30
1/22/86 JMD ① I-24581 1-10-86

SANDIA REPORT SAND85—0613 • Unlimited Release • UC—32
Printed January 1986

⑦ DR-1496-6

A Two-Dimensional Flux-Corrected Transport Solver for Convectively Dominated Flows

M. R. Baer, R. J. Gross

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-76DP00789

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof or any of their contractors or subcontractors.

Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

NTIS price codes
Printed copy: A05
Microfiche copy: A01

SAND--85-0613

DE86 005329

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Distribution
Category UC-32

SAND-85-0613

A Two-Dimensional Flux-Corrected Transport Solver for Convectively Dominated Flows

M. R. Baer and R. J. Gross
Fluid Mechanics and Heat Transfer Division 1513
Sandia National Laboratories

Abstract

A numerical technique designed to solve a wide class of convectively dominated flow problems is presented. An attractive feature of the technique is its ability to resolve the behavior of field quantities possessing large gradients and/or shocks. The method is a finite-difference technique known as flux-corrected transport (FCT) that maintains four important numerical considerations - stability, accuracy, monotonicity, and conservation. The theory and methodology of two-dimensional FCT is presented. The method is applied in demonstrative example calculations of a 2-D Riemann problem with known exact solutions and to the Euler equations in a study of classical Rayleigh-Taylor and Kelvin-Helmholtz instability problems. The FCT solver has been vectorized for execution on the Cray 1S - a typical call with a 50 by 50 mesh requires about 0.00428 cpu seconds of execution time per call to the routine. Additionally, we have maintained a modular structure for the solver that eases its implementation. Fortran listings of two versions of the 2-D FCT solvers are appended with a driver main program illustrating the call sequence for the modules.

(This page left blank intentionally.)

Contents

(This page left blank intentionally.)

1 Introduction

Many fluid mechanics and heat transfer processes are dominated by nonlinear convective phenomena. Such effects often produce flows which exhibit extreme gradients and/or shocks. Following von Neumann and Richtmyer's important discovery of the artificial viscosity method [1], continued research in computational fluid mechanics has been directed towards properly treating convective terms. Much of the current work has focused on methods based on several important mathematical issues. Godunov [2] was among the first to recognize the importance of positivity of convective equations, and most computationalists now affirm the superiority of methods which preserve positivity [3]. [1] Additionally, the integral conservation relations of the continuum equations have been recognized and widely used in many numerical techniques [4]. Finally, the inaccuracies of first-order methods, especially in regions of large gradients, have been studied in some depth [5]. Thus, investigations into numerical techniques for the solution of convectively-dominated problems have converged on the development of second- (and higher) order accurate numerical methods which are both conservative and monotone.

In one dimension, numerical solutions of convectively-dominated flows with adequate stability, accuracy, monotonicity and conservation are now commonplace. Current efforts are aimed towards applying these techniques in multidimensional problems where aspects such as shear and geometry effects are manifested. This paper presents a numerical technique which addresses the solution of convectively-dominated flows in two dimensions.

The large majority of *general* solvers for this class of problems utilize finite-difference techniques. Presently, finite-difference techniques appear to have an edge in accuracy, computational efficiency, and storage over other approaches. For certain problems, the moving finite element method discussed by Gelinas *et. al.* [6] has been demonstrated to be extremely accurate. However, to our knowledge, this technique has had little application to multidimensional problems, it is relatively computationally inefficient, and requires user-specified parameters that are problem-dependent. Thus, following the current direction in solving convective problems, our discussion shall focus on finite-difference methods. Furthermore, we choose to concentrate on the solution of a system of Eulerian conservation equations.

All finite-difference schemes exhibit numerical diffusion (dissipation) and/or dispersion error due to truncation. It is this fact which causes great difficulty in numerically describing convectively-dominated flows — especially flows which

[1] Positivity implies that if a property of the system is positive at some instant in time, then it must remain positive at all later times also. For example, if the field represents the density or energy of a material, negative values are not physically meaningful.

exhibit shock-type regions. In early techniques, the differential equations were replaced by Taylor series expansions that yield algebraic equations endowed with the desirable characteristics of monotonicity, conservation, high-order, and stability, although all four characteristics could not be concurrent in one method. In fact, Godunov [2] proved that there were no linear second- or higher-order schemes that could always guarantee monotonicity. In Van Leer's [5] study of conservative finite-difference methods, a generalized theory for hyperbolic systems of conservation laws was established which unified many schemes such as the methods of Lax [7], Lax-Wendroff [8], Rusanov [9] and Godunov [2]. Van Leer also recognized that only the first-order accurate methods are monotone and stable. However, these low-order methods gained their stability by being strongly dissipative, and often caused drastic smoothing of the solution. Thus, first-order accurate methods such as Lax and upwind differencing [10] yield solutions with an extreme lack of detail. On the other hand, linear second-order methods, such as the Lax-Wendroff and MacCormack [11] techniques are known to be weakly dissipative and susceptible to nonlinear instabilities [5]. In regions of high gradients and/or shocks, unphysical oscillations occur which excite false modes of behavior in the physical system. For example, enforced conservation under such false modes can cause a positive definite field to become negative. Introducing artificial diffusion into the numerical scheme has been a traditional approach to dampen spurious oscillations in regions of large gradients [1,12], that is, numerical dissipation is directly added to a scheme so that shocks and discontinuities are spread over distances which can be resolved on a practical computational mesh. This is, at best, a compromise approach. If the oscillations are totally damped, the solution exhibits the same massive diffusion as the linear first-order techniques. On the other hand, if the oscillations are not totally damped, then nothing has been gained over standard high-order methods.

In the discussion to this point, the first-order methods, higher-order methods, and the artificial diffusion approach were *linear* methods. That is, the same operations are applied to every finite-difference point during every timestep. Van Leer [13] and Boris and Book [14] were among the first to recognize that if accuracy *and* monotonicity are desired, *nonlinear* techniques are necessary. By allowing the numerical diffusion to be nonlinear in space and time, the severe limitation of large diffusion could be avoided in smooth regions and yet be incorporated locally near steep gradients to ensure monotonicity. These "positivity-preserving" methods as categorized in a recent review by Woodward and Colella [15], fall into two general classes: (1) linear hybridization, and (2) those based on Godunov's nonlinear technique. The recent appearance of Total Variational Diminishing (TVD) techniques, which are also positivity preserving [16], introduces yet a third category to this classification system.

A high-order, nonlinear scheme based on the Godunov method evolved from the studies by van Leer [13,17,18,19,20]. In Godunov's method, the nonlinearity is introduced explicitly into the method using exact solutions to the Riemann problem. Instead of building a solution by piecing together smooth, small-amplitude solutions, the solution is formed using discontinuous functions. However, this approach is the most complex of the monotonocity-preserving techniques. Another drawback is that it requires the construction of characteristics of the hyperbolic system of equations which can become complex especially if a nonideal gas law is used. Roe [21,22] and Woodward and Colella [15,23] are among the major contributors to this approach. The PPM (Piecewise-Parabolic Method) scheme of Woodward and Colella is an example utilizing this technique [23].

In linear hybridization, the results of a low-order scheme are blended with a high-order scheme to take advantage of the strong points of each. In smooth regions (regions with small gradients) the high-order scheme is used exclusively, assuring proper accuracy. Near regions of large gradients, however, the low-order scheme is weighted with the high-order scheme to an extent sufficient to maintain monotonicity. Currently, there are a number of numerical techniques based on the linear hybridization principle, such as the work of Boris and Book [14], Chapman [24], and Forester [25].

The TVD schemes represent an effort to place the study of hyperbolic systems of equations on a rigorous mathematical foundation. A partial list of contributors to this effort include Harten [16], Davis [26], Yee *et. al.* [27], Sweby [28] and Osher and Chakravarthy [29,30]. Although many of the TVD techniques developed utilize Riemann solvers and as such can be classified as Godunov-type approaches, the work of Davis is analogous to the linear hybridization approach.

In the present work, we focus on one of the first linear hybridization techniques to be developed - the flux-corrected transport (FCT) method devised by Boris and Book [14]. FCT is a well-founded and widely used monotonicity-preserving method. It has found use in a wide diversity of topics such as shallow water instability problems [31], plasma physics [32,33], transport of a barium cloud [34], atmospheric physics [35], two-phase flow problems [36], and recently in detonation problems [36,37,38]. In [39,40], FCT has been extensively generalized. Zalesak [41] reformulated the original FCT method in terms of a linear hybridization and developed a truly two-dimensional algorithm which maintains monotonicity. Early FCT algorithms were known to "terrace" the solution in regions of large gradients which was attributed to the specific nonlinear action of the monotonicity constraint on dispersive errors. Recent work by Book and Fry [42] and Zalesak [43] have proposed extensions of the FCT technique to deal with this problem. The present work incorporates the improvements discussed in [42].

In this study, we devise a vectorized version of FCT to solve two-dimensional transport-dominated fluids/heat transfer problems. As a benchmarking example, we consider two-dimensional Riemann problems which have known exact solutions [44]. Also, we examine several classical fluids problems which describe instabilities of the Kelvin-Helmholtz and Rayleigh-Taylor varieties. The instability problems were studied to show that the numerical stability and monotonicity of FCT does not preclude the description of a flow field which is physically unstable.

The remainder of the paper is organized as follows. First, we discuss the mathematical foundation of FCT. Then, the comparison of FCT to the exact 2-D Riemann problem for several different initial conditions is presented. In the next section, FCT is applied to the field equations for a single phase ideal compressible fluid to study flow instabilites. Finally, we provide a guide in the use of the FCT solution modules and include Fortran listing of these routines in an Appendix.

2 Two-Dimensional Flux-Corrected Transport

In this section, an FCT method is developed which incorporates the four important numerical considerations – stability, accuracy, monotonicity, and conservation – for solving two-dimensional, convectively-dominated flows with regions of large gradients. The generalized 2-D transport equation of interest in Eulerian form is:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho v) = C(x,y)\frac{\partial f}{\partial x} + D(x,y)\frac{\partial g}{\partial y} + S(x,y) \tag{1}$$

where ρ is any dependent property of the flow, *e.g.* density, momentum, or energy, x and y are the two spatial variables, t is time, and u and v are the velocities in the x and y directions, respectively. The terms on the right-hand side represent flux and algebraic source terms such as drag, heat transfer between two phases, combustion sources, volumetric heat input, etc.

We shall closely follow the 1-D FCT development given in [3] and [45], and also incorporate the approach in [42] which allows the method to have either sixth-order accurate amplitude error or sixth-order accurate phase error. Figure 1 shows the two-dimensional computational grid used by the algorithm; the points in the center of the indicated cells are the finite-difference grid points. The interfacial areas $A_{i,j}$, $B_{i,j}$ and the volumes $\Lambda_{i,j}$ are also depicted. Note that the geometry is generalized for a nonuniform mesh. However, if the mesh is uniform, then the interfacial areas and the volumes become $A_{i,j} = \Delta y$; $B_{i,j} = \Delta x$ and $\Lambda_{i,j} = \Delta x \Delta y$. Also, if the interfacial areas and volumes are properly chosen, then orthogonal systems other than the Cartesian system depicted in Figure 1 can be used.

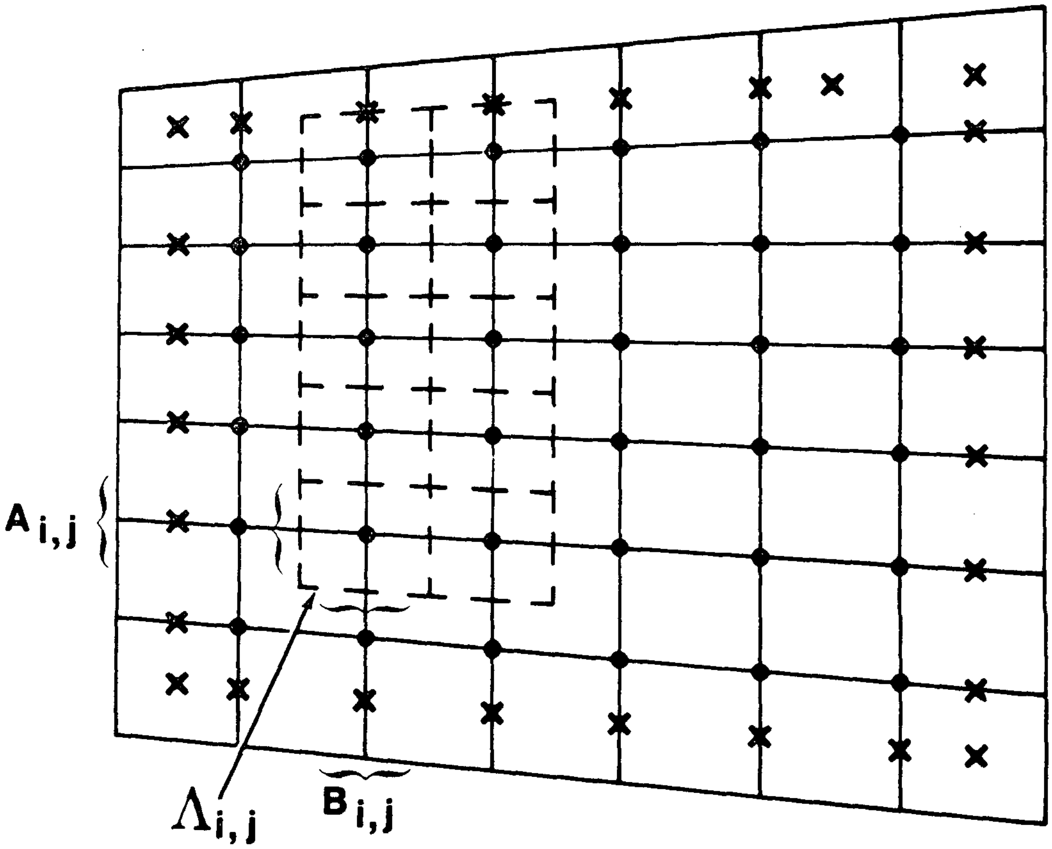

Figure 1. The two-dimensional grid used by the FCT routines. "X's" denote boundary points while all interior points are denoted by dots. The areas, $A_{i,j}$ and $B_{i,j}$, in each direction, and the volumes, $\Lambda_{i,j}$, of some of the interior computational cells are outlines by the dashed lines.

The initial field of values for the dependent variable ρ is denoted by "ρ^{c}". The first step in the FCT method is to convectively transport the ρ^{c} field according to

$$\begin{aligned}\Lambda_{i,j}\,\rho^{\cdot}_{i,j} = \;& \Lambda_{i,j}\,\rho^{c}_{i,j} - \delta t\,\rho^{c}_{i+\frac{1}{2},j}\,A_{i+\frac{1}{2},j}\,u_{i+\frac{1}{2},j} + \delta t\,\rho^{c}_{i-\frac{1}{2},j}\,A_{i-\frac{1}{2},j}\,u_{i-\frac{1}{2},j}\\ & - \delta t\,\rho^{c}_{i,j+\frac{1}{2}}\,B_{i,j+\frac{1}{2}}\,v_{i,j+\frac{1}{2}} + \delta t\,\rho^{c}_{i,j-\frac{1}{2}}\,B_{i,j-\frac{1}{2}}\,v_{i,j-\frac{1}{2}}\\ & - \gamma_{i+\frac{1}{2},j}(\rho^{c}_{i+1,j} - \rho^{c}_{i,j}) - \gamma_{i-\frac{1}{2},j}(\rho^{c}_{i,j} - \rho^{c}_{i-1,j})\\ & - \gamma_{i,j+\frac{1}{2}}(\rho^{c}_{i,j+1} - \rho^{c}_{i,j}) - \gamma_{i,j-\frac{1}{2}}(\rho^{c}_{i,j} - \rho^{c}_{i,j-1}),\end{aligned} \tag{2}$$

which is a general five-point conservative finite-difference operator acting on the ρ°. Equation (2) is the two-dimensional extension of the one-dimensional three-point formula found in [42]. The δt represents the timestep, while the coefficients represented by the symbol "γ" are for increased phase-error control, and will be discussed shortly. The midpoint velocities and interface areas on the cell interfaces in Equation (2) are defined by:

$$u_{i+\frac{1}{2},j} = \frac{1}{2}[u_{i,j} + u_{i+1,j}] \tag{3a}$$

$$v_{i,j+\frac{1}{2}} = \frac{1}{2}[v_{i,j} + v_{i,j+1}] \tag{3b}$$

$$A_{i+\frac{1}{2},j} = \frac{1}{2}[A_{i,j} + A_{i+1,j}] \tag{3c}$$

$$B_{i,j+\frac{1}{2}} = \frac{1}{2}[B_{i,j} + B_{i,j+1}] \tag{3d}$$

Source terms on the right-hand side in Equation (1) are then added to the ρ^{*} solution:

$$\Lambda_{i,j}\,\rho^{T}_{i,j} = \Lambda_{i,j}\,\rho^{*}_{i,j} + \frac{1}{4}\,\delta t\,C_{i,j}\,(A_{i+\frac{1}{2},j} + A_{i-\frac{1}{2},j})\,(f_{i+1,j} - f_{i-1,j})$$
$$+ \frac{1}{4}\,\delta t\,D_{i,j}\,(B_{i,j+\frac{1}{2}} + B_{i,j-\frac{1}{2}})\,(g_{i,j+1} - g_{i,j-1}) + \delta t\,\Lambda_{i,j}\,S_{i,j} \tag{4}$$

where ρ^{T} is denoted as the tranported quantity of ρ.

Next, the quantity ρ^{T} is diffused using the formula

$$\Lambda_{i,j}\,\rho^{\sim}_{i,j} = \Lambda_{i,j}\,\rho^{T}_{i,j} + \nu_{i+\frac{1}{2},j}\,\Lambda_{i+\frac{1}{2},j}\,(\rho^{\circ}_{i+1,j} - \rho^{\circ}_{i,j}) - \nu_{i-\frac{1}{2},j}\,\Lambda_{i-\frac{1}{2},j}\,(\rho^{\circ}_{i,j} - \rho^{\circ}_{i-1,j})$$
$$+ \lambda_{i,j+\frac{1}{2}}\,\Lambda_{i,j+\frac{1}{2}}\,(\rho^{\circ}_{i,j+1} - \rho^{\circ}_{i,j}) - \lambda_{i,j-\frac{1}{2}}\,\Lambda_{i,j-\frac{1}{2}}\,(\rho^{\circ}_{i,j} - \rho^{\circ}_{i,j-1}) \tag{5}$$

where the diffusion coefficients denoted by the symbols "λ" and "ν" are defined later. The interface volumes appearing in Equation 5 are as:

$$\Lambda_{i+\frac{1}{2},j} = \frac{1}{2}[\Lambda_{i,j} + \Lambda_{i+1,j}] \tag{6}$$

and boundary interface values are:

$$\Lambda_{\frac{1}{2},j} = \Lambda_{1,j} \tag{7a}$$

$$\Lambda_{N+\frac{1}{2},j} = \Lambda_{N,j}, \tag{7b}$$

where N is the maximum number of grid points in the x-direction. A similar definition describes the boundary interface values in the y-direction. The quantity $\rho^{\sim}_{i,j}$ has now been both transported and diffused.

