions he uses a diffusive difference method such as donor cell (although his difference equations are rearranged so the fluxes have the form of a high order convective flux plus a diffusive flux). He uses one difference method or the other in any given cell, not a mixture.

The algorithm we present here makes use of some of the features of the above techniques, especially FRAM. It is especially well suited for methods such as ALE where each cycle is split into a Lagrangian time step followed by a rezoning step. We begin by using interpolated donor cell or some similar method as the basic high order, low diffusion method. In those cells where more diffusion is needed to control the growth of extrema, some fraction of the fluxes are calculated with donor cell differencing. Those cells are identified and the donor cell fraction calculated by comparing the tentative high order solution with the Lagrangian solution from the first phase of the time step. The details of this "flux limiting on extrema" (FLOE) procedure are given in the next section. Section III shows two numerical examples and compares them with solutions obtained by other methods. Section IV contains a summary and conclusions.

II. The Floe Algorithm

The FLOE algorithm is quite general in its applicability. It can be incorporated into most existing programs in a straightforward manner. For Eulerian codes, the required Lagrangian solutions may be obtained in the manner described by Chapman [2]. For some methods, it is possible to simply postpone calculating the advection terms until the end of a time step (in effect performing a Lagrangian time step followed by a rezone step). In particular, the ALE method is already in this

form, and FLOE is easily added to it. As FLOE was first tested in the CONCHAS-SPRAY ALE program [1], the methodology of this program will be used as a concrete example to illustrate the method.

The CONCHAS-SPRAY hydrodynamics program is a recent implementation of ALE that was written primarily to produce numerical solutions of the complete multicomponent Navier–Stokes equations for the conditions found in internal combustion engines, although it is not restricted to those applications. All terms are retained in the governing equations, and these partial differential equations are approximated by partially implicit finite-difference equations. The velocities are defined at the vertices of the arbitrary quadrilateral cells, and all other quantities are defined at cell centers.

The FLOE procedure consists of several distinct steps. The first step is to make a tentative calculation of the transport phase using the selected high order scheme. In CONCHAS-SPRAY, this is a user-specified mixture of centered, donor cell, and interpolated donor cell differencing. For example, the original program transports specific internal energy I by

$$M_0^{n+1} I_0^{n+1} = M_0^L I_0^L + \rho_0^L I_0^L \sum_{i=1}^{4} S_{2i-1} + \sum_{i=1}^{4} S_{2i} \rho_i^L I_i^L, \tag{1}$$

where M_0 is the mass of cell 0, and ρ is the density. The L superscripts denote quantities from the Lagrangian phase, the subscripts on all quantities except the S_i refer to the cell center numbers illustrated in Fig. 1, and the $n+1$ superscript denotes an advanced time quantity. Analogous equations apply to the other cell-centered quantities.

The weighting factors S_i are determined by the finite difference method used. In CONCHAS-SPRAY, each weighting factor S_i is associated with a particular cell center and cell face, as illustrated in Fig. 1. For example, S_1 is the weight for $\rho_0^L I_0^L$

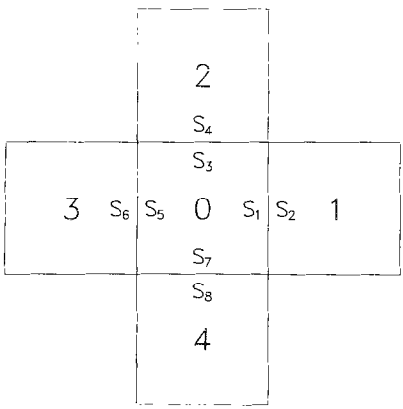

FIG. 1. Schematic drawing of a mesh cell and its neighbors. This diagram illustrates our numbering scheme and shows where the S_i weighting factors are defined.