The convective transport, sources, and diffusion have been broken into stages for two reasons. First, the vectorization process is more easily treated. Second, we compute antidiffusive fluxes using $\rho^T_{i,j}$ rather than the $\tilde{\rho}_{i,j}$. The uncorrected antidiffusive fluxes are then defined as:

$$F_{i+\frac{1}{2},j} = \mu_{i+\frac{1}{2},j}\,\Lambda_{i+\frac{1}{2},j}\,(\rho^T_{i+1,j} - \rho^T_{i,j}) \tag{8}$$

$$G_{i,j+\frac{1}{2}} = \kappa_{i,j+\frac{1}{2}}\,\Lambda_{i,j+\frac{1}{2}}\,(\rho^T_{i,j+1} - \rho^T_{i,j}) \tag{9}$$

where the antidiffusion coefficients are defined as:

$$\mu_{i+\frac{1}{2},j} = \frac{1}{6} - \frac{1}{6}\epsilon^2_{i+\frac{1}{2},j} \tag{10a}$$

$$\kappa_{i,j+\frac{1}{2}} = \frac{1}{6} - \frac{1}{6}\alpha^2_{i,j+\frac{1}{2}} \tag{10b}$$

and

$$\epsilon_{i+\frac{1}{2},j} = A_{i+\frac{1}{2},j}\,u_{i+\frac{1}{2},j}\,\frac{\delta t}{2}\,(\frac{1}{\Lambda_{i+\frac{1}{2},j}}) \tag{11a}$$

$$\alpha_{i,j+\frac{1}{2}} = B_{i,j+\frac{1}{2}}\,v_{i,j+\frac{1}{2}}\,\frac{\delta t}{2}\,(\frac{1}{\Lambda_{i,j+\frac{1}{2}}}) \tag{11b}$$

are modified local Courant numbers.

We next compute "corrected" antidiffusive fluxes $F^c_{i+\frac{1}{2},j}$ and $G^c_{i,j+\frac{1}{2}}$. This correction operates on the $F_{i+\frac{1}{2},j}$ and $G_{i,j+\frac{1}{2}}$ such that no new extrema are introduced numerically into the solution and is implemented to eliminate or reduce the numerical diffusion introduced by Equation (5). Although a number of methodologies could be conceived to mathematically perform this step, two methods are used in this study, one formulated by Boris and Book [3], and the other by Zalesak [41]. The flux-correction strategy by Boris and Book extended to two dimensions is:

$$\begin{aligned} F^c_{i+\frac{1}{2},j} = S_{i+\frac{1}{2},j}\max\{0, \min[|F_{i+\frac{1}{2},j}|,\ & S_{i+\frac{1}{2},j}\Lambda^n_{i+1,j}(\tilde{\rho}_{i+2,j} - \tilde{\rho}_{i+1,j}), \\ & S_{i+\frac{1}{2},j}\Lambda^n_{i,j}(\tilde{\rho}_{i,j} - \tilde{\rho}_{i-1,j})]\} \end{aligned} \tag{12}$$

$$\begin{aligned} G^c_{i,j+\frac{1}{2}} = S_{i,j+\frac{1}{2}}\max\{0, \min[|G_{i,j+\frac{1}{2}}|,\ & S_{i,j+\frac{1}{2}}\Lambda^n_{i,j+1}(\tilde{\rho}_{i,j+2} - \tilde{\rho}_{i,j+1}), \\ & S_{i,j+\frac{1}{2}}\Lambda^n_{i,j}(\tilde{\rho}_{i,j} - \tilde{\rho}_{i,j-1})]\} \end{aligned} \tag{13}$$

where $S_{i,j+\frac{1}{2}}$ is defined as the sign of the quantity $[\tilde{\rho}_{i+1,j} - \tilde{\rho}_{i,j}]$. It is obvious from Equations (2), (4), and (5) that the basic FCT algorithm is two-dimensional. However, the flux-correction method of Boris and Book requires that we first flux-correct in one direction using Equation (12), and then in the second direction using Equation (13). The flux-correction algorithm of Zalesak is a true two-dimensional limiter and is explained in detail in [41]. In Section 4, we provide a comparison of results obtained from these two limiters.

Once the corrected antidiffusive fluxes are obtained, the time advanced solution is given as:

$$\rho^n_{i,j} = \tilde{\rho_{i,j}} - \frac{1}{\Lambda_{i,j}}(F^c_{i+\frac{1}{2},j} - F^c_{i-\frac{1}{2},j} + G^c_{i,j+\frac{1}{2}} - G^c_{i,j-\frac{1}{2}}) \tag{14}$$

We have yet to define the coefficients for Equations (2) and (5). From [42], a reduction in error (in one dimension) occurs if:

$$\gamma + \nu - \mu = \frac{1}{2}\epsilon^2 \tag{15}$$

To reduce the relative phase errors to fourth order, the γ coefficients in Equation (2) are set equal to zero. Therefore, extending to two dimensions and using Equation (15), we have:

$$\nu_{i+\frac{1}{2},j} = \frac{1}{6} + \frac{1}{3}\epsilon^2_{i+\frac{1}{2},j} \tag{16a}$$

$$\lambda_{i,j+\frac{1}{2}} = \frac{1}{6} + \frac{1}{3}\alpha^2_{i,j+\frac{1}{2}} \tag{16b}$$

The FCT algorithms which reduce the phase errors to fourth order will henceforth be referred as "P4" algorithms.

The P4 algorithms are useful because they are simple (both mathematically and for programming) and reduce the operation count of the algorithm for efficiency. However, the P4 algorithm described above has a drawback. Near regions of large gradient, instead of smoothly increasing from some small to a large value, the P4 algorithm "terraces" its way through the gradient (See [42] and [46] for a more detailed discussion of the terracing phenomenom). This terracing problem may be greatly reduced by a further reduction in the relative phase errors to sixth order. The γ coefficients were introduced in Equation (2) for exactly this purpose. Setting,

$$\gamma_{i+\frac{1}{2},j} = \frac{1}{5} + \frac{1}{5}\epsilon^2_{i+\frac{1}{2},j} \tag{17a}$$

$$\gamma_{i,j+\frac{1}{2}} = \frac{1}{5} + \frac{1}{5}\alpha^2_{i,j+\frac{1}{2}} \tag{17b}$$

and using Equations (17) in Equation (15) implies that:

$$\nu_{i+\frac{1}{2},j} = -\frac{1}{30} + \frac{4}{30}\epsilon^2_{i+\frac{1}{2},j} \tag{18a}$$

$$\lambda_{i,j+\frac{1}{2}} = -\frac{1}{30} + \frac{4}{30}\alpha^2_{i,j+\frac{1}{2}} \tag{18b}$$

Equations (17) and (18) together reduce the phase errors in the FCT algorithm to sixth order; henceforth, algorithms which perform this reduction are termed "P6" algorithms.

Up to this point, the treatment of boundary conditions has to a large part been ignored. Because boundary conditions are so varied and problem-specific, it is difficult to provide universal guidance. The discussion of boundary conditions for the 1-D FCT algorithms (See [3] and [45]) applies directly to the two-dimensional algorithms. The listing in Appendix C includes subroutine GBCOND, a subroutine which calculates symmetric, periodic, or anti-symmetric boundary conditions. The FCT driver listed in Appendix C should also provide insight towards the proper treatment of boundary conditions.

3 Two-Dimensional FCT — Code Structure

The two different flux-limiting strategies discussed in the previous section have led to two FCT solvers, both of which are listed in the Appendices. The 2-D FCT routine which uses the 1-D Boris/Book limiter in an alternating direction fashion is called QWIK2D and is listed in Appendix A. The solver listed in Appendix B uses the Zalesak limiter and is called FCT2D.

The attractive characteristics of the one-dimensional FCT solver ETBFCT (discussed in [3] and [45]) were, for the most part, retained in the two-dimensional solvers. The modularization of the FCT algorithms simplifies the insertion of the solver into other software. The software modules are vectorized for optimum execution on the Cray. The present 2-D modules are designed to solve Eulerian systems only.

Both QWIK2D and FCT2D treat the equations outlined in the previous section. By comparing the listings of the two routines, it can quickly be confirmed that large blocks of software are virtually identical. In addition, both solvers make use of two auxiliary routines, VELOCE and GBCOND, listed in Appendix C. A major difference in the two routines is that QWIK2D incorporates the flux-limiter internally while FCT2D treats the Zalesak limiter as a separate entity. The Zalesak limiter is located in the routine FLIMIT, which we have adapted virtually unchanged from Reference [3].

The following is a brief description of each of the five routines:

1. QWIK2D(RHOO,RHON,SOURCE,LXBC,RXBC,LYBC,RYBC) - takes the solution of the previous timestep, RHOO, and calculates a new solution RHON using the theory outlined in the previous section. The boundary conditions on the periphery of the domain are specified by using the four vectors LXBC, RXBC, LYBC, and RYBC. SOURCE is an array which contains possible source terms of the problem as defined by Equation (1).

2. VELOCE(U,V,DT,ISWIT) - calculates velocity-dependent terms $\{\epsilon_{i+\frac{1}{2},j}, \alpha_{i,j+\frac{1}{2}}, \mu_{i+\frac{1}{2},j}, \kappa_{i,j+\frac{1}{2}}, \nu_{i+\frac{1}{2},j}, \lambda_{i,j+\frac{1}{2}}\}$. The time increment is DT. If ISWIT is zero, then the coefficients for fourth-order accurate phase error are calculated; if ISWIT is one, then coefficients for the sixth-order accurate phase error are

computed.

3. FCT2D(RHOO,RHON,SOURCE,LXBC,RXBC,LYBC,RYBC) - uses the same calling argument list as QWIK2D (see above).

4. FLIMIT(FLX,FLY,FTD,LXBC,RXBC,LYBC,RYBC,TX,TY,FAA) - performs the flux-correction strategy as outlined by Zalesak [41]. The arrays FLX and FLY contain the uncorrected antidiffusive fluxes $F_{i+\frac{1}{2},j}$ and $G_{i,j+\frac{1}{2}}$, respectively, as defined by Equations (8) and (9). FTD contains the array of the transported and diffused solution as defined by Equation (5). The four boundary vectors are again LXBC, RXBC, LYBC, RYBC. The arrays TX and TY are the corrected fluxes $F^C_{i+\frac{1}{2},j}$ and $G^C_{i,j+\frac{1}{2}}$ which are used in Equation (14). The array FAA contains the solution from the previous timestep which is used only if the FOLD array is TRUE. In the Common Block LIMIT, the logical variables PRLIM, JPRLIM, and FOLD must be specified in the driver to be either TRUE or FALSE. See the next section for a discussion on the importance for assigning proper values to these variables.

5. GBCOND(F,LXBC,RXBC,LYBC,RYBC) - is a subroutine for setting boundary conditions for the variable "F". The vectors LXBC, RXBC, LYBC, and RYBC are used to assign certain boundary conditions on the four sides of a computational domain. It is emphasized that the computational grid does not coincide with the boundary of the physical domain (See Figure 1). In fact, the computational boundary points lie one-half of a grid spacing away from the physical boundary. The reader is referred to Appendix A for more details.

4 The Two-Dimensional Riemann Problem

In this section, we describe the two-dimensional Riemann problem and the construction of exact solutions for four different initial conditions as presented by Wagner [44]. The two-dimensional Riemann problem was chosen to demonstrate the FCT algorithm because: (1) it is a *nonlinear* convective problem; (2) it contains features that introduce numerical difficulties such as shocks and regions of large gradient; and (3) a family of exact analytical solutions exist for this problem so that quantitative accuracy comparisons can be made. In this study, we include results showing the effects of grid resolution, the two flux-limiters, and the differences between P4 and P6 algorithms.

The Riemann problem in two space cartesian coordinates is:

$$\frac{\partial}{\partial t}u(t,x,y) + \frac{\partial}{\partial x}f[u(t,x,y)] + \frac{\partial}{\partial y}g[u(t,x,y)] = 0 \qquad (19)$$

where f and g are real nonlinear functions satisfying $f'' \geq 0$ and $g'' \geq 0$ for all real u. Here, $f' = \mathrm{d}f/\mathrm{d}u$ and $g' = \mathrm{d}g/\mathrm{d}u$.

The initial conditions chosen by Wagner are:

$$u(0, x, y) = \begin{cases} u_1 & \text{for } x \geq 0, \quad y \geq 0, \\ u_2 & \text{for } x \leq 0, \quad y \geq 0, \\ u_3 & \text{for } x \leq 0, \quad y \leq 0, \\ u_4 & \text{for } x \geq 0, \quad y \leq 0. \end{cases} \tag{20}$$

Thus, the infinite domain is divided into quadrants where discontinuities may exist as shown in Figure 2.

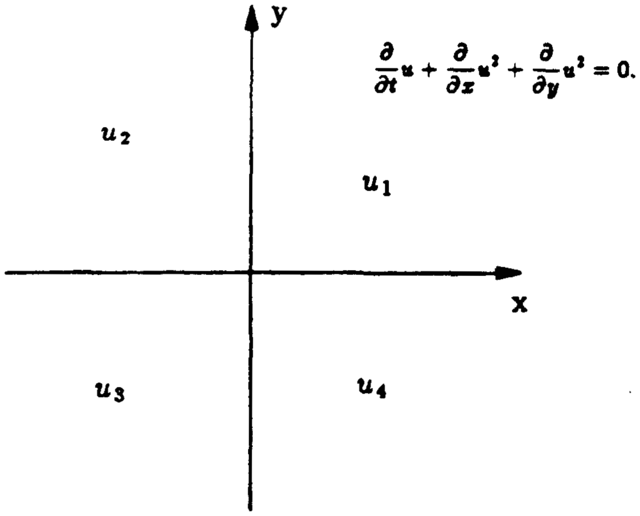
y
$\frac{\partial}{\partial t}u + \frac{\partial}{\partial x}u^2 + \frac{\partial}{\partial y}u^2 = 0.$
u_2
u_1
x
u_3
u_4

Figure 2. The two-dimensional Riemann problem with initial conditions as chosen by Wagner and $f(u) = g(u) = u^2$.

Wagner [44] shows that when $f(u) = g(u)$, the solutions are unique and obey the entropy condition (See [44] for further discussion of the entropy condition). We have chosen the simple nonlinear functions

$$f(u) = g(u) = u^2 \tag{21}$$

to satisfy this condition so that Equation (1) becomes

$$\frac{\partial}{\partial t}u + \frac{\partial}{\partial x}u^2 + \frac{\partial}{\partial y}u^2 = 0. \tag{22}$$

The form of the solution to Equation (22) with initial condition Equation (20) varies with the values of the constants u_1, u_2, u_3, u_4 relative to each other. There are a total of twenty-four cases to be considered, but reflections, inversions, and reflected inversions reduce the number to eight unique cases. Four of these eight cases were chosen to illustrate the accuracy and stability of the two-dimensional FCT algorithms.

Wagner's solution extends across an infinite two-dimensional domain. Since the computational domain must remain finite, comparison to Wagner's exact solution becomes invalid as the numerical calculation departs from the initial state near the boundaries. As such, computations were terminated before important structures such as shocks and rarefactions left the finite computational domain.

Case 1. A Simple Shock and Rarefaction

A simple shock and rarefaction results from setting initial conditions to $u_1 = u_2 = u_3 = w$ and $u_4 = v$, with $v > w$. Figure 3 depicts these initial conditions and a sketch showing the anticipated location of the shock and the rarefaction wave. For the initial conditions specified by Equation (20), rarefaction contours will *always* be parallel to one of the axes, as this case and the remaining cases will demonstrate.

In Figure 3, a smooth curve of discontinuity Γ (shock) is shown that separates the rarefaction region from regions of constant value. This curve is described by the equations

$$x_s = f'(s)t = 2st \qquad \text{and} \tag{23a}$$

$$y_s = \gamma(s)t, \tag{23b}$$

parametrized by s, t for $w \leq s \leq v$, where t is the time coordinate. (Do not confuse this $\gamma(s)$ with the coefficient γ used to denote a phase error control variable in the previous section.)

Along the rarefaction contours, shown in Figure 3, the value of s is a constant. Once the x and y locations of a point on the shock surface are known, the solution is constant above the shock surface (positive y-direction) and the solution along the rarefaction in the negative y-direction retain the value of s consistent with the shock surface. It remains to define the function $\gamma(s)$. Using the Rankine-Hugoniot condition and the proper initial condition, $\gamma(s)$ is given by the generalized integral:

$$\gamma(s) = \frac{g(s) - g(w)}{s - w} - \int_v^s exp\left(\int_s^r \frac{f''(z)(w-z)}{f(w) - f(z) - f'(z)(w-z)}dz\right)\frac{g(w) - g(r) - g'(r)(w-r)}{(w-r)^2}dr \tag{24}$$

which for our particular choice of functions of f and g (Equation (21)) reduces to

$$\gamma(s) = 2s + \frac{(w - s)^2}{(w - v)}. \tag{25}$$

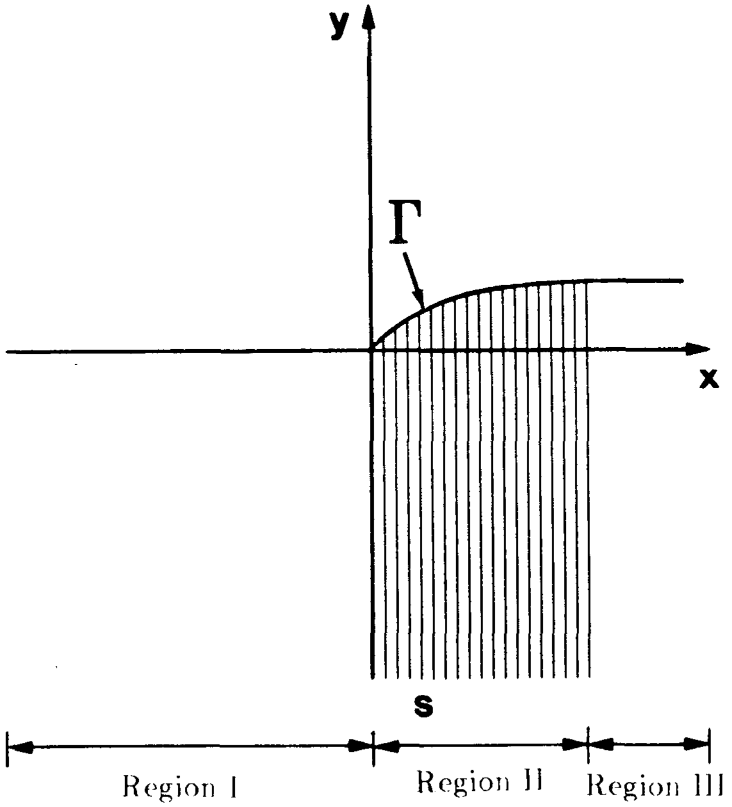

Figure 3. Case of a simple shock and rarefaction. The Γ denotes the shock surface. Vertical lines of constant value, s, denote the rarefaction region. When computing actual values it is convenient to break the domain into three parts in the x-direction, as shown.

On the computational grid, the exact solution is determined in a single sweep in the x-direction. Given the time and x location, s is determined from Equation (23a). If $s \leq w$, the solution must be in Region I in Figure 3. If $s \geq v$, the solution is in Region III. If s is between w and v, the solution falls in Region II and Equation (25) is used to determine $\gamma(s)$ while Equation (23b) is used to determine y_s. With a determination of x_s, y_s and s, all solution values are known for a given x. This procedure is continued across in the x-direction until the entire computational domain is covered. Finally, when $s = v$, the end of the curve Γ is known and the y location of the shock remains fixed for all remaining x. The particular choice of initial conditions chosen for this case is: $u_1 = u_2 = u_3 = 0$ and $u_4 = 5$.

Case 2. Four jump shocks

The case where $u_1 < u_2 < u_4 < u_3$ for the initial conditions specified by Equation (20) is shown in Figure 4. Figure 4 also shows the behavior of the field as predicted by [44]. The solution is a series of jump shocks in which rarefaction zones are absent.