on the right face of cell 0. We must define several functions needed in the evaluation of the S_i. During the rezoning phase, the movement of the vertices from their positions at the end of the Lagrangian phase to their specified final positions causes each cell face to move through the volume of space bounded by the Lagrangian and final cell faces and by the two lines connecting the Lagrangian and final positions of each of the two cell face endpoints. Let F_R be one half of the volume swept out by the right face of cell 0 during the rezoning. The sign of each of these volumes is determined by the direction the cell face moves. For example, F_R is positive if the cell face moves left during the rezone, and it is negative if it moves to the right. Let F_L be the same function applied to the left face of cell 0. Similarly, F_T is one half the volume swept out by the top face of cell 0, and F_B is the same function applied to the bottom face. For each of these four quantities, we define an additional function, A_X, where X is B, T, L, or R. For example,

$$A_R = \alpha \operatorname{sign}(F_R) + 4\beta F_R/(V_0 + V_1), \tag{2}$$

where V_i is the volume of cell i ($i = 0$ or 1), α is the donor cell fraction, and β is the interpolated donor cell fraction. The fraction of centered differencing is $(1 - \alpha - \beta)$. This same formula is applied to the other three faces by replacing R by B, T, or L. Then the weighting factors are

$$\begin{aligned} S_1 &= F_R(1 - A_R), \\ S_2 &= F_R(1 + A_R), \\ S_3 &= F_T(1 - A_T), \\ S_4 &= F_T(1 + A_T), \\ S_5 &= -F_L(1 + A_L), \\ S_6 &= -F_L(1 - A_L), \\ S_7 &= -F_B(1 + A_B), \end{aligned} \tag{3}$$

and

$$S_8 = -F_B(1 - A_B).$$

Please note that FLOE may be based on a different set of difference methods simply by redefining the S_i as required.

Equations (1)–(3) are used to compute a provisional value of the cell internal energy,

$$(M_0 I_0)' = M_0^L I_0^L + \sum_{i=1}^{4} [S_{2i-1}(\rho_0^L I_0^L) + S_{2i}(\rho_i^L I_i^L)]. \tag{4}$$

For runs in which FLOE is not desired, the provisional values are accepted as the

final values, and this completes the rezoning phase. In general, FLOE will be desired, and it will be necessary to perform the flux limiting step.

The next step is to compare the provisional solution to local limits and then calculate a new value of α for each cell in which the provisional solution falls outside of the limits. Selection of the limits is not a unique process, and even simple prescriptions may be used to good advantage. We can merely require certain quantities to be positive. For example, CONCHAS-SPRAY contains the Robin Hood procedure, which is an *ad hoc* method of stealing mass and energy from neighboring "rich" cells to maintain positivity of species densities and internal energy in cells having a negative value for one or more of those quantities. The Robin Hood algorithm may be approximated by using pure donor cell in those cells in which the provisional value of internal energy or any species density is negative, and using the user-specified values of α and β in all other cells. This procedure will introduce somewhat more diffusion than Robin Hood, and it will not prevent dispersive oscillations altogether, but it will limit their amplitude to the extent that energy and species densities remain positive.

We also tried a more sophisticated prescription that is based upon the observation that advection by the rezoning algorithm should not enhance a local extremum of the Lagrangian solution. Therefore, reasonable choices for the limits are the minimum and maximum values of the Lagrangian quantities in the cells used by the finite difference advection operator. For the internal energy, this choice would be

$$I_{\max} = \max(0, \max_{0 \leqslant i \leqslant 4} (I_i^L)) \tag{5}$$

and

$$I_{\min} = \max(0, \min_{0 \leqslant i \leqslant 4} (I_i^L)). \tag{6}$$

Similar choices would be made for all other variables. Once the choice of limits has been made, the provisional solution is checked to see if it falls within the limits. If it does, no further action on the current cell is taken by FLOE. If one or more limits are violated, we set $\alpha = 1$ and $\beta = 0$ in that cell. This procedure imposes the constraint that $\alpha + \beta \leqslant 1$. This version of FLOE is very similar to FRAM, and we will refer to it as "FRAM" hereinafter.