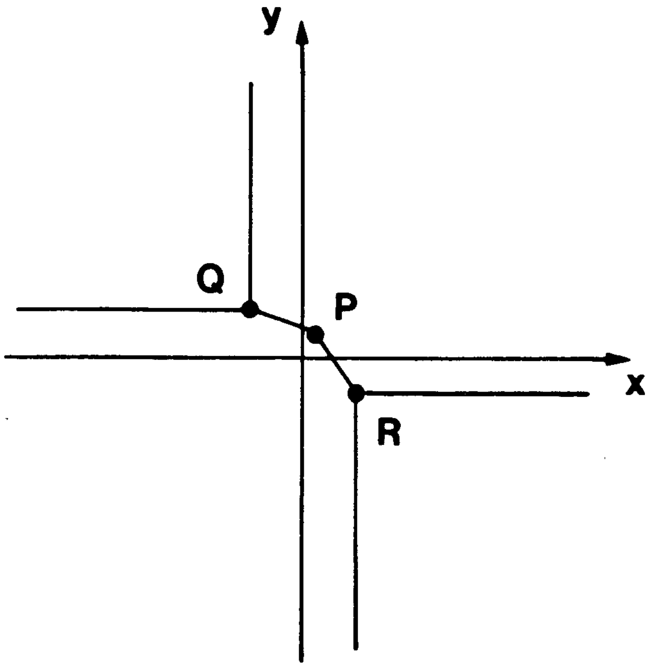

Figure 4. A case consisting of a series of shocks with no rarefactions. When the points Q, P, and R are found, the solution is complete.

To construct the exact solution, we use a theorem by Guckenheimer [47] which states:

THEOREM . *A shock surface between two constant states a,b, in a solution to the Riemann problem lies in a plane which passes through the line*

$$x = \frac{f(a) - f(b)}{a - b}t \quad y = \frac{g(a) - g(b)}{a - b}t \tag{26}$$

Using Equation (21) in Equation (26), the relation of interest is simply

$$x = (a + b)t, \qquad y = (a + b)t \tag{27}$$

Thus, in Figure 4, the point P is located at

$$x = (u_1 + u_3)t, \qquad y = (u_1 - u_3)t. \tag{28}$$

Points Q and R may also be deduced from this theorem. They are

Point Q $[(u_1 + u_2)t, (u_2 + u_3)t]$
Point R $[(u_3 + u_4)t, (u_1 + u_4)t]$

It is easy to see that, once Points Q, P, and R are known, all of the necessary regions have been defined. The specific initial conditions chosen for our study are, $u_1 = 0.5$, $u_2 = 1.0$, $u_3 = 3.0$, $u_4 = 2.0$.

Case 3. Triple Shock with Shock Stem

When the initial conditions are set such that $u_1 < u_2 < u_3 < u_4$ the exact solution is physically analogous to a triple shock with a shock stem. Figure 5 shows the location of the "triple shock" at Point P with the shock stem running from Point Q to Point P. We note from Figure 5, that once Points Q, P, and R are located, all of the regions may be mapped. Using Equations (23),(24), and (26), the points are given as

Point Q $[(u_1 - u_2)t, (u_2 + u_3)t]$

Point P $[(u_1 - u_3)t, (u_1 + u_3)t]$

Point R $\left[2u_3t, \left(2u_3 + \frac{(u_1 - u_3)^2}{(u_1 - u_4)}t\right)\right]$

The shock curve Γ is created in the same manner as in Case 1. The specific initial condition for this case, $u_1 = -2.0$, $u_2 = -1.0$, $u_3 = -0.5$, $u_4 = 2.0$, is shown in Figure 5. The negative initial conditions for this example point out that, although physical analogy is convenient in interpreting the results, it can be carried too far. The system solved is strictly mathematical.

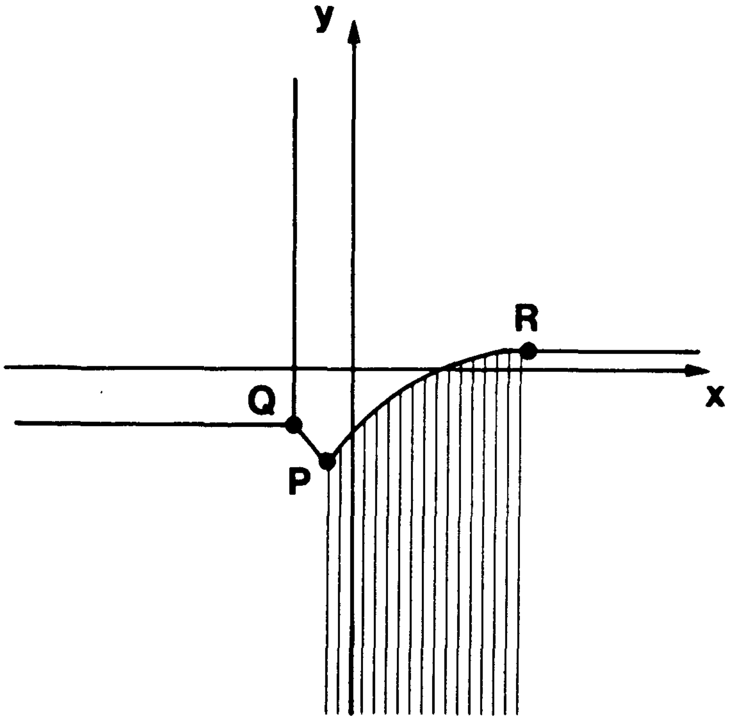

Figure 5. The case resulting in a triple shock with a shock stem (from point Q to P) and a single rarefaction region. Again, vertical lines denote the rarefaction zone where the value of the dependent variable is invariant in the y direction. The points Q, P, and R must be found to obtain a solution.

Case 4. Cusp Shock

When the initial conditions of Equation (20) are set such that $u_2 \leq u_4 \leq u_3 \leq u_1$, the solution produces a set cusp shock. Figure 6 depicts this situation and pertinent points which must be found. In our study we chose $u_1 = u_3 \geq u_2 = u_4$ such that Point Q and Point T coincide. Thus, Point Q determines where the two rarefactions meet and, in our case, where the shocks begin. Point Q is located at $(2u_1t, 2u_1t)$. Points R and S are locations where shocks end and rarefactions begin. The construction of this case can be simplified by noting that the solution is symmetric about line $\mathbf{L_1}$ (see Figure 6). Therefore, we need only locate Point Q and construct only one of the rarefactions according to the procedure outlined for the first case. Then, the region on the other side of line $\mathbf{L_1}$ is constructed by reflecting the solution about the line $x = y$. Our specific initial condition for this case was $u_1 = u_3 = 1.0$ and $u_2 = u_4 = -2.0$.

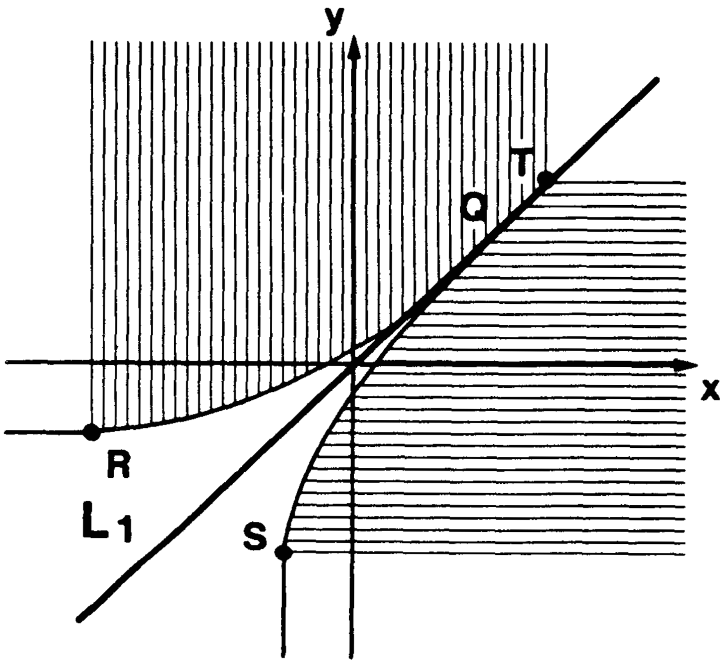

Figure 6. The case resulting in a cusp shock with two rarefaction zones. The solution exhibits symmetry, so only half the solution must be calculated. For the initial condition of interest, points Q and T coincide. Points S and T are the two points which must be found.

Results for the 2-D Riemann Problem

In each of the four cases described, the effects of grid resolution, type of flux-limiter, and the order accuracy of the phase error of the solver were investigated. A total of eight runs were required for each example to perform such an investigation. Table 1 summarizes the required schedule for each example. In Appendix C a listing of the driver which sets up the initial and boundary conditions, shows proper use of the CFL condition, and illustrates the calls to the FCT subroutines is provided. An auxiliary routine that computes the exact solution, error measures, and then prints outs the results has been omitted.

Table 1. The eight cases run for each example problem to compare the effects of flux-limiter, phase accuracy and grid resolution.

Algorithm/Limiter	Phase Accuracy	Grid Resolution
QWIK2D/Boris & Book	Fourth Order	100 × 100
QWIK2D/Boris & Book	Sixth Order	100 × 100
QWIK2D/Boris & Book	Fourth Order	50 × 50
QWIK2D/Boris & Book	Sixth Order	50 × 50
FCT2D/Zalesak	Fourth Order	100 × 100
FCT2D/Zalesak	Sixth Order	100 × 100
FCT2D/Zalesak	Fourth Order	50 × 50
FCT2D/Zalesak	Sixth Order	50 × 50

To evaluate and compare the accuracies of each of the runs, the following error measures are introduced:

$$L_1 = \frac{1}{N} \sum_i \sum_j |u - u_{i,j}| \tag{29}$$

$$L_2 = \frac{1}{N} \left[\sum_i \sum_j (u - u_{i,j})^2 \right]^{1/2} \tag{30}$$

$$L_\infty = \max |u - u_{i,j}| \tag{31}$$

where N is the number of grid points (either 100 × 100 or 50 × 50), u is the analytic solution at $\{i, j\}$ and $u_{i,j}$ is the calculated solution. Since all dependent and independent variables for this problem are nondimensional, the domain goes from 0 to 1 in both dimensions.

The values of L_1, L_2, and L_∞ versus time for each of the eight cases of our four example problems have been included in Appendix D. The L_1 values have been multiplied by a factor of 100; the L_2 values have been multiplied by 1000. Therefore, it may be noted immediately that the FCT solvers are highly accurate in that the L_1-error usually falls between 0.003 and 0.042 and the L_2-error falls between 0.0003 and 0.0043. Since the Courant number for all of the cases was held constant at 0.2, the 50 × 50 gridpoint cases required only half the number of timesteps as the 100 × 100 gridpoint cases. Thus, although the number of timesteps for the two grid systems is different, the absolute time at any point along the abscissa for a particular case is approximately the same. Because of the different initial conditions, times from each example problem cannot be compared. The numerical values of time were deemed irrelevant.

A common characteristic for all of the L-error measures in all cases is the oscillatory behavior. These oscillations are explained by noting the difference in "wavelength" of the 100 × 100 calculations and the 50 × 50 calculations. The observed wavelength for the latter is twice that of the former. When a shock for the exact solution synchronizes spatially with the finite-difference grid, the L-error measures are at their minimum values. When the shock is directly between two finite-difference points, the errors are at a maximum. Thus, as the shock travels across the grid system, minima and maxima are continually observed with continuous values connecting the extrema. Since the distance between finite-difference points on the 50 × 50 grid is twice that of the 100 × 100 grid (since the Courant number remains the same), the discrepancy in wavelength is explained.

Figure 7 compares the error measures obtained on a 100 × 100 grid and on a 50 × 50 grid for the simple shock with rarefaction example. (Note the scale difference on the ordinate between the two figures.) The QWIK2D software utilizing the Boris/Book flux-limiter and the fourth-order accurate phases were used. Not surprisingly, the higher resolution grid provides more accuracy. All three of the error measures for the 100 × 100 grid are about a factor of two lower than those for the 50 × 50 grid. The effect of grid resolution on accuracy is well-known [48]. A smaller Δx implies a small error, and our results confirm this fact. However, the fact that grid refinement by a factor of two only produced a factor of two increase in accuracy is significant. Most of the error is occurring near shocks where the FCT algorithm is only first order accurate. This study was conducted only to determine the results of varying grid size; the execution times were not weighted in these results to determine an "accuracy/efficiency" factor.

A prerequisite to understanding the discussion comparing the 1-D Boris/Book to the Zalesak limiter requires a more detailed explanation of the manner in which Zalesak's limiter is used. Zalesak's limiter has three user-specified options that must be set by the FCT solver prior to the call to the flux-limiter routine FLIMIT. These option variables are Fortran logical variables and are called PRLIM, JPRLIM, and FOLD. Activating the PRLIM parameter (*i.e.* setting it equal to TRUE) zeroes the flux under certain specific circumstances, causing the limiter to be more diffusive. The second logical parameter, JPRLIM, "pre-limits" the fluxes using the Boris/Book limiter before passing on to the multidimensional limiter of Zalesak. (PRLIM is ignored when JPRLIM is set equal to TRUE.) The third parameter, FOLD, allows the limiter to look back to the solution from the previous timestep to find upper and lower bounds on the new solution. Since PRLIM is ignored when JPRLIM is TRUE, it can be shown that there are only six unique cases of values for these three parameters. These six cases have been run for the simple rarefaction and shock problem and the cusp shock problem using fourth order accurate phases on the 50 by 50 grid system. The six cases

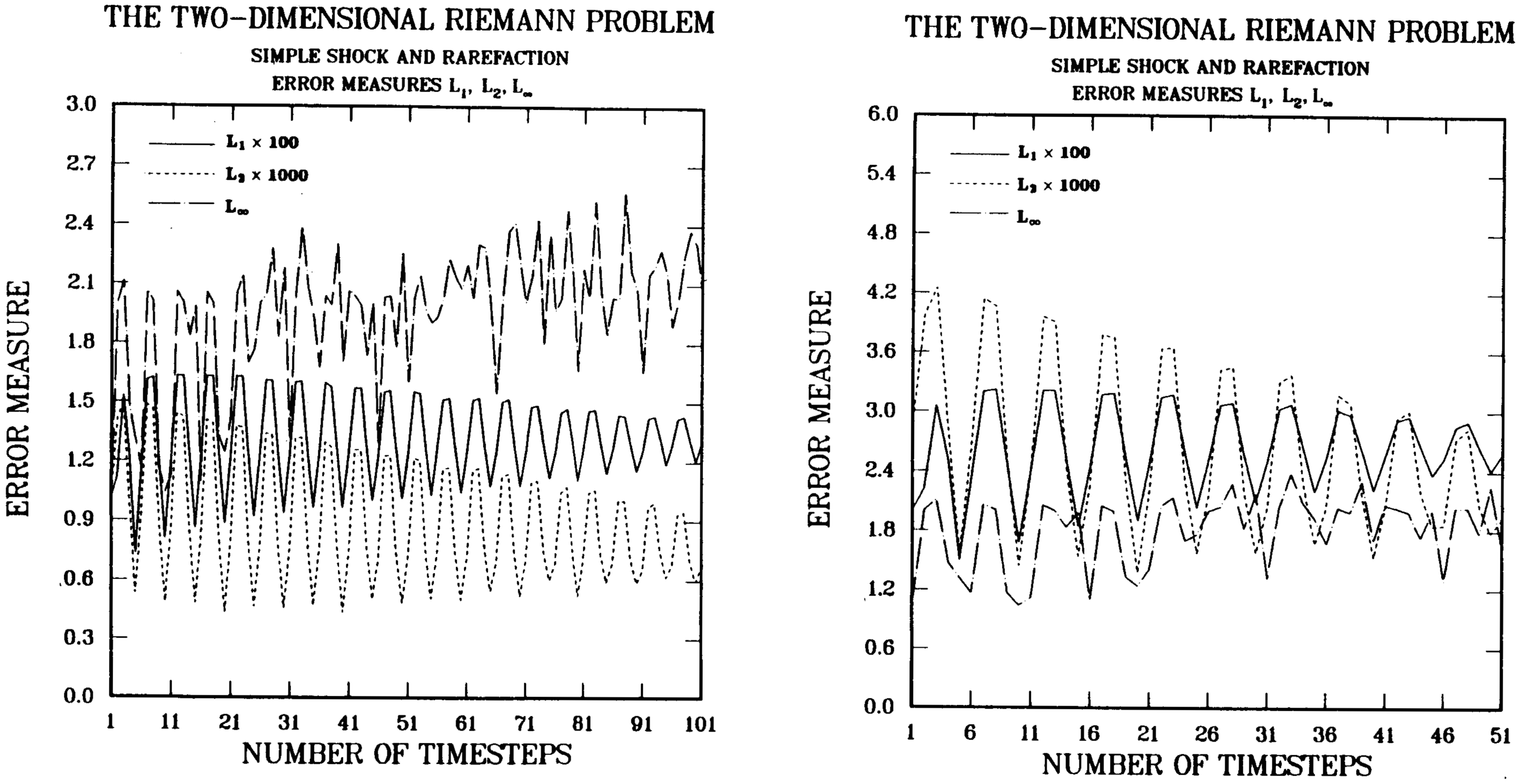

Figure 7. The effect of grid resolution. Fig. 7a results obtained on a 100 × 100 grid; Fig. 7b results on a 50 × 50 grid. Note the difference of scale on the ordinate. The factor of two difference in accuracy indicates that most of the error accrues near shocks where the algorithm is only first-order accurate.

for the cusp shock example are shown in Appendix E. In studying these figures, we observe that the first two cases, Figures E.1 and E.2, are much more accurate than Figures E.3 through E.6 (In fact, in E.3 and E.4 the errors are so large as to go off the scale of the graph. The important observation to note, however, is that these results are highly inaccurate). In the first two cases JPRLIM was TRUE while in the latter four cases it was set to FALSE. Obviously, "pre-limiting" the fluxes using the Boris/Book limiter provides the most accurate results. Finally, in comparing E.1 to E.2, the FOLD parameter has little effect on the results, but setting it FALSE does result in a slightly smaller L_∞. Thus, the combination which, for the 2-D Riemann problem, appears to offer the most accuracy using the FCT2D algorithm which utilizes FLIMIT is: $\{$ PRLIM, JPRLIM, FOLD $\} \Longrightarrow \{$ TRUE, TRUE, FALSE$\}$. This combination specifies pre-limiting the fluxes using the Boris/Book limiter and *not* looking back at the solution on the previous timestep. The above specification of variables was applied throughout the study. Although this discussion has centered around the results obtained for the cusp shock example problem, we have also performed this study using the simple shock and rarefaction problem. The same results were obtained.

Since our study using FCT2D utilized the pre-limiter of Boris and Book on Zalesak's multidimensional limiter, it is natural to ask what difference there is between QWIK2D which *only* uses the Boris/Book limiter, and FCT2D with the JPRLIM parameter turned on. This difference can be resolved by comparing any QWIK2D result in Appendix D to any FCT2D result which shares the same order of accuracy of algorithm, and grid resolution, *e.g.* [D.1] and [D.5]. In the figures, a difference is not discernable; numerically, the two agreed well within 0.1% of one another. However, QWIK2D which uses the Boris/Book limiter is six times faster on a Cray 1S than the FCT2D/FLIMIT software which uses the Zalesak limiter. The same result is observed for several problems other than the 2-D Riemann problem. Clearly, the QWIK2D algorithm, since it is comparable in accuracy and much greater in efficiency, is the method of choice.

We now focus on comparing the results obtained from the P4 algorithm to those of the P6 algorithm. The "plateau", or "terracing", effect of the P4 schemes can be clearly seen in Figure 8a. The additional phase accuracy of the P6 algorithm does to a certain extent cure the stair-stepping problem, as shown in Figure 8b. However, the additional reduction of dispersion error does not reduce the error measures. Figures D1 and D2 compare the P4 to the P6 results for the simple shock and rarefaction example on the 100 by 100 grid using QWIK2D. The L_1 measures are higher for the P6 case than for the P4 case. Even the L_2 measures are marginally larger in the P6 case than for P4, although, on the average, the P6 case does have a slightly smaller L_∞. For the four jump shock example (See Figures D9 and D10) the L_1 errors are slightly larger, the L_2 errors are comparable, and the L_∞ errors are slightly smaller for the P6 case compared

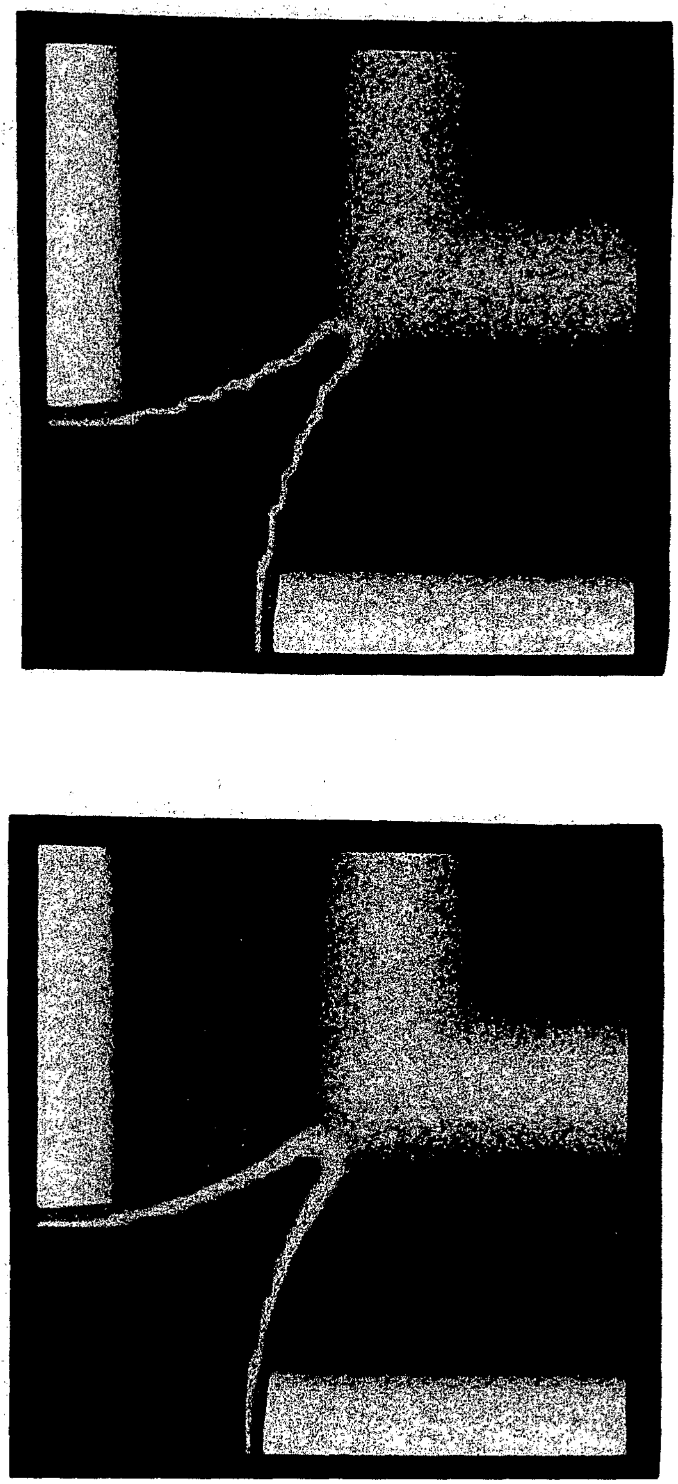

Figure 8. Results for the cusp shock problem demonstrating the differences in terracing between the P4 scheme (top figure) and the P6 scheme (bottom figure). Note the difference in the width of the yellow band around the red cusp region. Difference in the red color is due only to the photographic reproduction process.

to the P4 results. For the triple shock example (Figures D17 and D18) all of the error measures were smaller for the P4 results compared to the error measures of the P6 results. Finally, Figures D25 and D26 compare the error measures for the cusp shock example. As in the first two examples, the P4 L_1 error measure is smaller than its P6 counterpart, the L_2 errors are comparable, and the P6 L_∞ error is noticeably smaller than those for P4. Thus, the results depend on the particular initial conditions chosen as applied to the 2-D Riemann problem.

The P6 algorithm eliminates the stair-stepping effect by introducing diffusion in regions of large gradient. This additional diffusion explains the slightly greater accuracy of the P4 algorithm, especially for the L_1 and L_2 errors which are taken over the entire region.

5 A Numerical Study of Hydrodynamic Instability

In this section, we use the two-dimensional FCT algorithm to numerically compute a class of hydrodynamic instabilities, demonstrating that the gain in numerical stability does not preclude the description of a flow field that is known to be physically unstable. Specifically, we examine three test problems to illustrate Kelvin-Helmholtz and Rayleigh-Taylor instabilities in a single component, compressible ideal gas. These classical unstable flows have been theoretically and experimentally well-studied [49], and thus, we shall not present a detailed discussion of instability theory. Furthermore, the FCT method has been previously applied to this class of flows by Boris and his coworkers at the Naval Research Laboratory, and it is this work which we follow in assessing the qualitative agreement of the numerical computations.

The principles of conservation of mass, momentum and energy for a frictionless, nonconducting compressible gas are represented by the Euler equations. A precise mathematical statement of these laws is expressed in the following set of nonlinear partial differential equations:

Conservation of Mass

$$\frac{\partial}{\partial t}\rho + \frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho v) = 0 \tag{32}$$

Conservation of Momentum

$$\frac{\partial}{\partial t}(\rho u) + \frac{\partial}{\partial x}(P + \rho u^2) + \frac{\partial}{\partial y}(\rho u v) = b_x \tag{33a}$$

$$\frac{\partial}{\partial t}(\rho v) + \frac{\partial}{\partial x}(\rho v u) + \frac{\partial}{\partial y}(P + \rho v^2) = b_y \tag{33b}$$

Conservation of Energy

$$\frac{\partial}{\partial t}(\rho(e + \frac{u^2}{2} + \frac{v^2}{2})) + \frac{\partial}{\partial x}(\rho u(\epsilon + \frac{u^2}{2} + \frac{v^2}{2} + \frac{P}{\rho}))+$$
$$\frac{\partial}{\partial y}(\rho v(e + \frac{u^2}{2} + \frac{v^2}{2} + \frac{P}{\rho})) = (b_x u + b_y v) \tag{34}$$

where the two-dimensional velocity field (u, v) are, respectively, the components in the x and y directions, ρ is the gas density, P is the gas pressure, e is the specific internal energy, and b_x, b_y are body force components.

For the sake of simplicity, the computations are restricted to a thermally perfect ideal gas and introduce a nondimensionalization based on a reference undisturbed sound speed and an arbitrary length scale which is appropriately chosen to resolve the fluid motions. With this choice, the conservation equations are retained in their identical form and the ideal gas law is given by:

$$P = (\gamma - 1)\rho e \tag{35}$$

where γ is the constant adiabatic index.

A discussion of the problem-specific initial and boundary conditions are deferred to those sections that detail the example calculations. In each of the following computations, a fixed mesh of 50 by 50 uniformly spaced cells is used and an appropriate time step is chosen based on the CFL condition:

$$\Delta t \leq \beta \ \max\{\frac{(|u| + c)^2}{\delta x^2} + \frac{(|v| + c)^2}{\delta y^2}\}^{\frac{1}{2}}$$

where c is the local sound speed and β is a constant whose value must be taken less than 0.5.

1. Kelvin-Helmholtz Instability

It has long been known that a shear layer is unstable and when perturbed produces a Kelvin-Helmholtz instability which occurs even in the absence of viscosity. To demonstrate a computation of this unstable flow, we consider a metastable flow initial condition defined by $u(y)$, $v = 0$. In particular, a velocity profile is chosen with a stagnation plane along the centerline of the computational domain, and a free shear layer is specified as:

$$u(y) = U \tanh(Ay)$$

where y is measured from the stagnation plane, U is a characteristic outer flow velocity and A is related to the inverse of the shear layer thickness. In these calculations, nondimensional U and A are given the values 0.1 and 0.56, respectively.

This field is then perturbed by superimposing a disturbance along the stagnation plane. This perturbed, divergent-free velocity field, (u', v'), is taken as

$$u' = \epsilon \sin(\pi x l \lambda) \exp(-\pi y l \lambda) \cdot \operatorname{sgn}(y)$$

$$v' = \epsilon \cos(\pi x l \lambda) \exp(-\pi y l \lambda)$$

where $\operatorname{sgn}(y)$ has the value of 1 above the stagnation plane and -1 below it. With the proper choice of λ a fixed number of disturbances can be produced that grow and interact with time. At the flow entrance and exit planes, periodic boundary conditions are assigned, implying that any variable crossing these planes reappears at the opposite boundary. Zero gradients are imposed on the top and bottom boundaries for all transported variables. To graphically display the resultant flow field, massless marker particles are placed on the stagnation plane that are allowed to advect in a Lagrangian fashion according to the local flow field. (This Lagrangian marker particle scheme is given in Appendix F.)

In Figure 9, three time planes displaying the marker particle field are shown. Actual numerical computations are represented on one-half of the domain and we have placed adjacent, repetitive calculations to better display the growth and interaction of the disturbances. In these figures, the development of vortical cells which roll up the shear region is observed. Later in time, these vortices grow and interact to eventual coalescence. These calculations closely resemble Boris' prior calculations [50] and the interested reader should consult this reference for discussion of a comparison to linearized theory.

2. Rayleigh-Taylor Instability

When a fluid is accelerated in the direction of a density gradient, the fluid motion is unstable and a Rayleigh-Taylor instability will be produced. For example, when a fluid is heated from below an unstable density stratification is produced whereby a thin light layer is formed under a heavy field. The less dense layer eventually breaks into mushroom-shaped plumes or thermals [51] that are well-recognized features of the instability.

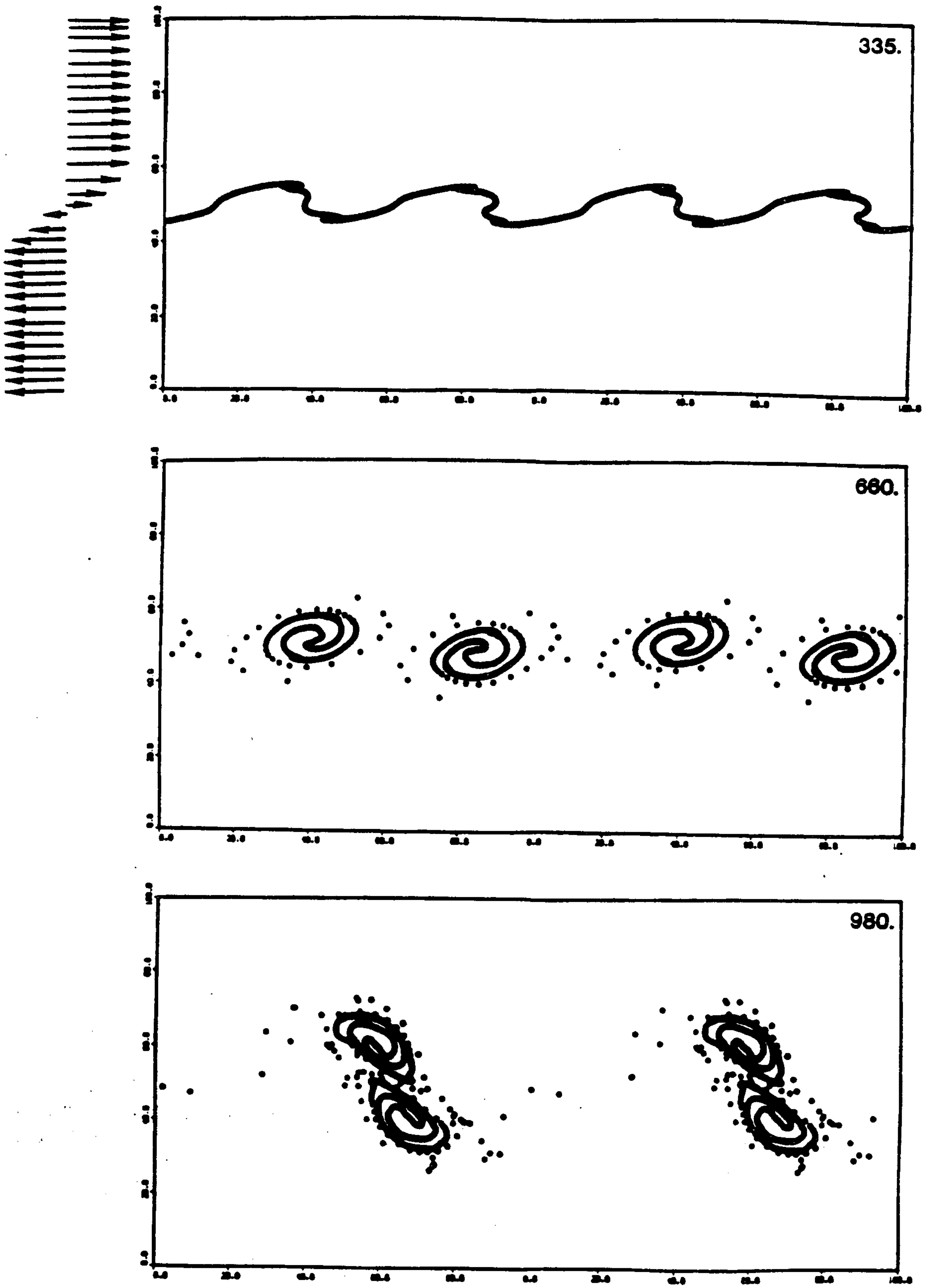

Figure 9. Three time frames showing the evolution of the Kelvin-Helmholtz instability using a marker particle field.

A similar flow field is produced when considering the acceleration of a thin dense layer [52]. We examine this case by treating the Euler equations with an intial density field given by a thin layer of dense particles held in metastable equilibrium by a hydrostatic pressure field. In this calculation, a single component body force is assigned within each cell as determined by the number of particles contained within each cell. As such, the marker particles participate in the computations by forcing the flow to be unstable. Each particle is assigned a density and moved with the frictionless compressible fluid in a Lagrangian fashion. The body force is determined by depositing the particle density at the computational cell corners according to bilinear interpolation (See Appendix F for details).

To produce the instability, the dense layer is perturbed with a prescribed periodic disturbance of the marker field. In this rectangular domain, we treat the top and bottom boundaries as impermeable walls and impose periodic conditions on the right and left boundaries. A fixed number of instability "cells" are produced by the perturbed field which are then allowed to grow and interact. In the following calculation, we use 2000 marker particles, placed initially across the domain in two cells. Each marker was assigned a density of 1.0 and a gravity constant of 0.05.

Figure 10 shows a sequence of the particle field which drive the unstable flow. The initial quiescent field was perturbed to form two disturbances. As the particles seek a new state of equilibrium, a downwardly-moving spike is first seen. These disturbances eventually open to produce mushroom-shaped caps that later become dispersed due to the spreading of the marker particles. Further in time, the markers move to the bottom of the domain, establishing a stable density stratification.

3. Shock-wave Interaction with Density/Temperature Discontinuity

In this final study, the unstable flow generated when a shock wave interacts with a density and temperature discontinuity (*i.e.* a combustion wave) is examined. Markstein [53] first studied this flow in experiments in which he passed a shock through a spherical flame surface. He found that the combustion wave quickly becomes "wrinkled", producing a mechanism to accelerate combustion. Markstein correctly recognized that the resulting unstable flow was actually a manifestation of a Rayleigh-Taylor instability.

In this study, we examine a similar flow configuration and treat the evolution of the flow field when a shock passes through a spherical "bubble" interface separating a discontinuity in density and temperature. A shock tube geometry

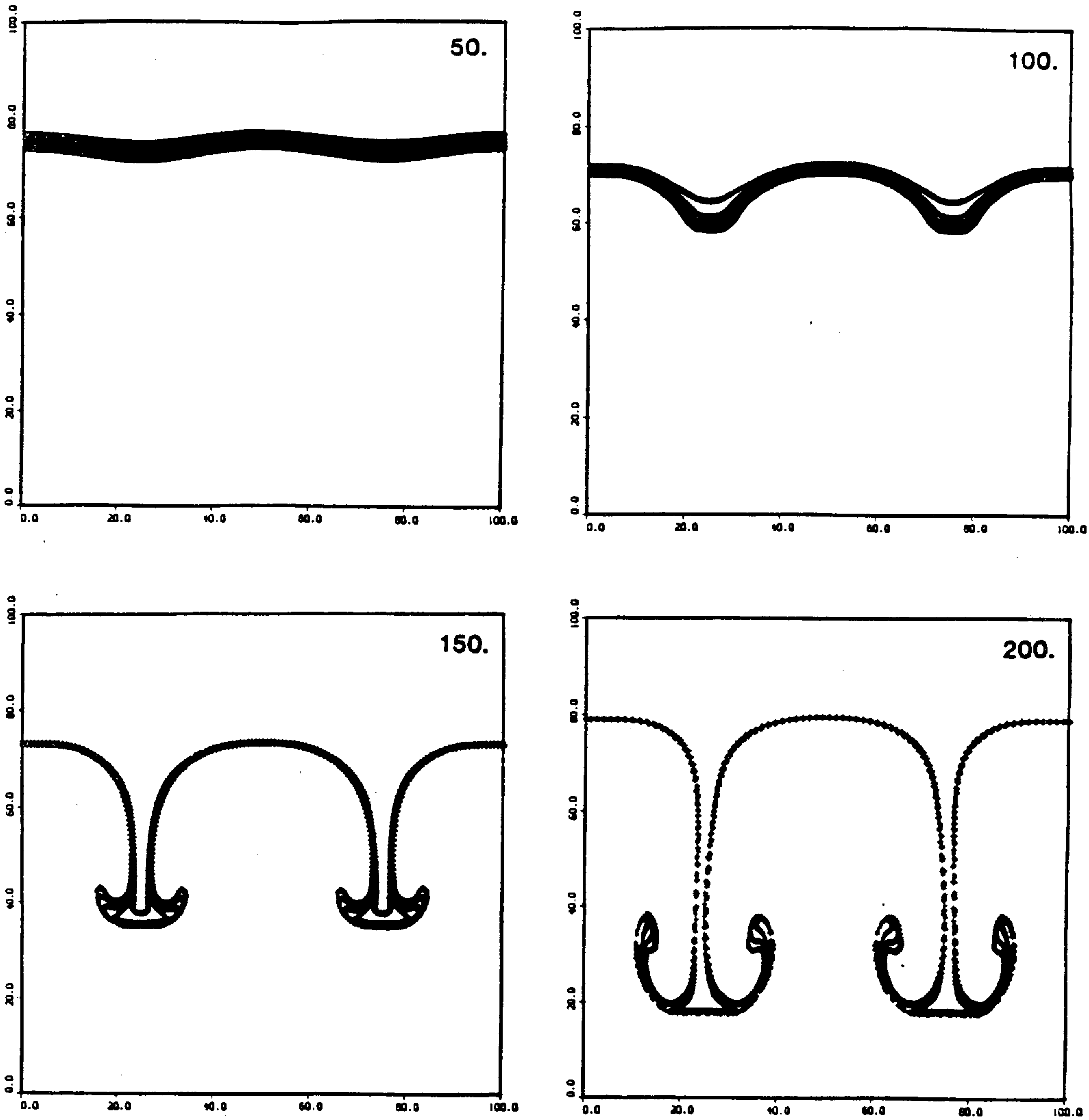

Figure 10. Four time frames showing the development of the Rayleigh-Taylor instability using 2000 marker particles.