Another, still more sophisticated, version was tried. Hereinafter the label FLOE will refer to this version only. In cells where one or more limits are violated by the provisional solution, the limiting values become the new desired values of the solution, denoted by an asterisk. That is, I^* is either $I_{\min}$ or $I_{\max}$. The next step is to modify the local values of α and β to obtain the new limited value of the solution. We begin the derivation of the prescription for α by algebraically eliminating the S_i and the A_X from Eq. (4) with Eqs. (2) and (3) to obtain

$$(M_0 I_0)' = M_0^L I_0^L + \sum_{i=1}^{4} G_i(\rho_0^L I_0^L + \rho_i^L I_i^L) + \sum_{i=1}^{4} \left(\frac{4\beta G_i^2}{V_0 + V_i} + \alpha |G_i| \right) (\rho_i^L I_i^L - \rho_0^L I_0^L), \tag{7}$$

where $G_1 = F_R$, $G_2 = F_T$, $G_3 = -F_L$, and $G_4 = -F_B$. A prime denotes the provisional solution. We obtain a new value of α by setting β to zero, replacing $(M_0 I_0)'$ in Eq. (7) by the desired value $M_0^{n+1} I^*$, and solving for the new value of the donor cell fraction, α^*:

$$\alpha^* \sum_{i=1}^{4} |G_i| (\rho_i^L I_i^L - \rho_0^L I_0^L) = M_0^{n+1} I^* - M_0^L I_0^L - \sum_{i=1}^{4} G_i(\rho_0^L I_0^L + \rho_i^L I_i^L). \tag{8}$$

If the sum on the left-hand side is zero, set $\alpha^* = 1$. Otherwise, use Eq. (8) to calculate α^* for the internal energy. Equations analogous to Eq. (8) must also be used to calculate an α^* for each cell-centered variable that violates its limits, and the maximum value of α^* so obtained is used for all quantities in that cell. This value of α^* may be multiplied by a safety factor slightly greater than unity if desired, although we have not found it necessary. Please note that in the program we transport the species densities first so we can always use the latest possible estimates of M_0^{n+1} for transporting other variables.

The final value of α^*, including any possible safety factor, must be restricted to certain limits. It must never exceed unity, and it must never be so small that it produces less diffusion than the original choices of α and β. In our current implementation of FLOE, we use the limits

$$\alpha \leqslant \alpha^* \leqslant 1. \tag{9}$$

Once α^* has been computed for every cell in the mesh, a second sweep of the mesh calculates the S_i for the final fluxing. The value of α^* used to update the A_X on each cell face is the mximum value of α^* in the two cells on either side of the face. We set $\beta^* = 1 - \alpha^*$ for this face, and then Eq. (1) is used to find the final solution. Although only the cell-centered quantities are presently treated with FLOE in CONCHAS-SPRAY, flux limiting should also be added to the velocity fluxing, which is vertex-centered.

Please note that FLOE is not strictly monotonic. This can be seen heuristically by noting that Eq. (8) implicitly assumes that the same α^* will be applied to all four cell faces during the final advection calculation, which will not always be the case. Imposition of a strict monotonicity constraint would require solution of an elliptic problem to properly account for the coupling of neighboring cells. However, our goal is a simple, efficient algorithm that limits the undesirable growth of extrema while reducing the residual numerical diffusion as much as possible.

III. Numerical Examples

In this section we present two examples of applications of FLOE. The first problem is a simple one-dimensional shock tube. The second is numerical simulation of homogeneous propane combustion in an experimental engine. These engine simulations are a subset of the calculations undertaken to compare the calculations to some experimental pressure histories. These examples show why it is desirable to employ a scheme such as FLOE, and they illustrate its effect upon the solution.

The first problem is a simple shock tube with a five-to-one initial jump in the pressure and density 45 cm from the left end of the tube. The gas has $\gamma = 1.4$ and a mean molecular weight of 28.5. The initial temperature is 340 K. The computational cell size is 0.5 cm.

Figure 2 shows the density as a function of position at 0.75 ms. The dotted line shows the initial condition, and solution A is the analytical solution. Solution B through F are numerical solutions that have been shifted vertically by successive increments of 0.1 for clarity. The shock is the discontinuity at the right, and the contact surface is the discontinuity near the middle of the plot. The region of nearly linear variation is a rarefaction wave moving to the left.

The numerical solutions were computed with $\alpha = 0.1$, $\beta = 0.75$, $\nu = 50\ \text{cm}^2/\text{s}$, and an artificial viscous pressure given by

$$q(\mathbf{x}, t) = -0.05\, \rho \delta x^2 (\nabla \cdot \mathbf{u}) \min(0, \nabla \cdot \mathbf{u}) \tag{10}$$

unless otherwise noted. These parameters result in relatively weak numerical

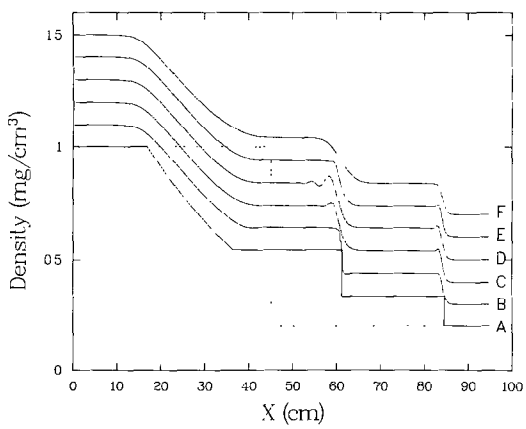

FIG. 2. Density as a function of position at $t = 0.75$ ms in the shock tube problem. The dotted line is the initial condition. Solution A is the analytical solution. Solutions B through F are numerical solutions that have been shifted vertically by successive increments of 0.1 for clarity. Solution B is a Lagrangian solution, solution C is the FLOE solution, solution D is the base case ($\alpha = 0.1$, $\beta = 0.75$, and no flux limiting), solution E is a "FRAM" solution, and solution F is a donor cell solution ($\alpha = 1.0$).

diffusion, but it is adequate for this weak shock (for strong shocks, it might be necessary to use nearly pure donor cell and to increase the leading coefficient of Eq. (10) to 0.25 or more before accurate jump conditions are obtained without FLOE).

Curve B is a Lagrangian solution, and presumably has little numerical diffusion. However, discretization errors must necessarily round off the sharp corners of the rarefaction wave as these points move relative to the gas and, hence, relative to the grid. The artificial viscous pressure plays no role in the rarefaction as q is zero in regions of expansion. The value of ν is likewise too small to have an effect.

Curve C, obtained with FLOE, shows only small oscillations at the shock and contact surface. Because FLOE does not rigorously obey a monotonicity constraint and because the momentum equations do not include FLOE, those oscillations have not been completely suppressed. Curve D is the basic Eulerian calculation with no atempt at flux limiting. Dispersively-driven oscillations severely degrade accuracy behind the contact surface. "FRAM" produced curve E, and the oscillations are completely eliminated. Pure donor-cell differencing ($\alpha = 1$) also eliminates the oscillations, but there is also significant smearing of the contact surface. Interestingly, all calculations produced approximately the same degree of sharpness in the shock. At 0.75 ms, both FLOE and "FRAM" confine their activity to the region between 46 and 84 cm. "FRAM" is active in 10 cells and FLOE in 18. Seven of the FLOE α^*'s are greater than 0.5, with a maximum value of 0.98. Most of the rest are near 0.26.

In some ways, the shock tube is a pathological test problem. Except for the rarefaction (which is a segment of a polynomial well approximated by a straight line), the solution is piecewise constant. All but the worst numerical methods will model a straight line accurately. On the other hand, finite differences tend to have difficulties at large gradients and curvatures. The four points separating the five linear or nearly linear segments have infinite high-order derivatives. Thus, even if one works very hard to do a reasonable job of modeling the solution near the discontinuities, there is no guarantee that an accurate solution away from the discontinuities means the method is accurate or robust for other classes of problems.

In the second example, the engine is the Sandia National Laboratories' experimental DISC engine used with homogeneous propane–air mixtures and an unshrouded valve (to prevent swirl). It has a right circular cylindrical combustion chamber with a 7.65 cm bore, a 1.8 cm clearance height at top dead center (TDC). and a compression ratio of 5.5. The present results are limited to 1203 rpm and stoichiometric mixtures. The spark plug was loccated at the center of the cylinder head. A more complete description of the engine is given by Smith [13].

The computer program is nearly the same as that reported by Cloutman *et al.* [1] except that the turbulent Schmidt and Prandtl numbers were decreased to $\frac{1}{3}$ and the eddy diffusivity is calculated from a subgrid scale (SGS) turbulence kinetic energy density which is in turn a numerical solution of a transport equation [14]. The solution algorithm for this transport equation included FLOE. We note that in principle, SGS is valid only in three dimensions. However, its use in these

exploratory two-dimensional engine simulations produces accuracy sufficient for our present purpose of illustrating the use of FLOE.