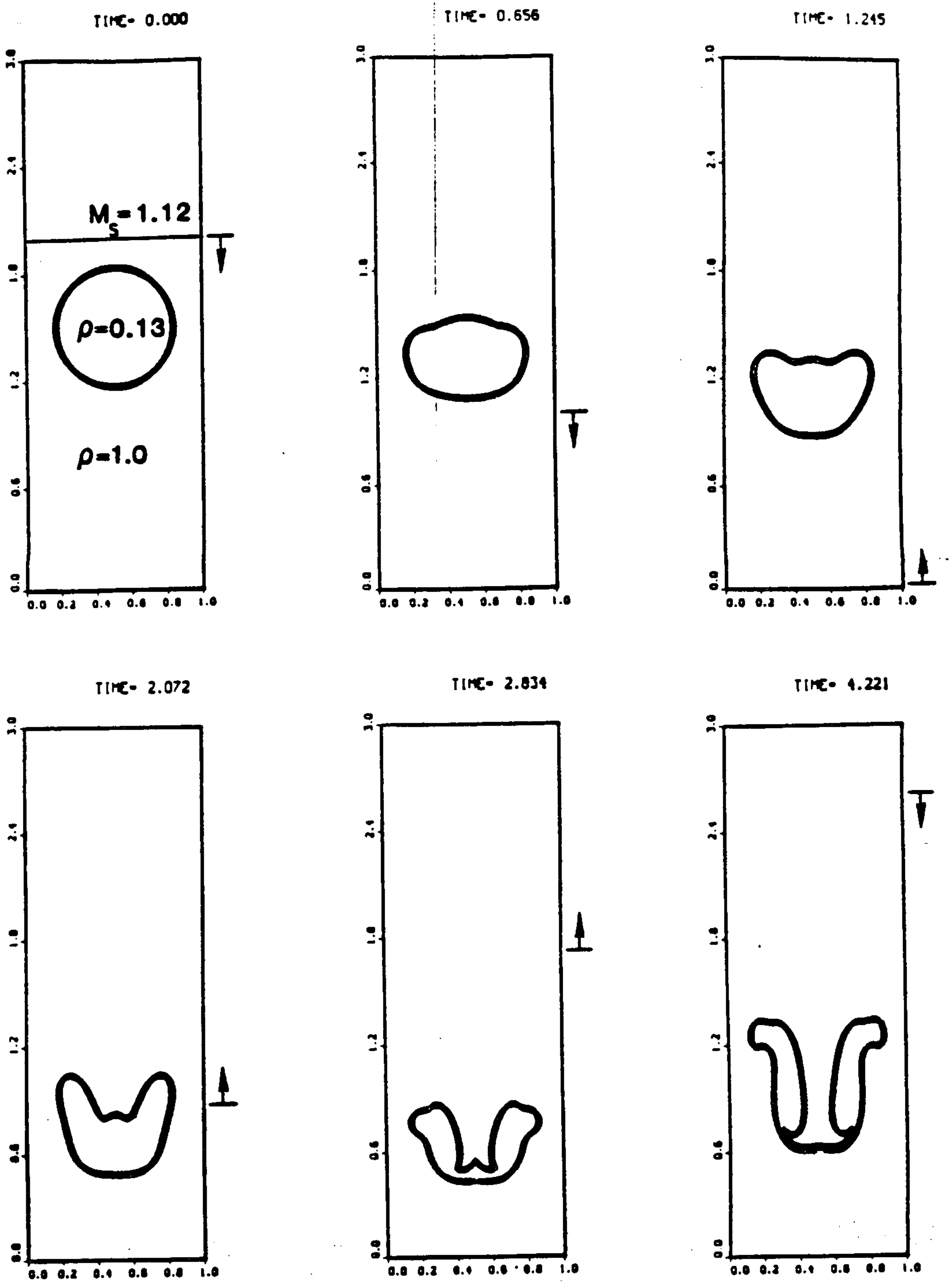

Figure 11. A shock wave interacts with a spherical discontinuity in density and temperature in a cylindrical coordinate system. The reflected shocks continue to interact with the bubble surface to form a Kelvin vortex ring.

is selected by treating reflective wall boundaries on the periphery of the computational domain. A uniform high pressure ($P_s/P_o = 1.7$) is placed on the gas driver section in the upper one-fifth of the domain. Release of this pressure produces a shock which moves at a Mach number of 1.12, which is consistent with Markstein's experiments. A sequence of shock rarefaction patterns are then seen to interact with the low-density bubble in Figure 11. The contact surface of the bubble was marked with tracer particles which follow local gas velocities.

In these numerical calculations, a domain with a 3 to 1 aspect ratio using 150 by 50 computational cells is considered. To simulate the experiment of Markstein, a cylindrical coordinate system is used. The low density, constant pressure bubble is taken to be 0.13 that of the surrounding ambient gas density (the time constant for the time variable is scaled by 0.48 *ms*). After the shock passes through the discontinuity, a center jet is seen to evolve as high density gas is collapsed into the center of the bubble. Continual interaction with reflected shocks cause the bubble surface to be stretched over the jet spike forming a Kelvin vortex ring. Clearly, as Markstein first noted, the shock interaction easily produces vorticity by the Rayleigh-Taylor and Kelvin-Helmholtz mechanisms. Similar experimental flow patterns were seen in [53] and the interested reader should refer to [53] and [54] for further details.

6 Conclusions and Summary

In this study, we present a two-dimensional numerical technique capable of addressing a large class of convectively-dominated heat transfer/fluid mechanics problems. In keeping with the current direction of research, the method is a nonlinear finite-difference technique that exhibits stability, accuracy, monotonicity and conservation even for flows containing regions with large gradients. The technique is an extension of the well-documented 1-D FCT method introduced by Boris and Book. The algorithms in the Appendices are vectorized and modularized so as to be computationally efficient, compact, and easy-to-use.

All of the aforementioned attribues of the FCT technique has been demonstrated in benchmarking calculations of a 2-D Riemann problem for four different sets of initial conditions. The time-dependent L-error measures shown in Appendix D clearly indicate the accuracy and stability of the technique. The method's efficiency has also been gauged by a call to QWIK2D for a 50 by 50 computational grid requiring only 0.00428 cpu second of execution time on the Cray 1S. In solving several of the classical instability problems, we have shown that the stability of FCT does not preclude the solution of problems known to be physically unstable. It has been shown previously [39] that FCT does not preclude new maxima and minima to be introduced *physically* into the solution. Thus, the FCT approach can truly track the physics of a problem — not an artificial numerical path that can mask physical phenomena.

The Zalesak limiter is a true two-dimensional limiter. In [41], the Zalesak limiter was shown to be more accurate than using a 1-D version of FCT with the Boris/Book limiter in an alternating direction methodology. However, we have observed that our 2-D FCT solver which uses the 1-D Boris/Book limiter to be a most accurate approach in solving the 2-D Riemann problem for several sets of initial conditions. Additionally, QWIK2D is much more efficient. These observations concur with [37]. We therefore recommend the QWIK2D software over that of the FCT2D/FLIMIT combination.

The P6 FCT algorithm, generally, is slightly less accurate than the P4 algorithm due to the additional diffusion terms. However, the sixth-order accurate phases greatly reduces the terracing effect that many researchers find objectionable. This additional diffusion may also be necessary for the solution of certain complex problems which have very large amplitude shocks and/or large gradients. Thus, we recommend the P6 algorithm as a first choice. If slightly higher accuracy is desired and the stair-stepping phenomena is not objectionable, then the P4 algorithm may be tried.

Because software development, especially with today's complex vector machines and the new parallel processors, is a time consuming art, the mathematical description of a numerical technique alone supplies an incomplete tool to the user. Boris and Book have set an open, broad-minded example for the positivity-preserving computational community by publishing a listing of their 1-D FCT technique along with detailed instruction of its use. We continue in this spirit by including listings of FCT software for two-dimensional calculations.

Acknowledgments

We gratefully acknowledge the suggestion by Alex Treadway to examine Wagner's work as a two-dimensional test problem with an exact solution to compare against the results of the FCT method. We also thank Alex and Rick Givler who took the time to review this work.

7 References

1. Von Neumann, J. and Richtmyer, R.D., "A Method for the Numerical Calculation of Hydrodynamic Shocks," *J. Appl. Physics* , **21** , p. 232, 1950.

2. Godunov, S.K., "Finite Difference Method for Numerical Computation of Discontinuous Solutions of the Equations of Fluid Dynamics," *Matematicheskii Sbornik* , **47** , p. 271, 1959.

3. Book, D.L., editor, **Finite-Difference Techniques for Vectorized Fluid Dynamics Calculations**, Springer-Verlag, New York, N.Y., 1981.

4. Roache, P.J., **Computational Fluid Dynamics**, Hermosa Publishers, Albuquerque, N.M., 1976.

5. Van Leer, B., "A Choice of Difference Schemes for Ideal Compressible Flow," Thesis, University Observatory, Leiden, Netherlands, 1970.

6. Gelinas, R.J., Doss, S.K., and Miller, K., "The Moving Finite Element Method: Applications to General Partial Differential Equations with Multiple Large Gradients," *J. Appl. Physics* , **40** , p. 202, 1981.

7. Lax, P.D., "Weak Solutions of Nonlinear Hyperbolic Equations and their Numerical Computation," *Comm. Pure Appl. Math.* , **7** , p. 159, 1954.

8. Lax, P.D. and Wendroff, B., "Systems of Conservation Laws," *Comm. Pure Appl. Math.* , **13** , p. 217, 1960.

9. Rusanov, V.V., "Non-linear Analysis of the Shock Profile in Difference Schemes," **Proceedings of the Second International Conference on Numerical Methods in Fluid Dynamics, Lecture Notes in Physics**, (M. Holt, Ed.), **8** , Springer-Verlag, New York, N.Y., 1971.

10. Courant, R., Isaacson, E., and Rees, M. "On the Solution of Nonlinear Hyperbolic Differential Equations by Finite Differences," *Comm. Pure Appl. Math.* , **5** , p. 243, 1952.

11. MacCormack, R.W., "The Effect of Viscosity in Hypervelocity Impact Cratering." AIAA Paper No. 69-354, 1969.

12. Lapidus, A., "A Detached Shock Calculation by Second-Order Finite Differences," *J. Appl. Physics* , **2** , p. 154, 1967.

13. Van Leer, B., "Towards the Ultimate Conservative Difference Scheme. I. The Quest of Monotonicity, **Lecture Notes in Physics," (H. Cabannes and R. Temam, eds.)** , **18** , Springer, Berlin, 1973.

14. Boris, J.P. and Book, D.L., "Flux-Corrected Transport. I. SHASTA, A Fluid Transport Algorithm that Works," *J. Comp. Physics* , **11** , p. 38, 1973.

15. Woodward, P. and Colella, P., "The Numerical Simulation of Two-Dimensional Fluid Flow with Strong Shocks," *J. Comp. Physics* , **54** , p. 115, 1984.

16. Harten, A., "High Resolution Schemes for Hyperbolic Conservation Laws," *J. Comp. Physics* , **49** , p. 357, 1983.

17. Van Leer, B., "Toward the Ultimate Conservative Difference Scheme. II. Monotonicity and Conservation Combined in a Second-Order Scheme," *J. Comp. Physics* , **14** , p. 361, 1974.

18. Van Leer, B., "Toward the Ultimate Conservative Difference Scheme. III. Upstream-Centered Finite-Difference Schemes for Ideal Compressible Flow," *J. Comp. Physics* , **23** , p. 263, 1977.

19. Van Leer, B., "Toward the Ultimate Conservative Difference Scheme. IV. A New Approach to Numerical Convection," *J. Comp. Physics* , **23** , p. 276, 1977.

20. Van Leer, B., "Toward the Ultimate Conservative Difference Scheme. V. A Second-Order Sequel to Godunov's Method," *J. Comp. Physics* , **32** , p. 101, 1979.

21. Roe, P.L., "The use of the Riemann Problem in Finite Difference Schemes." **Proceedings of the Seventh International Conference on Numerical Methods in Fluid Dynamics Lecture Notes in Physics**, (W.C. Reynolds and R.W. MacCormack, Eds.), **141** , Springer-Verlag, N.Y., 1971.

22. Roe, P.L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *J. Comp. Physics* , **43** , p. 357, 1981.

23. Colella, P. and Woodward, P.R., "The Piecewise Parabolic Method (PPM) for Gas-Dynamical Simulations," *J. Comp. Physics* , **54** , p. 174, 1984.

24. Chapman, M., "FRAM-Nonlinear Damping Algorithms for the Continuity Equation," *J. Comp. Physics* , **44** , p. 84, 1981.

25. Forester, C.K., "Higher Order Monotonic Convective Difference Schemes," *J. Comp. Physics* , **23** , p. 1, 1977.

26. Davis, S.F., "TVD Finite Difference Schemes and Artificial Viscosity," ICASE Report No. 84-20, June, 1984.

27. Yee, H.C., Warming, R.F., and Harten, A., "Implicit Total Variation Diminishing (TVD) Schemes for Steady-State Calculations," AIAA Paper No. 83-1902, 1983.

28. Sweby, P.K., "High Resolution Schemes using Flux Limiters for Hyperbolic Conservation Laws," *SIAM J. Numer. Anal.* , **5** , p. 995, 1984.

29. Chakravarthy, S.R., and Osher, S., "A New Class of High Accuracy TVD Schemes for Hyperbolic Conservation Laws," AIAA Paper No. 85-0363, 1985.

30. Chakravarthy, S.R., and Osher, S., "Application of a New Class of High Accuracy TVD Schemes to the Navier-Stokes Equations," AIAA Paper No., 85-0165, 1985.

31. Book, D.L., Ott, E., and Sulton, A.L., "The Rayleigh-Taylor Instability in the Shallow-Water Approximation," *Phys. Fluids* , **17** , p. 676, 1974.

32. Anderson, D.V., "Axisymmetric Multifluid Simulation of High Beta Plasmas with Anisotropic Transport using a Moving Flux Coordinate Grid," *J. Comp. Physics* , **17** , p. 246, 1975.

33. Liewer, P.C., Krall, N.A., and Wagner, C.E., "Numerical Studies of Theta-pinch Implosions in Turbulent Plasmas," *Bull. Amer. Phys. Soc.* , **18** , p. 1366, 1975.

34. Book, D.L., Goldman, S.R., and Ossakow, S.L., "Interaction of a Small F-region Barium Cloud with the Neutral Wind," *EOS* , **55** , p. 1154, 1974.

35. Boris, J.P., Gardner, J.H., and Zalesak, S.T., "Atmosphere Hydrocodes using FCT Algorithms," NRL Report No. 3081, 1975.

36. Baer, M.R., and Gross, R.J., "A Numerical Study of Transport and Phase Interaction in Two Phase Conservation Equations," Sandia Report No. 84-1633, November, 1984.

37. Oran, E.S., Boris, J.P., Young, T., Flanigan, M., Burks, T. and Picone, M., "Numerical Simulations of Detonations in Hydrogen-Air and Methane-Air Mixtures," **Proceedings of the Eighteenth Symposium (International) on Combustion, The Combustion Institute,** p. 1641, 1981.

38. Oran, E.S., Young, T.R., Boris, J.P., Picone, J.M., and Edwards, D.H., "A Study of Detonation Structure: The Formation of Unreacted Gas Pockets," **Proceedings of the Nineteenth Symposium (International) on Combustion, The Combustion Institute,** p. 573, 1982.

39. Boris, J.P. Book, D.L. and Hain, K.H., "Flux-Corrected Transport. II: Generalizations of the Method," *J. Comp. Physics* , **18** , p. 248, 1975.

40. Boris, J.P. and Book, D.L., "Flux-Corrected Transport. III: Minimal-Error FCT Algorithms," *J. Comp. Physics* , **20** , p. 397, 1976.

41. Zalesak, S.T., "Fully Multidimensional Flux-Corrected Transport Algorithms for Fluids," *J. Comp. Physics* , **31** , p. 335, 1979.

42. Book, D.L, and Fry, M.A., "Airblast Simulations Using Flux-Corrected Transport Codes," NRL Report No. 5334, May, 1984.

43. Zalesak, S.T., "Very High Order and Pseudospectral Flux-Corrected Transport (FCT) Algorithms for Conservation Laws," **Proceedings of the Fourth IMACS International Symposium on Computer Methods for PDE's**, Lehigh University, Bethelehm, PA, 1981.

44. Wagner, D.H., "The Riemann Problem in two Space Dimensions for a single Conservation Law," *SIAM J. Math. Anal.* , **14** , p. 534, 1983.

45. Gross, R.J. and Baer, M.R., "ETBFCT - A Solver for One-Dimensional Transport Equations," Sandia Report No. 85-1273, July, 1985.

46. McDonald B.E. and Ambrosiano, J., "High-Order Upwind Flux Correction Methods for Hyperbolic Conservation Laws," *J. Appl. Physics* , **56** , p. 448, 1984.

47. Guckenheimer, J., "Shocks and rarefactions in two space dimensions," *Arch. Rational Mech. Anal.* , **59** , p. 281, 1975.

48. Carnahan, B., Luther, H.A., and Wilkes, J.O., **Applied Numerical Methods** , John Wiley and Sons, Inc., N.Y., p. 433, 1969.

49. Daly, B.J., "Numerical Study of Two-Fluid Rayleigh-Taylor Instability," *The Physics of Fluids* , **19** , 1967.

50. Boris, J.P., Coffey, T.P. and Fisher, S., "The Kelvin-Helmholtz Instability and Turbulent Mixing," NRL Report No. 3125, Sept., 1975.

51. Van Dyke, M., **An Album of Fluid Motion** , The Parabolic Press, Stanford, CA, 1982.

52. Ott, E., "Nonlinear Evolution of the Rayleigh-Taylor Instability of a Thin Layer," *Physical Review Letters* , **29** , 1972.

53. Markstein, G.H., "Experimental Studies of Flame-Front Instability," *Nonsteady Flame Propagation,* AGARDograph No. 25, Pergamon Press, Oxford, 1964.

54. Picone, J.M., Oran, E.S., Boris, J.P., and Young, J.R., "Theory of Vorticity Generation by Shock Waves and Flame Interactions," NRL Report No. 5366, July, 1984.

55. Harlow, F.M., "The Particle-in-Cell Computing Method for Fluid Dynamics," **Methods in Comp. Physics**, B. Alder *et. al.*, eds., Academic Press, N.Y., p. 319, 1964.

56. Harlow, F.H. and Welch, J.F., "Numerical Calculation of Time Dependent Viscous Incompressible Flow of Fluid with Free Surface," *Physics of Fluids* , **18** p. 2185, 1965.

57. Christiansen, J.P., "Numerical Simulation of Hydrodynamics by the Method of Point Vortices," *J. Comp. Physics* , **13** p. 363, 1973.

58. Leonard, A., "Vortex Methods for Flow Simulation," *J. Comp. Physics* , **37** p. 289, 1980.

59. Mansour, N.N., "A Hybrid Vortex-in-Cell Finite Difference Method for Shear Layer Calculations," Sandia National Laboratories, SAND84-8788, 1985.