Figure 3 shows pressure as a function of crank angle for three different flux limiting options plus cycle-averaged experimental data. The dot-dash line represents the $\beta = 1$ solution with no flux limiting. The dashed line is the Robin Hood solution. The solid line represents the FLOE solution using the reaction rate and turbulence parameters listed above. The mesh is 30 by 15 cells at top dead center ($\theta = 360°$). The cells are approximately 1.27 mm on each side at all times, with rows of cells added or deleted as necessary to maintain this resolution as the piston moves. The cells remain rectangular but change axial size even on cycles when the number of rows is not changed, resulting in an example of a non-Lagrangian, non-Eulerian calculation. Runs with $\frac{2}{3}$ as many cells in each direction give essentially the same results. These axisymmetric solutions are ignited at the experimental ignition time, 10° before top dead center, by dumping 40 mJ of heat into the upper leftmost cell over a period of 50 μs. This is approximately the physical amount of energy discharged through the spark gap. The cell is a disk 2.54 mm in diameter and 1.27 mm thick, which is nearly the actual spark plug gap volume, and it produces a realistic temperature rise. The rate of pressure rise is very small at first since the flame is small. The agreement with the cycle-averaged experimental results (the dotted line in Fig. 3) can be made reasonable in the two non-FLOE cases until approximately half of the fuel is burned. The experimental curve then shows a point of inflection, and the pressurization rate decreases thereafter. The calculated pressurization rate keeps increasing until the fuel is consumed. The SGS calculation simulates an individual engine cycle, not a cycle average. Examination of several individual experimental pressure traces shows that the pressure peaks are indeed somewhat sharper than the cycle average, but not enough to bring the calculations into

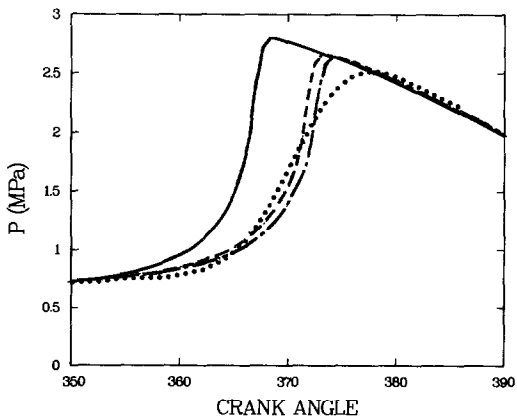

FIG. 3. Pressure versus crank angle from ignition through the end of combustion. The solid curve is the FLOE result, the dashed curve is from interpolated donor cell with no smoothing, the dot-dash curve is the Robin Hood result, and the dotted curve is the cycle-averaged experimental result.

agreement with experiment. Except for the top third or so of the pressure peak, the calculation agrees with the measurements to within the cyclic variation. Exploratory calculations with an improved turbulent reaction rate model [15] show a reduced discrepancy at the pressure peak.

The flame speed in the FLOE solution is significantly faster than in the non-FLOE solutions. Unfortunately, this effect is caused by the inextricably combined numerical and physical effects that determine the numerical flame speed. For a laminar flame, the flame speed is proportional to the square root of the product of the thermal diffusivity and the reaction rate. Our model treats turbulent flames as thickened laminar flames, and this scaling law holds approximately. However, our computational zones are larger than the physical flame thickness, so numerical effects (including Eulerian diffusion) can strongly affect the calculated flame speed. Indeed, the chemistry parameters were chosen by forcing the pressure peak to occur at nearly the correct time for a fixed set of turbulence parameters in the pure $\beta = 1$ case. Robin Hood adds a little extra numerical diffusion, so the flame is a little bit faster. FLOE adds even more diffusion, and the flame is faster yet. By holding the turbulence and chemistry parameters fixed, we can use the pressure-time histories to provide a rough comparison of the amounts of numerical diffusion from the various flux limiting algorithms.