Appendix A
A Fortran Listing of the FCT Subroutine QWIK2D

```
C******************************************************************
C
C                    *************************
C                    *       ISSUED BY       *
C                    *  SANDIA LABORATORIES, *
C                    *   A PRIME CONTRACTOR  *
C                    ********    TO THE      *
C                           *  UNITED STATES *
C                           *   DEPARTMENT   *
C                           *      OF        *
C                           *     ENERGY     *
C      ********************** ---NOTICE---   *********************
C      *THIS REPORT WAS PREPARED AS AN ACCOUNT OF WORK SPONSORED*
C      *  BY THE UNITED STATES GOVERNMENT.  NEITHER THE UNITED  *
C      *   STATES NOR THE UNITED STATES DEPARTMENT OF ENERGY,   *
C      *             NOR ANY OF THEIR EMPLOYEES,                *
C      * NOR ANY OF THEIR CONTRACTORS, SUBCONTRACTORS, OR THEIR *
C      * EMPLOYEES, MAKES ANY WARRANTY, EXPRESS OR IMPLIED, OR  *
C      * ASSUMES ANY LEGAL LIABILITY OR RESPONSIBILITY FOR THE  *
C      *           **********   ACCURACY,    **********          *
C      *           *        *  COMPLETENESS  *        *          *
C      *           *        *  OR USEFULNESS *        *          *
C      *           *        *    OF ANY      *        *          *
C      *           *        *  INFORMATION,  *        *          *
C      *           *        *   APPARATUS,   *        *          *
C      *        ****        *    PRODUCT     *        ****       *
C      *        *           *   OR PROCESS   *           *       *
C      *        *           *   DISCLOSED,   *           *       *
C      *        *           *  OR REPRESENTS *           *       *
C      *        *          **    THAT ITS    **          *       *
C      *        *          **  USE WOULD NOT **          *       *
C      *********           **    INFRINGE    **          *********
C                          **    PRIVATELY   **
C                          **      OWNED     **
C                          **     RIGHTS.    **
C                          **                **
C                          **                **
C                          **                **
C                          ********************
C
      SUBROUTINE QWIK2D(RHOO,RHON,SOURCE,LXBC,RXBC,LYBC,RYBC)
C
C**************************************************************
C
C     SUBROUTINE FCT2D IS A TWO-DIMENSIONAL NUMERICAL SOLVER
C     WHICH USES THE FLUX-CORRECTED TRANSPORT TECHNIQUE OF
C     BORIS AND BOOK.  THE APPROACH IS ANALOGOUS TO
C     THAT USED IN BORIS' FCT CODE - ETBFCT
C
C**************************************************************
C    THE EQUATION SOLVED:
C
C        D(R)  + D(U R) + D(V R) = SOURCE
```

```
C        DT      DX       DY
C
C     WHERE U AND V ARE VELOCITY COMPONENTS
C     IN THE X AND Y DIRECTIONS.
C
C*************************************************************
C     THE GRID SYSTEM IS NX BY NY.
C
C     PRIOR TO CALLING THIS SUBPROGRAM, THE SUBROUTINE VELOCE
C     MUST BE CALLED TO CALCULATE THE VELOCITY-RELATED VECTORS.
C     THESE VECTORS ARE THEN PASSED THROUGH COMMON BLK3.
C
C     THE INTERFACIAL AREAS AND VOLUMES OF THE GRID NETWORK
C     MUST BE FIRST DEFINED IN THE MAIN DRIVER AND THEN PASSED
C     IN COMMON BLK4.
C
C     BOUNDARY CONDITIONS ARE SPECIFIED IN THE MAIN DRIVER
C     AND PASSED BY THE ARGUMENT LIST USING THE VECTORS
C     LXBC, RXBC, LYBC AND RYBC.
C
        LXBC VECTOR DEFINES B.C. AT BOTTOM (IN X DIRECTION)
C       RXBC VECTOR DEFINES B.C. AT TOP    (IN X DIRECTION)
C       LYBC VECTOR DEFINES B.C. AT LEFT   (IN Y DIRECTION)
C       RYBC VECTOR DEFINES B.C. AT RIGHT  (IN Y DIRECTION)
C
C     BOUNDARY CONDITIONS ARE SET TO SYMMETRIC,REFLECTIVE
C     OR PERIODIC. SEE THE GBCOND SUBROUTINE IN ANOTHER APPENDIX.
C
C     SOURCE TERMS ARE TAKEN AS FOLLOWS:
C
C     SOURCE = C(X,Y) D(F) + D(X,Y) D(G) + S(X,Y)
C                     DX            DY
C
C********************************************************************
C
C***  DIMENSION AND DEFINE ALL OF THE ARRAYS USED IN THE
C***  SOLVER
C
      PARAMETER (NX=50, NXM1=NX-1, NY=50, NYM1=NY-1, MXNX=100)
      REAL MULH, NULH, LALH, KALH
      REAL LXBC(NX),RXBC(NX),LYBC(NY),RYBC(NY),FTDO(MXNX),FTDP(MXNX)
      REAL RHOO(NX,NY), RHON(NX,NY), SOURCE(NX,NY)
C
C***  INCLUDE THE NECESSARY COMMON BLOCKS
C
      COMMON /SHARE1/ BIGF(NX,NY), BIGG(NX,NY), TX(NX,NY), TY(NX,NY),
     1               SCRH1(NX,NY), SCRH2(NX,NY), RHOT(NX,NY)
C
      COMMON /BLK3/ ADUDTH(NX,NY), BDUDTH(NX,NY), MULH(NX,NY),
     1              LALH(NX,NY)  , NULH(NX,NY)  , KALH(NX,NY),
     2              GX(NX,NY)    , GY(NX,NY)
C
      COMMON /BLK4/ AH(NX,NY),BH(NX,NY),VOL(NX,NY),RVOL(NX,NY)
C
C***  DEFINE THE CONVECTIVE TRANSPORT FLUX AT THE MIDPOINTS
C
      DO 10 J=2,NYM1
      DO 10 I=1,NXM1
   10 SCRH1(I,J)= 0.5*ADUDTH(I,J)*(RHOO(I,J)+RHOO(I+1,J))
     1 - GX(I,J)*(RHOO(I+1,J)-RHOO(I,J))
C
```

```
      DO 20 I=2,NXM1
      DO 20 J=1,NYM1
   20 SCRH2(I,J)= 0.5*BDUDTH(I,J)*(RHOO(I,J)+RHOO(I,J+1))
     1 - GY(I,J)*(RHOO(I,J+1)-RHOO(I,J))
C
C***  TRANSPORT THE SOLUTION AND ADD THE SOURCE TERMS.
C
      DO 30 I=2,NXM1
      DO 30 J=2,NYM1
   30 RHON(I,J)= RHOO(I,J) + RVOL(I,J)*(- SCRH1(I,J) + SCRH1(I-1,J)
     1 - SCRH2(I,J) + SCRH2(I,J-1) + SOURCE(I,J) )
C
C*** CALL GBCOND TO SET BOUNDARY VALUES
C
      CALL GBCOND(RHON,LXBC,RXBC,LYBC,RYBC)
C
C***  DEFINE THE UNCORRECTED ANTIDIFFUSIVE FLUXES AT THE INTERIOR
C***  GRID POINTS AND THE DIFFUSIVE FLUX EMBEDDED IN A LOW ORDER SCHEME
C
      DO 40 J=2,NYM1
      DO 40 I=1,NXM1
      BIGF(I,J)= MULH(I,J)*(RHON(I+1,J) - RHON(I,J))
   40 SCRH1(I,J)= NULH(I,J)*(RHOO(I+1,J) - RHOO(I,J))
C
      DO 50 I=2,NXM1
      DO 50 J=1,NYM1
      BIGG(I,J)= KALH(I,J)*(RHON(I,J+1) - RHON(I,J))
   50 SCRH2(I,J)= LALH(I,J)*(RHOO(I,J+1) - RHOO(I,J))
C
C***  CALCULATE THE TRANSPORTED AND DIFFUSED SOLUTION
C
      DO 60 I=2,NXM1
      DO 60 J=2,NYM1
   60 RHOT(I,J)= RHON(I,J) + RVOL(I,J)*(SCRH1(I,J) - SCRH1(I-1,J)
     1                                + SCRH2(I,J) - SCRH2(I,J-1))
C
C*** CALL GBCOND TO SET BOUNDARY VALUES
C
      CALL GBCOND(RHOT,LXBC,RXBC,LYBC,RYBC)
C
C***  PERFORM THE STRONG FLUX LIMITER IN SPLIT
C***  DIRECTIONS ON THE ANTIDIFFUSION FLUX
C
      DO 70 J=1,NY
      DO 70 I=2,NXM1
      SCRH1(I,J)= (RHOT(I+1,J) - RHOT(I,J))*VOL(I,J)
   70 SCRH2(I,J)= (RHOT(I,J) - RHOT(I-1,J))*VOL(I,J)
C
      DO 80 J=1,NY
      FTDO(J)= (1.0-ABS(LYBC(J)))*RHOT(NX-2,J) + LYBC(J)*RHOT(3,J)
      FTDP(J)= RYBC(J)*RHOT(NX-2,J) + (1.0-ABS(RYBC(J)))*RHOT(3,J)
      SCRH2(1,J)= (RHOT(1,J) - FTDO(J))*VOL(1,J)
   80 SCRH1(NX,J)= (FTDP(J) - RHOT(NX,J))*VOL(NX,J)
C
      DO 90 J=2,NYM1
      DO 90 I=1,NXM1
      RHON(I,J)= SIGN(1.0,BIGF(I,J))
      BIGF(I,J)= ABS(BIGF(I,J))
      TY(I,J)= RHON(I,J)*SCRH2(I,J)
      BIGF(I,J)= AMIN1( BIGF(I,J), TY(I,J) )
      TY(I,J)= RHON(I,J)*SCRH1(I+1,J)
      BIGF(I,J)= AMIN1( BIGF(I,J), TY(I,J) )
      BIGF(I,J)= AMAX1( BIGF(I,J), 0.0 )
   90 TX(I,J)= BIGF(I,J)*RHON(I,J)
C
      DO 100 J=2,NYM1
      DO 100 I=1,NX
      SCRH1(I,J)= (RHOT(I,J+1) - RHOT(I,J))*VOL(I,J)
  100 SCRH2(I,J)= (RHOT(I,J) - RHOT(I,J-1))*VOL(I,J)
C
      DO 110 I=1,NX
      FTDO(I)= (1.0-ABS(LXBC(I)))*RHOT(I,NY-2) + LXBC(I)*RHOT(I,3)
      FTDP(I)= RXBC(I)*RHOT(I,NY-2) + (1.0-ABS(RXBC(I)))*RHOT(I,3)
      SCRH2(I,1)= (RHOT(I,1) - FTDO(I))*VOL(I,1)
  110 SCRH1(I,NY)= (FTDP(I) - RHOT(I,NY))*VOL(I,NY)
C
      DO 120 J=1,NYM1
      DO 120 I=2,NXM1
      RHON(I,J)= SIGN(1.0,BIGG(I,J))
      BIGG(I,J)= ABS(BIGG(I,J))
      TY(I,J)=  RHON(I,J)*SCRH2(I,J)
      BIGG(I,J)= AMIN1( BIGG(I,J), TY(I,J) )
      TY(I,J)=  RHON(I,J)*SCRH1(I,J+1)
      BIGG(I,J)= AMIN1( BIGG(I,J), TY(I,J) )
      BIGG(I,J)= AMAX1( BIGG(I,J), 0.0 )
  120 TY(I,J)= BIGG(I,J)*RHON(I,J)
C
C***  CALCULATE THE FINAL SOLUTION BY ANTIDIFFUSING
C***  THE TRANSPORTED AND DIFFUSED SOLUTION
C
      DO 130 I=2,NXM1
      DO 130 J=2,NYM1
  130 RHON(I,J)= RHOT(I,J) - RVOL(I,J)*( TX(I,J) - TX(I-1,J)
     1 + TY(I,J) - TY(I,J-1))
C
C*** SET BOUNDARY VALUES USING GBCOND
C
      CALL GBCOND(RHON,LXBC,RXBC,LYBC,RYBC)
C
      RETURN
      END
```