Figure 4 also shows pressure histories for pure donor cell (dashed curve) and "FRAM" (dot-dashed line) runs. Since the donor cell numerical diffusivity can be several times the SGS viscosity, the flame speed is dramatically increased. It is possible to reduce the reaction rate to produce the correct flame speed, but there is no point to doing so because the effects of the SGS turbulence model are swamped by the numerical diffusivity. Therefore, donor cell differencing is unacceptable. "FRAM" concentrates donor cell differencing along the flame front. This procedure

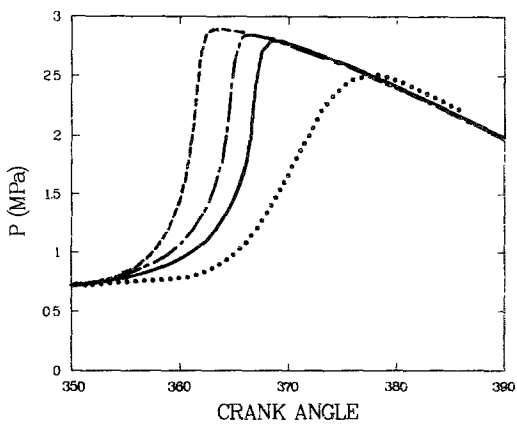

FIG. 4. Pressure versus crank angle from ignition through the end of combustion. The solid curve is the FLOE result, the dashed curve is the donor cell result, the dot-dash curve is the "FRAM" result, and the dotted curve is the cycle-averaged experimental result.

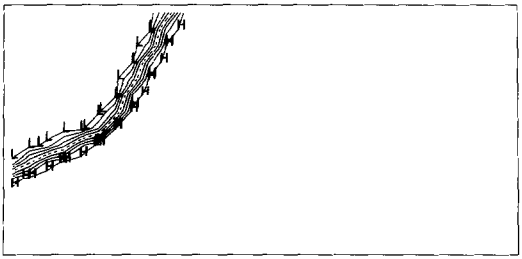

FIG. 5. Contours of fuel mass fraction 5.8° before top dead center. The minimum and maximum values are 1.08×10^{-12} and 4.38×10^{-2}, respectively. The L and H contour values are 4.37×10^{-3} and 3.93×10^{-2}, respectively. This solution is typical of those variables that are essentially step functions with the step at the flame front.

introduces less difussion than pure donor cell (although it is still significant) since the flame speed lies between the donor cell and FLOE flame speeds. Because the high diffusion rate is now confined to the region of the flame where the turbulence model is most unreliable, it is not clear that "FRAM" should be dismissed on grounds of producing too much diffusion.

Figure 5 shows contours of fuel mass fraction 5.8° before top dead center when the mass of fuel has been reduced from 14.45 to 14.26 mg. The solution is essentially a step function with the step at the flame front. Figure 6 shows mass fraction contours of CO, which shows the species is concentrated in the upper left corner, decreasing smoothly in abundance out to the flame front. There is a jump in the CO concentration at the flame front, but it is less severe than for the fuel. All species mass fraction distributions are either the step function typified by the fuel in Fig. 5 or the smoother distribution typified by CO.

Table I illustrates the phenomenon that prompted us to investigate flux limiters: the minimum fuel mass fraction is negative for the unmodified interpolated donor cell ($\beta = 1$) solution. This is caused by a small undershoot near the flame front due

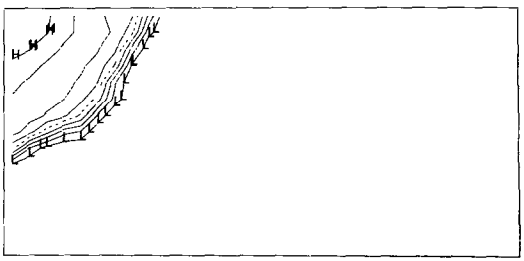

FIG. 6. Contours of CO mass fraction 5.8° before top dead center. The minimum and maximum values are 4.09×10^{-57} and 5.16×10^{-3}, respectively. The L and H contour values are 5.16×10^{-4} and 4.64×10^{-3}, respectively. This solution is typical of those species with a high concentration at the point of ignition and a gradually decreasing abundance as the flame front is approached.

TABLE I

Minimum and Maximum C_3H_8 and OH Mass Fractions with 14.37 mg of Fuel Remaining

	C_3H_8	OH
A. $\beta = 1.0$	−4.59E−04	−4.41E−07
	4.37E−02	1.02E−03
B. "FRAM"	3.69E−09	3.31E−72
	4.37E−02	1.00E−03
C. Donor Cell	1.73E−09	1.12E−63
	4.37E−02	9.38E−04
D. Robin Hood	8.80E−11	0.00E+00
	4.37E−02	9.77E−04
E. FLOE	1.54E−09	4.62E−72
	4.37E−02	9.86E−04

to dispersive truncation errors. While tiny negative numbers can be tolerated and treated as zeroes, the magnitude of this minimum value is fully one percent of the maximum value. While this situation is probably not disastrous, it is certainly not desirable.

The remaining entries in Table I show the effects of the various techniques for controlling dispersive oscillations. The table illustrates two important points. First, all four techniques help control the negative species mass fractions. Second, the physical minimum fuel and OH mass fractions are zero, and the calculated minima are at least seven orders of magnitude below the associated maxima. With the

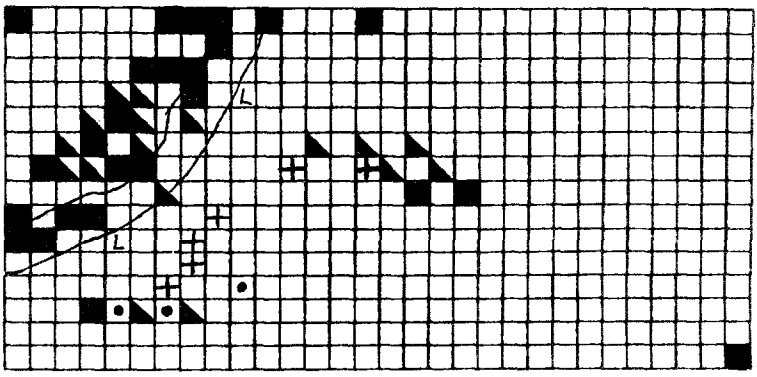

FIG. 7. Computational grid 5.8° before top dead center showing those cells with FLOE active. Cells with a bullet have $\alpha^* < 0.01$, those with a plus have $0.01 < \alpha^* < 0.1$, those that are half shaded have $0.1 < \alpha^* < 0.5$, and those that are fully shaded have $\alpha^* > 0.5$. The contour lines are isotherms and indicate the flame shape and location. The L isotherm corresponds to 804 K, and the other isotherm corresponds to 2079 K.

Courant number limited to 0.2 by stability considerations, donor cell differencing in two dimensions is guaranteed to produce positive mass fractions. This behavior is in fact exhibited in the donor cell species solutions at all times. "FRAM" shows the same behavior, although there is not a similar guarantee of positivity. The *ad hoc* Robin Hood procedure is quite successful at preventing negative mass fractions. The FLOE donor cell fraction α^* is typically well under unity and mostly confined to the neighborhood of the flame front, as we shall see in Fig. 7. This is sufficient to preserve positivity of fuel mass fraction in this calculation, but in a less *ad hoc* manner than Robin Hood and with less diffusion than "FRAM" and donor cell. The performance of the various techniques on the OH mass fraction is the same as for the fuel.

Figure 7 shows the computational grid 5.8° before top dead center with cells shaded where FLOE is active. Cells with a bullet have $\alpha^* < 0.01$, those with a plus have $0.01 < a^* < 0.1$, those that are half shaded have $0.1 < \alpha^* < 0.5$, and those that are fully shaded have $\alpha^* > 0.5$. Isotherms are also shown to mark the flame front. The L isotherm corresponds to 804 K, and the other isotherm corresponds to 2079 K. Of the 450 cells, FLOE is active in 51. Of these, three have $\alpha^* < 0.01$, six have $0.01 < \alpha^* < 0.1$, 17 have $0.1 < \alpha^* < 0.5$, and 25 have $\alpha^* > 0.5$. Only four are nearly pure donor cell with $\alpha^* > 0.95$. Of course, more than 51 cells use enhanced donor cell fractions on at least one face since the value of α used to compute the fluxing factors for any face is the maximum α^* on either side of the face. The highest diffusion rates are largely confined to the immediate post-flame gases and are somewhat smaller than with "FRAM." Both "FRAM" and FLOE are definite improvements over pure donor cell, which puts massive amounts of diffusion even in places where it is not needed.