Appendix B
A Fortran Listing of the FCT Subroutine FCT2D and the Flux-limiting Subroutine FLIMIT

```
C*******************************************************************
C
C                    *************************
C                    *       ISSUED BY       *
C                    *  SANDIA LABORATORIES, *
C                    *   A PRIME CONTRACTOR  *
C                    ********     TO THE     *
C                           *  UNITED STATES *
C                           *   DEPARTMENT   *
C                           *       OF       *
C                           *     ENERGY     *
C      **********************  ---NOTICE---  ********************
C      *THIS REPORT WAS PREPARED AS AN ACCOUNT OF WORK SPONSORED*
C      *  BY THE UNITED STATES GOVERNMENT.  NEITHER THE UNITED  *
C      *   STATES NOR THE UNITED STATES DEPARTMENT OF ENERGY,   *
C      *               NOR ANY OF THEIR EMPLOYEES,              *
C      * NOR ANY OF THEIR CONTRACTORS, SUBCONTRACTORS, OR THEIR *
C      * EMPLOYEES, MAKES ANY WARRANTY, EXPRESS OR IMPLIED, OR  *
C      * ASSUMES ANY LEGAL LIABILITY OR RESPONSIBILITY FOR THE  *
C      *           **********   ACCURACY,    **********         *
C      *           *        *  COMPLETENESS  *        *         *
C      *           *        *  OR USEFULNESS *        *         *
C      *           *        *     OF ANY     *        *         *
C      *           *        *  INFORMATION,  *        *         *
C      *           *        *   APPARATUS,   *        *         *
C      *        ****        *     PRODUCT    *        ****      *
C      *        *           *   OR PROCESS   *           *      *
C      *        *           *   DISCLOSED,   *           *      *
C      *        *           *  OR REPRESENTS *           *      *
C      *        *          **   THAT ITS     **          *      *
C      *        *          **  USE WOULD NOT **          *      *
C      *********           **   INFRINGE     **         *********
C                          **   PRIVATELY    **
C                          **     OWNED      **
C                          **    RIGHTS.     **
C                          **                **
C                          **                **
C                          **                **
C                          ********************
C
      SUBROUTINE FCT2D(RHOO,RHON,SOURCE,LXBC,RXBC,LYBC,RYBC)
C
C*************************************************************
C
C     SUBROUTINE FCT2D IS A TWO-DIMENSIONAL NUMERICAL SOLVER
C     WHICH USES THE FLUX-CORRECTED TRANSPORT TECHNIQUE OF
C     BORIS AND BOOK AND THE FLUX-LIMITER BY ZALESAK.
C     THE APPROACH IS ANALOGOUS TO THAT USED IN BORIS' FCT
C     CODE - ETBFCT.
C
C     THIS SUBROUTINE ALSO CALLS SUBROUTINES FLIMIT AND GBCOND.
C     THESE ROUTINES MUST BE INCLUDED IN ORDER FOR THIS ROUTINE
C     TO EXECUTE.
```

```
C
C*****************************************************************
C    THE EQUATION SOLVED:
C
C        D(R)  + D(U R) + D(V R) = SOURCE
C        DT      DX       DY
C
C     WHERE U AND V ARE VELOCITY COMPONENTS
C     IN THE X AND Y DIRECTIONS.
C
C*****************************************************************
C     THE GRID SYSTEM IS NX BY NY.
C
C     PRIOR TO CALLING THIS SUBPROGRAM, THE SUBROUTINE VELOCE
C     MUST BE CALLED TO CALCULATE THE VELOCITY-RELATED VECTORS.
C     THESE VECTORS ARE THEN PASSED THROUGH COMMON BLK3.
C
C     THE INTERFACIAL AREAS AND VOLUMES OF THE GRID NETWORK
C     MUST BE FIRST DEFINED IN THE MAIN DRIVER AND THEN PASSED
C     IN COMMON BLK4.
C
C     BOUNDARY CONDITIONS ARE SPECIFIED IN THE MAIN DRIVER
C     AND PASSED BY THE ARGUMENT LIST USING THE VECTORS
C     LXBC, RXBC, LYBC AND RYBC.
C
C       LXBC VECTOR DEFINES B.C. AT BOTTOM (IN X DIRECTION)
C       RXBC VECTOR DEFINES B.C. AT TOP    (IN X DIRECTION)
C       LYBC VECTOR DEFINES B.C. AT LEFT   (IN Y DIRECTION)
C       RYBC VECTOR DEFINES B.C. AT RIGHT  (IN Y DIRECTION)
C
C     BOUNDARY CONDITIONS ARE SET TO SYMMETRIC,REFLECTIVE
C     OR PERIODIC. SEE THE GBCOND SUBROUTINE IN ANOTHER APPENDIX.
C
C     SOURCE TERMS ARE TAKEN AS FOLLOWS:
C
C     SOURCE = C(X,Y) D(F) + D(X,Y) D(G) + S(X,Y)
C                     DX            DY
C
C***********************************************************************
C
C***  DIMENSION AND DEFINE ALL OF THE ARRAYS USED IN THE
C***  SOLVER
C
      PARAMETER (NX=50, NXM1=NX-1, NY=50, NYM1=NY-1)
      REAL MULH, NULH, LALH, KALH
      REAL LXBC(NX),RXBC(NX),LYBC(NY),RYBC(NY)
      REAL RHOO(NX,NY), RHON(NX,NY), LNRHOT(NX,NY), SOURCE(NX,NY)
C
C***  INCLUDE THE NECESSARY COMMON BLOCKS
C
      COMMON /SHARE1/ BIGF(NX,NY), BIGG(NX,NY), TX(NX,NY), TY(NX,NY),
     1                SCRH1(NX,NY), SCRH2(NX,NY), RHOT(NX,NY)
C
      COMMON /BLK3/ ADUDTH(NX,NY), BDUDTH(NX,NY), MULH(NX,NY),
     1              LALH(NX,NY)  , NULH(NX,NY)  , KALH(NX,NY),
     2              GX(NX,NY)    , GY(NX,NY)
C
      COMMON /BLK4/ AH(NX,NY),BH(NX,NY),VOL(NX,NY),RVOL(NX,NY)
C
C***  DEFINE THE CONVECTIVE TRANSPORT FLUX AT THE MIDPOINTS
C
```

```
      DO 10 J=2,NYM1
      DO 10 I=1,NXM1
   10 SCRH1(I,J)= 0.5*ADUDTH(I,J)*(RHOO(I,J)+RHOO(I+1,J))
     1 - GX(I,J)*(RHOO(I+1,J)-RHOO(I,J))
C
      DO 20 I=2,NXM1
      DO 20 J=1,NYM1
   20 SCRH2(I,J)= 0.5*BDUDTH(I,J)*(RHOO(I,J)+RHOO(I,J+1))
     1 - GY(I,J)*(RHOO(I,J+1)-RHOO(I,J))
C
C***  TRANSPORT THE SOLUTION AND ADD THE SOURCE TERMS.
C
      DO 30 I=2,NXM1
      DO 30 J=2,NYM1
   30 RHON(I,J)= VOL(I,J)*RHOO(I,J) - SCRH1(I,J) + SCRH1(I-1,J)
     1 - SCRH2(I,J) + SCRH2(I,J-1) + SOURCE(I,J)
C
C*** CALL GBCOND TO SET BOUNDARY VALUES
C
      CALL GBCOND(RHON,LXBC,RXBC,LYBC,RYBC)
C
C***  DEFINE THE UNCORRECTED ANTIDIFFUSIVE FLUXES AT THE INTERIOR
C***  GRID POINTS
C
      DO 40 I=1,NXM1
      DO 40 J=1,NYM1
      BIGF(I,J)= MULH(I,J)*(RHON(I+1,J)*RVOL(I+1,J)
     1 - RHON(I,J)*RVOL(I,J))
   40 BIGG(I,J)= KALH(I,J)*(RHON(I,J+1)*RVOL(I,J+1)
     1 - RHON(I,J)*RVOL(I,J))
C
      DO 50 I=1,NXM1
   50 BIGF(I,NY)= MULH(I,NY)*(RHON(I+1,NY)*RVOL(I+1,NY)
     1 - RHON(I,NY)*RVOL(I,NY))
C
      DO 60 J=1,NYM1
   60 BIGG(NX,J)= KALH(NX,J)*(RHON(NX,J+1)*RVOL(NX,J+1)
     1 - RHON(NX,J)*RVOL(NX,J))
C
C***  DEFINE THE DIFFUSIVE FLUXES
C
      DO 70 I=1,NXM1
      DO 70 J=2,NYM1
   70 SCRH1(I,J)= NULH(I,J)*(RHOO(I+1,J) - RHOO(I,J))
C
      DO 80 I=2,NXM1
      DO 80 J=1,NYM1
   80 SCRH2(I,J)= LALH(I,J)*(RHOO(I,J+1) - RHOO(I,J))
C
C***  CALCULATE THE TRANSPORTED AND DIFFUSED SOLUTION
C
      DO 90 I=2,NXM1
      DO 90 J=2,NYM1
   90 LNRHOT(I,J)= RHON(I,J) + SCRH1(I,J) - SCRH1(I-1,J)
     1                        + SCRH2(I,J) - SCRH2(I,J-1)
C
C*** CALL GBCOND TO SET BOUNDARY VALUES

      CALL GBCOND(LNRHOT,LXBC,RXBC,LYBC,RYBC)
C
C***  DIVIDE OUT THE GRID VOLUME
```

```
C
      DO 100 I=1,NX
      DO 100 J=1,NY
  100 RHOT(I,J)= RVOL(I,J)*LNRHOT(I,J)
C
C***  CALL THE FLUX LIMITER SUBROUTINE WHICH CORRECTS THE
C***  ANTIDIFFUSION
C
      CALL FLIMIT(BIGF,BIGG,RHOT,LXBC,RXBC,LYBC,RYBC,TX,TY,RHOO)
C
C***  CALCULATE THE FINAL SOLUTION BY ANTIDIFFUSING THE
C***  TRANSPORTED AND DIFFUSED SOLUTION
C
C
      DO 110 I=2,NXM1
      DO 110 J=2,NYM1
  110 RHON(I,J)=RHOT(I,J)-RVOL(I,J)*(TX(I,J)-TX(I-1,J)
     1+TY(I,J)-TY(I,J-1))
C
C*** SET BOUNDARY VALUES USING GBCOND
C
      CALL GBCOND(RHON,LXBC,RXBC,LYBC,RYBC)
C
      RETURN
      END
C
C***************************************************************************
C
C     FLIMIT IS A VECTORIZED ASC FORTRAN MODULE WHICH IMPLEMENTS ZALESAK'S
C     MULTIDIMENSIONAL FLUX LIMITER [JCP 31, 335 (1979)] IN 2D CARTESIAN
C     GEOMETRY.  IT INCORPORATES AS OPTION A "PRE-LIMITING" STEP UTILIZING
C     THE STRONG 1-D LIMITER OF BORIS AND BOOK (WHEN JPRLIM = .TRUE. ),
C     AND THE ABILITY TO LOOK BACK TO THE PREVIOUS TIMESTEP FOR UPPER AND
C     LOWER BOUNDS ON THE NEW SOLUTION ( WHEN FOLD = .TRUE. ).
C
C
C                                UTILIZATION
C
C     FLIMIT                   - SUBROUTINE CALLED AT EVERY TIMESTEP FOR EACH
C                                CONVECTIVE EQUATION BEING SOLVED
C
C     AUXILIARY ROUTINES CALLED BY ABOVE:
C
C     GBCOND                   - APPLY SIMPLE PERIODIC OR REFLECTING BOUNDARY
C                                CONDITIONS TO SCRATCH AND INTERMEDIATE ARRAYS
C                                AFTER INTERIOR VALUES ARE KNOWN
C
C
C                                CALLING SEQUENCE
C
C     FLX, FLY                 - RAW (UNLIMITED) ANTIDIFFUSIVE FLUXES. DIMENSIONALLY
C                                THESE FLUXES SHOULD BE IN THE SAME UNITS AS FTD
C                                (BELOW) MULTIPLIED BY AN AREA
C
C     FTD                      - ARRAY CONTAINING THE TIME-ADVANCED, LOW ORDER
C                                ("TRANSPORTED AND DIFFUSED") SOLUTION
C
C     NX, NY                   - DIMENSIONS OF MESH (RESTRICTED AT COMPILE TIME
C                                BY VARIABLE DIMENSIONING TO NX <= MX AND NY <= MY,
C                                WHERE MX AND MY ARE DEFINED IN PARAMETER STATEMENT
C
```

```
C     SX, SY, SM          - SCRATCH ARRAYS
C
C     TX, TY              - CORRECTED ANTIDIFFUSIVE FLUXES RETURNED BY FLIMIT
C
C     FAA                 - ARRAY CONTAINING THE SOLUTION FROM THE PREVIOUS
C                           TIMESTEP (USED ONLY IF FOLD = .TRUE. )
C
C     LXBC,RXBC           - BOUNDARY CONDITION VECTORS ( SEE COMMENTS IN
C     LYBC,RYBC                     SUBROUTINE GBCOND )
C
C
C     THE COMMON BLOCK /LIMIT/ CONTAINS THREE SCALAR LOGICAL VARIABLES
C     WHICH MAY BE SET BY THE USER FROM OUTSIDE THE SUBROUTINE:
C
C     PRLIM                 SETTING PRLIM = .TRUE. ZEROES THE FLUX WHENEVER
C                           ITS SIGN DIFFERS FROM THAT OF THE CORRESPONDING
C                           FIRST DIFFERENCE IN FTD - IT IS A SIMPLIFIED VERSION
C                           OF EQ (14') IN JCP 31, PG 349.
C
C     JPRLIM                SETTING JPRLIM = .TRUE. CAUSES THE FLUXES TO BE
C                           "PRE-LIMITED" USING THE STRONG 1-D LIMITER OF BORIS
C                           AND BOOK BEFORE PASSING THE RESIDUAL FLUXES ON TO
C                           THE MULTIDIMENSIONAL LIMITER - SEE JCP 31, PP 349-350
C
C     FOLD                  SETTING FOLD = .TRUE. ALLOWS THE LIMITER TO LOOK BACK
C                           TO THE SOLUTION FROM THE PREVIOUS TIMESTEP (WHICH
C                           MUST BE STORED IN ARRAY FAA ) TO FIND UPPER AND
C                           LOWER BOUNDS ON THE NEW SOLUTION
C
C     THE MOST CONSERVATIVE (I.E., MOST DIFFUSIVE) CHOICE IS OBTAINED BY SETTING
C     JPRLIM = .TRUE. AND FOLD = .FALSE. ( PRLIM IS IRRELEVANT WHEN JPRLIM
C     IS .TRUE. ) THIS IS THE RECOMMENDED CHOICE FOR THE FIRST ATTEMPT.
C
C     THE BOUNDARY CONDITION SUBROUTINE GBCOND CAN BE REPLACED BY A MORE
C     COMPLICATED PROBLEM-DEPENDENT PRESCRIPTION WHICH VARIES AS A FUNCTION
C     OF THE ARRAY BEING BOUNDED.
C
      SUBROUTINE FLIMIT(FLX,FLY,FTD,LXBC,RXBC,LYBC,RYBC,TX,TY,FAA)
C
      PARAMETER (MXNX=100, NX=100, NXM=NX-1, NY=100, NYM=NY-1)
      LOGICAL PRLIM, JPRLIM, FOLD
      REAL LXBC(NX),RXBC(NX),LYBC(NY),RYBC(NY)
      REAL ALXBC(NX),ARXBC(NX),ALYBC(NY),ARYBC(NY)
      REAL PLXBC(NX),PRXBC(NX),PLYBC(NY),PRYBC(NY)
      REAL FTDO(MXNX), FTDP(MXNX)
      DIMENSION FLX(NX,NY), FLY(NX,NY), FTD(NX,NY), TX(NX,NY),
     1 TY(NX,NY), SX(NX,NY), SY(NX,NY), FAA(NX,NY), SM(NX,NY)
C
      COMMON /BLK4/ AH(NX,NY),BH(NX,NY),VOL(NX,NY),RVOL(NX,NY)
      COMMON /LIMIT/ PRLIM, JPRLIM, FOLD
C
C     NOTE THAT FLX AND FLY MUST BE REAL FLUXES (LIKE GRAMS FOR EX.)
C     FLX(I,J) IS CENTERED BETWEEN FTD(I,J) AND FTD(I+1,J)
C     FLY(I,J) IS CENTERED BETWEEN FTD(I,J) AND FTD(I,J+1)
C     FLX(I,J) DEFINED I=1,NXM  J=1,NY
C     FLY(I,J) DEFINED J=1,NYM  I=1,NX
C
C** SET VECTORS FOR BOUNDARY CONDITIONS
C
      DO 1 I=1,NX
      ALXBC(I)=ABS(LXBC(I))
      ARXBC(I)=ABS(RXBC(I))
      PLXBC(I)=1.-ALXBC(I)
    1 PRXBC(I)=1.-ARXBC(I)
C
      DO 5 J=1,NY
      ALYBC(J)=ABS(LYBC(J))
      ARYBC(J)=ABS(RYBC(J))
      PLYBC(J)=1.-ALYBC(J)
    5 PRYBC(J)=1.-ARYBC(J)
C
C
      IF ( .NOT. JPRLIM ) GO TO 70
C
      DO 10 J=1,NY
      DO 10 I=2,NXM
      SX(I,J)= (FTD(I+1,J) - FTD(I,J))*VOL(I,J)
   10 SY(I,J)= (FTD(I,J) - FTD(I-1,J))*VOL(I,J)
C
      DO 20 J=1,NY
      FTDO(J)= PLYBC(J)*FTD(NX-2,J) + LYBC(J)*FTD(3,J)
      FTDP(J)= RYBC(J)*FTD(NX-2,J) + PRYBC(J)*FTD(3,J)
      SY(1,J)= (FTD(1,J) - FTDO(J))*VOL(1,J)
   20 SX(NX,J)= (FTDP(J) - FTD(NX,J))*VOL(NX,J)
C
      DO 30 J=1,NY
      DO 30 I=1,NXM
      SM(I,J)= SIGN(1.0,FLX(I,J))
      FLX(I,J)= ABS(FLX(I,J))
      TY(I,J)= SM(I,J)*SY(I,J)
      FLX(I,J)= AMIN1( FLX(I,J), TY(I,J) )
      TY(I,J)= SM(I,J)*SX(I+1,J)
      FLX(I,J)= AMIN1( FLX(I,J), TY(I,J) )
      FLX(I,J)= AMAX1( FLX(I,J), 0.0 )
   30 FLX(I,J)= FLX(I,J)*SM(I,J)
C
      DO 40 J=2,NYM
      DO 40 I=1,NX
      SX(I,J)= (FTD(I,J+1) - FTD(I,J))*VOL(I,J)
   40 SY(I,J)= (FTD(I,J) - FTD(I,J-1))*VOL(I,J)
C
      DO 50 I=1,NX
      FTDO(I)= PLXBC(I)*FTD(I,NY-2) + LXBC(I)*FTD(I,3)
      FTDP(I)= RXBC(I)*FTD(I,NY-2) + PRXBC(I)*FTD(I,3)
      SY(I,1)= (FTD(I,1) - FTDO(I))*VOL(I,1)
   50 SX(I,NY)= (FTDP(I) - FTD(I,NY))*VOL(I,NY)
C
      DO 60 J=1,NYM
      DO 60 I=1,NX
      SM(I,J)= SIGN(1.0,FLY(I,J))
      FLY(I,J)= ABS(FLY(I,J))
      TY(I,J)=  SM(I,J)*SY(I,J)
      FLY(I,J)= AMIN1( FLY(I,J), TY(I,J) )
      TY(I,J)=  SM(I,J)*SX(I,J+1)
      FLY(I,J)= AMIN1( FLY(I,J), TY(I,J) )
      FLY(I,J)= AMAX1( FLY(I,J), 0.0 )
   60 FLY(I,J)= FLY(I.J)*SM(I,J)
C
   70 CONTINUE
C
      IF ( .NOT. PRLIM ) GO TO 100
C
```

```
      DO 80 J=1,NY
      DO 80 I=1,NXM
      SM(I,J)= FLX(I,J)*(FTD(I+1,J) - FTD(I,J))
   80 FLX(I,J)= FLX(I,J)*AMAX1(0.0,SIGN(1.0,SM(I,J)) )
C
      DO 90 J=1,NYM
      DO 90 I=1,NX
      SM(I,J)= FLY(I,J)*(FTD(I,J+1) - FTD(I,J))
   90 FLY(I,J)= FLY(I,J)*AMAX1(0.0,SIGN(1.0,SM(I,J)) )
C
  100 CONTINUE
C
      DO 110 J=1,NY
      DO 110 I=1,NX
      SX(I,J)= 1.0
  110 SY(I,J)= 1.0
C
      IF ( FOLD ) GO TO 130
C
      DO 120 J=1,NY
      DO 120 I=1,NX
  120 SM(I,J)= FTD(I,J)
C
      GO TO 150
C
  130 CONTINUE
      DO 140 J=1,NY
      DO 140 I=1,NX
  140 SM(I,J)= AMAX1(FTD(I,J),FAA(I,J))
C
  150 CONTINUE
      DO 160 J=2,NYM
      DO 160 I=2,NXM
      TX(I,J)= (AMAX1( SM(I-1,J), SM(I,J), SM(I+1,J), SM(I,J-1),
     1      SM(I,J+1)) - FTD(I,J) )*VOL(I,J)
  160 CONTINUE
C
      DO 170 J=2,NYM
      DO 170 I=2,NXM
  170 SM(I,J)= AMAX1(0.0,FLX(I-1,J)) - AMIN1(0.0,FLX(I,J))
C
      DO 180 J=2,NYM
      DO 180 I=2,NXM
      SM(I,J)= SM(I,J) + (AMAX1(0.0,FLY(I,J-1)) - AMIN1(0.0,FLY(I,J)))
  180 TY(I,J)= 1.0
C
      DO 185 J=2,NYM
      DO 185 I=2,NXM
      IF (SM(I,J) .LE. TX(I,J)) GO TO 185
      TY(I,J)= TX(I,J)/SM(I,J)
  185 CONTINUE
C
      DO 190 J=2,NYM
      DO 190 I=2,NXM
  190 TX(I,J)= TY(I,J)
C
      CALL GBCOND(TX, ALXBC, ARXBC, ALYBC, ARYBC)
C
      DO 200 J=1,NY
      DO 200 I=1,NXM
  200 SM(I,J)= -SIGN(TX(I,J),FLX(I,J))
```

```
C
     DO 210 J=1,NY
     DO 210 I=1,NXM
 210 SM(I,J)= AMAX1( SM(I,J), SIGN(TX(I+1,J),FLX(I,J)) )
C
     DO 220 J=1,NY
     DO 220 I=1,NXM
 220 SX(I,J)= AMIN1(SX(I,J),SM(I,J))
C
     DO 230 J=1,NYM
     DO 230 I=1,NX
 230 SM(I,J)= -SIGN(TX(I,J),FLY(I,J))
C
     DO 240 J=1,NYM
     DO 240 I=1,NX
 240 SM(I,J)= AMAX1( SM(I,J), SIGN(TX(I,J+1),FLY(I,J)) )
C
     DO 250 J=1,NYM
     DO 250 I=1,NX
 250 SY(I,J)= AMIN1(SY(I,J),SM(I,J))
C
     IF ( FOLD ) GO TO 270
C
     DO 260 J=1,NY
     DO 260 I=1,NX
 260 SM(I,J)= FTD(I,J)
C
     GO TO 290
C
 270 CONTINUE
     DO 280 J=1,NY
     DO 280 I=1,NX
 280 SM(I,J)= AMIN1(FTD(I,J),FAA(I,J))
C
 290 CONTINUE
     DO 300 J=2,NYM
     DO 300 I=2,NXM
     TX(I,J)= (FTD(I,J) - AMIN1( SM(I-1,J), SM(I,J), SM(I+1,J),
    1      SM(I,J-1), SM(I,J+1)) )*VOL(I,J)
 300 CONTINUE
C
     DO 310 J=1,NY
     DO 310 I=1,NXM
 310 TY(I,J)= SX(I,J)*FLX(I,J)
C
     DO 320 J=2,NYM
     DO 320 I=2,NXM
 320 SM(I,J)= AMAX1(0.0,TY(I,J)) - AMIN1(0.0,TY(I-1,J))
C
     DO 330 J=1,NYM
     DO 330 I=1,NX
 330 TY(I,J)= SY(I,J)*FLY(I,J)
C
     DO 340 J=2,NYM
     DO 340 I=2,NXM
     SM(I,J)= SM(I,J) + (AMAX1(0.0,TY(I,J)) - AMIN1(0.0,TY(I,J-1)) )
 340 TY(I,J) =1.0
C
     DO 350 J=2,NYM
     DO 350 I=2,NXM
     IF (SM(I,J) .LE. TX(I,J)) GO TO 350
```

```
      TY(I,J)= TX(I,J)/SM(I,J)
  350 CONTINUE
C
      DO 355 J=2,NYM
      DO 355 I=2,NXM
  355 TX(I,J)= TY(I,J)
C
      CALL GBCOND(TX, ALXBC, ARXBC, ALYBC, ARYBC)
C
      DO 360 J=1,NY
      DO 360 I=1,NXM
  360 SM(I,J)= SIGN(TX(I,J),FLX(I,J))
C
      DO 370 J=1,NY
      DO 370 I=1,NXM
  370 SM(I,J)= AMAX1( SM(I,J), -SIGN(TX(I+1,J),FLX(I,J)) )
C
      DO 380 J=1,NY
      DO 380 I=1,NXM
  380 SX(I,J)= SX(I,J)*SM(I,J)
C
      DO 390 J=1,NYM
      DO 390 I=1,NX
  390 SM(I,J)= SIGN(TX(I,J),FLY(I,J))
C
      DO 400 J=1,NYM
      DO 400 I=1,NX
  400 SM(I,J)= AMAX1( SM(I,J), -SIGN(TX(I,J+1),FLY(I,J)) )
C
      DO 410 J=1,NYM
      DO 410 I=1,NX
  410 SY(I,J)= SY(I,J)*SM(I,J)
C
      DO 420 J=1,NY
      DO 420 I=1,NXM
  420 TX(I,J)= FLX(I,J)*SX(I,J)
C
      DO 430 J=1,NYM
      DO 430 I=1,NX
  430 TY(I,J)= FLY(I,J)*SY(I,J)
C
      RETURN
      END
```