IV. Summary and Conclusions

The FLOE technique is a relatively simple and effective advective flux limiter that alleviates spurious oscillations near steep gradients. In one numerical example, the flame in a homogeneous charge engine, the FLOE algorithm allowed us to run with interpolated donor cell differencing while suppressing the small overshoots and undershoots near the flame that tend to produce negative densities in the trace species. The added numerical diffusion noticeably increased the flame speed. Unfortunately, this increase indicates that numerical diffusion is significant compared to the eddy viscosity in the neighborhood of the flame front. Similarly, FLOE was effective in reducing dispersive errors at the contact surface in the shock tube problem, and it was marginally effective at the shock front.

FLOE had little impact on the running speed of CONCHAS-SPRAY. The original advection subroutine used approximately one percent of the computational time. FLOE approximately doubles the amount of arithmetic in the advection

routine, so it has little effect on the overall run time, even on scalar machines. On a Cray Research, Inc. computer, the routine vectorized naturally, offsetting the extra arithmetic.

Acknowledgments

This study was begun at the Combustion Research Facility of Sandia National Laboratories, Livermore, during a change of duty station assignment. I would like to thank T. M. Dyer and W. C. Robinson for their hospitality and for providing a stimulating atmosphere. I also thank J. R. Smith and P. O. Witze for their help in setting up the engine test problem and for access to their experimental data. I am grateful to T. D. Butler, L. G. Margolin, and J. D. Ramshaw for their helpful comments on the manuscript.

References

1. L. D. Cloutman, J. K. Dukowicz, J. D. Ramshaw, and A. A. Amsden, "CONCHAS-SPRAY: A Computer Code for Reactive Flows with Fuel Sprays," Los Alamos National Laboratory Report LA-9294-MS, Los Alamos, NM, 1982.
2. W. E. Pracht, *J. Comput. Phys.* **17**, 132 (1975).
3. J. G. Trulio and K. R. Trigger, "Numerical Solution of the One-Dimensional Hydrodynamic Equations in an Arbitrary Time-Dependent Coordinate System," Lawrence Radiation Laboratory Report UCRL-6522, Berkeley, CA, 1961.
4. H. G. Horak, E. M. Jones, J. W. Kodis, and M. T. Sandford, II, *J. Comput. Phys.* **26**, 277 (1978).
5. J. K. Dukowicz, *J. Comput. Phys.* **54**, 411 (1984).
6. J. D. Ramshaw, *J. Comput. Phys.* **59**, 193 (1985).
7. W. C. Rivard, O. A. Farmer, T. D. Butler, and P. J. O'Rourke, "A Method for Increased Accuracy in Eulerian Fluid Dynamics Calculations," Los Alamos National Laboratory Report LA-5426-MS, Los Alamos, NM, 1973.
8. L. D. Cloutman and L. W. Fullerton, *J. Comput. Phys.* **29**, 141 (1978).
9. L. D. Cloutman, *Astrophys. J.* **227**, 614 (1979).
10. S. T. Zalesak, *J. Comput. Phys.* **31**, 335 (1979).
11. G. D. van Albada, B. van Leer, and W. W. Roberts, Jr., *Astron. Astrophys.* **108**, 76 (1982).
12. M. Chapman, *J. Comput. Phys.* **44**, 84 (1981).
13. J. R. Smith, "Turbulent Flame Structure in a Homogeneous-Charge Engine," SAE Paper 820043, Detroit, MI, 1982.
14. A. A. Amsden, J. D. Ramshaw, P. J. O'Rourke, and J. K. Dukowicz, "KIVA: A Computer Program for Two- and Three-Dimensional Fluid Flows with Chemical Reactions and Fuel Sprays," Los Alamos National Laboratory Report LA-10245-MS, Los Alamos, NM, 1985.
15. J. Abraham, F. V. Bracco, and R. D. Reitz, *Combust. Flame* **60**, 309 (1985).