Appendix C
A Fortran Listing of the FCT Driver for the 2-D Riemann Problem and the Auxiliary Subroutines VELOCE and GBCOND

```
C******************************************************************
C
C                   *************************
C                   *       ISSUED BY       *
C                   *  SANDIA LABORATORIES, *
C                   *   A PRIME CONTRACTOR  *
C                   ********     TO THE     *
C                          *  UNITED STATES *
C                          *   DEPARTMENT   *
C                          *       OF       *
C                          *     ENERGY     *
C      *********************  ---NOTICE---  **********************
C      *THIS REPORT WAS PREPARED AS AN ACCOUNT OF WORK SPONSORED*
C      *  BY THE UNITED STATES GOVERNMENT.  NEITHER THE UNITED   *
C      *   STATES NOR THE UNITED STATES DEPARTMENT OF ENERGY,    *
C      *               NOR ANY OF THEIR EMPLOYEES,               *
C      * NOR ANY OF THEIR CONTRACTORS, SUBCONTRACTORS, OR THEIR  *
C      * EMPLOYEES, MAKES ANY WARRANTY, EXPRESS OR IMPLIED, OR   *
C      * ASSUMES ANY LEGAL LIABILITY OR RESPONSIBILITY FOR THE   *
C      *          **********    ACCURACY,   **********           *
C      *          *        *  COMPLETENESS  *        *           *
C      *          *        *  OR USEFULNESS *        *           *
C      *          *        *     OF ANY     *        *           *
C      *          *        *  INFORMATION,  *        *           *
C      *          *        *   APPARATUS,   *        *           *
C      *       ****        *     PRODUCT    *        ****        *
C      *       *           *   OR PROCESS   *           *        *
C      *       *           *   DISCLOSED,   *           *        *
C      *       *           *  OR REPRESENTS *           *        *
C      *       *          **    THAT ITS    **          *        *
C      *       *          **  USE WOULD NOT **          *        *
C      *********          **    INFRINGE    **          **********
C                         **    PRIVATELY   **
C                         **      OWNED     **
C                         **     RIGHTS.    **
C                         **                **
C                         **                **
C                         **                **
C                         ********************
C
      PROGRAM DRIVER
C********************************************************
C*** THIS PROGRAM TESTS THE 2-D FLUX CORRECTING        *
C*** ALGORITHM ON A RIEMANN PROBLEM IN THE INFINITE    *
C*** DOMAIN WITH CONSTANT VALUE INITIAL CONDITIONS IN  *
C*** EACH QUADRANT.  REFER TO THE WORK BY WAGNER.      *
C********************************************************
C
      PARAMETER (NX=50, NY=50, NXNY=NX*NY, SMALL=1.0E-20)
C
C*** COMMON BLK5, BLK7, AND BLK8 ARE USED IN THE SUBROUTINE
C*** WHICH COMPUTES THE EXACT SOLUTION. A LISTING OF THIS
C*** LATTER ROUTINE HAS NOT BEEN APPENDED.
```

```
C
      LOGICAL PRLIM,JPRLIM,FOLD
      REAL LXBC(NX),RXBC(NX),LYBC(NY),RYBC(NY),ZERO(NX,NY)
      DIMENSION U(NX,NY),V(NX,NY)
      COMMON /BLK4/ AH(NX,NY),BH(NX,NY),VOL(NX,NY),RVOL(NX,NY)
      COMMON /BLK5/ R(NX,NY), RN(NX,NY), REXACT(NX,NY)
      COMMON /BLK7/ XG(NX), XREF(NX), YG(NY), YREF(NY)
      COMMON /BLK8/ RQ1,RQ2,RQ3,RQ4,A1,A2,SUM,DEPRESS
      COMMON /LIMIT/ PRLIM,JPRLIM,FOLD
      DATA ZERO /NXNY*0.0/
C
C*****************************************************************
C
C*** SPECIFY NONDIMENSIONAL LENGTH OF THE BOUNDARIES IN
C*** BOTH DIRECTIONS, INITIALIZE A COUNTER, AND INITIALIZE
C*** A SUMMATION PARAMETER USED IN THE EXACT SOLUTION ROUTINE
C
      BOUNDX= 1.0
      BOUNDY= 1.0
      ICOUNT= 1
      RTOTTOT= 0.0
C
C******************************************************************
C
C*** SPECIFY NCASE, THE SELECTION OF EXACT SOLUTION DESIRED
C*** TO BE COMPUTED BY THE SUBROUTINE EXACT1
C***      NCASE=1 : SIMPLE SHOCK AND RAREFRACTION
C***      NCASE=2 : CASE OF FOUR SIMPLE SHOCK JUMPS
C***      NCASE=3 : A TRIPLE SHOCK WITH SHOCK STEM
C***      NCASE=4 : THE CUSP SHOCK
C
      READ(5,*) NCASE
      WRITE(6,10) NCASE
   10 FORMAT(1H ,"NCASE = ",I2/)
C
C******************************************************************
C
C*** SPECIFY INITIAL CONDITIONS BASED ON THE CHOICE OF NCASE
C
      IF (NCASE .EQ. 1) THEN
         RQ1= 0.0
         RQ2= 0.0
         RQ3= 0.0
         RQ4= 5.0
      ELSE IF (NCASE .EQ. 2) THEN
         RQ1= 0.5
         RQ2= 1.0
         RQ3= 3.0
         RQ4= 2.0
      ELSE IF (NCASE .EQ. 3) THEN
         RQ1= -2.0
         RQ2= -1.0
         RQ3= -0.5
         RQ4=  2.0
      ELSE IF (NCASE .EQ. 4) THEN
         RQ1=  1.0
         RQ2= -2.0
         RQ3=  1.0
         RQ4= -2.0
      END IF
      IF (NCASE .NE. 1) THEN
         A1= (RQ2-RQ1)/(RQ2-RQ3)
         A2= (RQ4-RQ3)/(RQ4-RQ1)
      END IF
C
C*************************************************************
C
C*** DEFINE GRID QUANTITIES. XREF AND YREF ARE USED IN THE
C*** EXACT SOLUTION SUBROUTINE. AH AND BH ARE THE AREAS,
C*** VOL IS THE VOLUME OF A CELL, AND RVOL IS THE RECIPROCAL
C*** VOLUME OF A CELL.
C
      DX= BOUNDX/FLOAT(NX-1)
      DY= BOUNDY/FLOAT(NY-1)
      DO 20 I=1,NX
   20 XG(I)= DX*(I-1)
      DO 30 J=1,NY
   30 YG(J)= DY*(J-1)
      DO 40 I=1,NX
   40 XREF(I)= XG(I) - 0.5*BOUNDX
      DO 50 J=1,NY
   50 YREF(J)= YG(J) - 0.5*BOUNDY
C
      DO 60 I=1,NX
      DO 60 J=1,NY
      AH(I,J)= DY
      BH(I,J)= DX
      VOL(I,J)= DX*DY
   60 RVOL(I,J)= 1.0/VOL(I,J)
C
C***************************************************************
C
C*** MAP THE INITIAL CONDITIONS ONTO THE 2-D GRID
C
      NXHALF= 0.5*NX
      NYHALF= 0.5*NY
C
      DO 70 I=1,NXHALF
      DO 70 J=1,NYHALF
   70 R(I,J)= RQ3
C
      DO 80 I=NXHALF+1,NX
      DO 80 J=1,NYHALF
   80 R(I,J)= RQ4
C
      DO 90 I=1,NXHALF
      DO 90 J=NYHALF+1,NY
   90 R(I,J)= RQ2
C
      DO 100 I=NXHALF+1,NX
      DO 100 J=NYHALF+1,NY
  100 R(I,J)= RQ1
C
C***************************************************************
C
C*** READ IN THE VALUES OF PRLIM, JPRLIM (WHICH CAN
C*** PRE-LIMIT THE FLUXES), AND FOLD (WHICH ALLOWS
C*** THE LIMITER TO LOOK BACK TO THE PREVIOUS SOLUTION)
C
      READ(5,110) PRLIM,JPRLIM,FOLD
  110 FORMAT(3L7)
```

```
      WRITE(6,120) PRLIM,JPRLIM,FOLD
  120 FORMAT(1H /5X,"PRLIM   = ",L7/5X,"JPRLIM = ",L7/5X,
     1 "FOLD    = ",L7//)
C
C*****************************************************************
C
C*** DEFINE THE BOUNDARY CONDITIONS
C
      DO 130 I=1,NX
      LXBC(I)= 1.0
  130 RXBC(I)= 1.0
C
      DO 140 J=1,NY
      LYBC(J)= 1.0
  140 RYBC(J)= 1.0
C
C*****************************************************************
C
C*** READ IN INITIAL TIME, SET THE COURANT NUMBER, THE MAXIMUM
C*** NUMBER OF TIME STEPS, THE PRINT COUNTER, THE PRINT
C*** INTERVAL, THE PLOT COUNTER, THE PLOT INTERVAL, AND THE
C*** SWITCH FOR FOURTH- OR SIXTH-ORDER ACCURATE PHASE ERRORS
C
      READ(5,*) TIME,COURANT,NSTEP,IPRINT,IPRINTX,IPLOT,IPLOTX,ISWIT
      WRITE(6,150) TIME,COURANT,NSTEP,IPRINT,IPRINTX,IPLOT,IPLOTX,ISWIT
  150 FORMAT(1H /5X,"INITIAL TIME = ",G10.4/5X,
     1 "COURANT NUMBER = ",G10.4/5X,"NSTEP = ",I6/5X,
     2 "PRINT COUNTER, IPRINT = ",I3/5X,"PRINT INTERVAL = ",I3/5X,
     3 "PLOT COUNTER, IPLOT = ",I3/5X,"PLOT INTERVAL = ",I3/5X,
     4 "SWITCH PARAMETER = ",I3//5X,"ISWIT = 0 IMPLIES FOURTH ORDER"
     5 /5X,"ISWIT = 1 IMPLIES SIXTH ORDER"//)
C
      IF (NX .EQ. 100) THEN
      WRITE(6,160)
  160 FORMAT(1H /5X,"COMPUTING ON A 100 BY 100 GRID"/)
      ELSE IF (NX .EQ. 50) THEN
      WRITE(6,170)
  170 FORMAT(1H /5X,"COMPUTING ON A 50 BY 50 GRID"/)
      END IF
C
C*****************************************************************
C
C*** BEGIN STEPPING THE SOLUTION
C
      DO 500 ISTEP=1,NSTEP
C
C*** ESTABLISH THE VELOCITY FIELD
C
      DO 180 I=1,NX
      DO 180 J=1,NY
      U(I,J)= R(I,J)
  180 V(I,J)= R(I,J)
C
C*** SET THE TIME STEP ACCORDING TO THE COURANT CONDITION
C
      UMAX= 0.0
      DO 190 I=1,NX
      DO 190 J=1,NY
      VELABS= ABS(U(I,J))
  190 UMAX= AMAX1(UMAX,VELABS)
```

```
      VMAX= 0.0
      DO 200 I=1,NX
      DO 200 J=1,NY
      VELABS= ABS(V(I,J))
  200 VMAX= AMAX1(VMAX,VELABS)
C
      DT1= DX*COURANT/(UMAX + SMALL)
      DT2= DY*COURANT/(VMAX + SMALL)
      DT=  AMIN1(DT1,DT2)
      DTH= 0.5*DT
C
C*** COMPUTATION AT HALF STEP
C
      CALL VELOCE(U,V,DTH,ISWIT)
      CALL QWIK2D(R,RN,ZERO,LXBC,RXBC,LYBC,RYBC)
C
C*** COMPUTATION TO FULL STEP
C
      DO 210 I=1,NX
      DO 210 J=1,NY
      U(I,J)= RN(I,J)
  210 V(I,J)= RN(I,J)
C
      CALL VELOCE(U,V,DT,ISWIT)
      CALL QWIK2D(R,RN,ZERO,LXBC,RXBC,LYBC,RYBC)
C
C*** SHIFT SOLUTION VECTOR
C
      DO 220 I=1,NX
      DO 220 J=1,NY
  220 R(I,J)= RN(I,J)
C
      TIME= TIME + DT
C
C******************************************************************
C
C*** PRINTOUT OF SOLUTION
C
      IPRINT= IPRINT+1
      IF (IPRINT .EQ. IPRINTX) THEN
         CALL EXACT1(ISTEP,DT,TIME,NXHALF,RTOTTOT,NCASE,ICOUNT)
         IPRINT= 1
      END IF
C
C******************************************************************
C
  500 CONTINUE
C
      END
C
C******************************************************************
C
      SUBROUTINE VELOCE(U,V,DT,ISWIT)
C
C     SUBROUTINE VELOCE COMPUTES DIFFUSION AND ANTIDIFFUSION
C     COEFFICIENTS AND THE CONVECTIVE MASS FLUXES OVER DENSITY
C
C      U,V ARE RESPECTIVELY THE VELOCITY COMPONENTS IN THE X AND Y
C      DIRECTION.
C
      PARAMETER (NX=50, NXM1=NX-1, NY=50, NYM1=NY-1)
      REAL MULH, NULH, LALH, KALH, U(NX,NY), V(NX,NY)
C
      COMMON /SHARE1/ BIGF(NX,NY), BIGG(NX,NY), TX(NX,NY), TY(NX,NY),
     1                SCRH1(NX,NY), SCRH2(NX,NY), RHOT(NX,NY)
      COMMON /BLK3/ ADUDTH(NX,NY), BDUDTH(NX,NY), MULH(NX,NY),
     1              LALH(NX,NY)  , NULH(NX,NY)  , KALH(NX,NY),
     2              GX(NX,NY)    , GY(NX,NY)
      COMMON /BLK4/ AH(NX,NY),BH(NX,NY),VOL(NX,NY),RVOL(NX,NY)
C
      DTH= 0.5*DT
C
      DO 100 I=1,NXM1
      DO 100 J=1,NY
      ADUDTH(I,J)= AH(I,J)*DTH*(U(I,J)+U(I+1,J))
  100 SCRH1(I,J)= 0.5*ADUDTH(I,J)*(RVOL(I,J)+RVOL(I+1,J))
C
      IF (ISWIT .EQ. 1) GO TO 115
C
      DO 110 I=1,NXM1
      DO 110 J=1,NY
C
C*** FOURTH-ORDER PHASE ERRORS -- ORIGINAL VERSION ******************
C
      GX(I,J)= 0.0
  110 NULH(I,J)=(0.5+SCRH1(I,J)*SCRH1(I,J))*0.5
     1                                      *(VOL(I,J)+VOL(I+1,J))/3.
C
      GO TO 125
C
  115 DO 120 I=1,NXM1
      DO 120 J=1,NY
C
C*** SIXTH-ORDER DISPERSIVE (PHASE) ERRORS **************************
C
      GX(I,J)= (1.0+SCRH1(I,J)*SCRH1(I,J))*0.5*
     1                                     (VOL(I,J)+VOL(I+1,J))/5.0
  120 NULH(I,J)= (-1.0 + 4.0*SCRH1(I,J)*SCRH1(I,J))*0.5*
     1                                      (VOL(I,J)+VOL(I+1,J))/30.0
C
  125 CONTINUE
C
C*******************************************************************
C
      DO 130 I=1,NXM1
      DO 130 J=1,NY
  130 MULH(I,J)=(1.0-SCRH1(I,J)*SCRH1(I,J))*0.5
     1                                      *(VOL(I,J)+VOL(I+1,J))/6.
C
      DO 140 I=1,NX
      DO 140 J=1,NYM1
      BDUDTH(I,J)= BH(I,J)*DTH*(V(I,J)+V(I,J+1))
  140 SCRH1(I,J)= 0.5*BDUDTH(I,J)*(RVOL(I,J)+RVOL(I,J+1))
C
      IF (ISWIT .EQ. 1) GO TO 160
C
      DO 150 I=1,NX
      DO 150 J=1,NYM1
C
C*** FOURTH-ORDER PHASE ERRORS -- ORIGINAL VERSION ******************
C
      GY(I,J)= 0.0
  150 LALH(I,J)=(0.5+SCRH1(I,J)*SCRH1(I,J))*0.5
     1                                     *(VOL(I,J)+VOL(I,J+1))/3.
C
      GO TO 180
C
  160 DO 170 I=1,NX
      DO 170 J=1,NYM1
C
C*** SIXTH-ORDER DISPERSIVE (PHASE) ERRORS ***************************
C
      GY(I,J)= (1.0+SCRH1(I,J)*SCRH1(I,J))*0.5
     1                                    *(VOL(I,J)+VOL(I,J+1))/5.0
  170 LALH(I,J)= (-1.0 + 4.0*SCRH1(I,J)*SCRH1(I,J))*0.5*
     1                                      (VOL(I,J)+VOL(I,J+1))/30.0
C
  180 CONTINUE
C
C*********************************************************************
C
      DO 190 I=1,NX
      DO 190 J=1,NYM1
  190 KALH(I,J)=(1.0-SCRH1(I,J)*SCRH1(I,J))*0.5
     1                                     *(VOL(I,J)+VOL(I,J+1))/6.
C
      RETURN
      END
C
C**********************************************************************
C
      SUBROUTINE GBCOND(F, LXBC, RXBC, LYBC, RYBC)
C
C     GBCOND IS A SIMPLE ROUTINE FOR SETTING BOUNDARY CONDITIONS
C     ON THE NX BY NY ARRAY OF F.
C
C     CONSIDER THE RECTANGULAR REGION:
C
C     LXBC IS THE VECTOR OF B.C. ON BOTTOM (IN X DIRECTION)
C     RXBC IS THE VECTOR OF B.C. ON TOP (IN X DIRECTION)
C     LYBC IS THE VECTOR OF LEFT B.C. (IN Y DIRECTION)
C     RYBC IS THE VECTOR OF RIGHT B.C. (IN Y DIRECTION)
C
C     THESE VECTORS TAKE ON VALUES -1.0, 0.0, AND 1.0:
C          RBC = -1.0 - ANTISYMMETRIC BOUNDARY CONDITIONS
C          RBC =  0.0 - PERIODIC BOUNDARY CONDITIONS
C          RBC =  1.0 - SYMMETRIC BOUNDARY CONDITIONS
C
      PARAMETER (NXNY=50, NX=50, NXM=NX-1, NY=50, NYM=NY-1)
      REAL LXBC(NX),RXBC(NX),LYBC(NY),RYBC(NY),BC1(NXNY),BC2(NXNY)
      DIMENSION F(NX,NY)
C
      DO 10 J=2,NYM
      BC1(J)=1.-LYBC(J)*LYBC(J)
      BC2(J)=1.-RYBC(J)*RYBC(J)
      F(1,J)=  BC1(J)*F(NXM,J) +LYBC(J)*F(2,J)
   10 F(NX,J)= RYBC(J)*F(NXM,J) + BC2(J)*F(2,J)
C
      DO 20 I=1,NX
      BC1(I)=1.-LXBC(I)*LXBC(I)
      BC2(I)=1.-RXBC(I)*RXBC(I)
      F(I,1)= BC1(I)*F(I,NYM) + LXBC(I)*F(I,2)
   20 F(I,NY)= RXBC(I)*F(I,NYM) + BC2(I)*F(I,2)
      RETURN
      END
```

Appendix D

Error-measure results for each of the four 2-D Riemann examples.

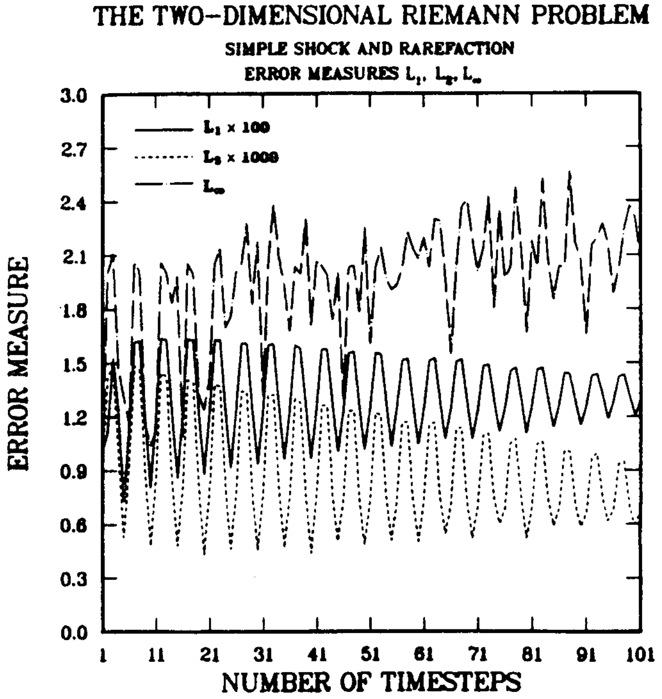

Figure D1

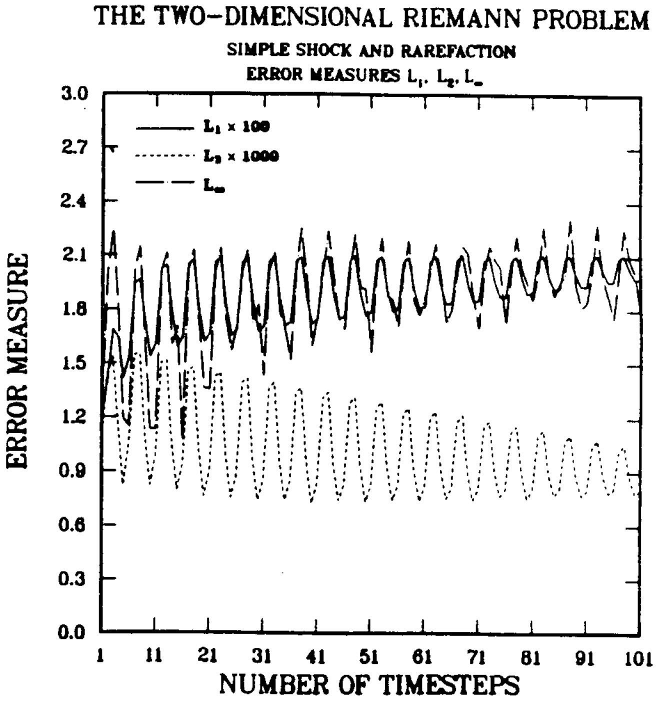

Figure D2

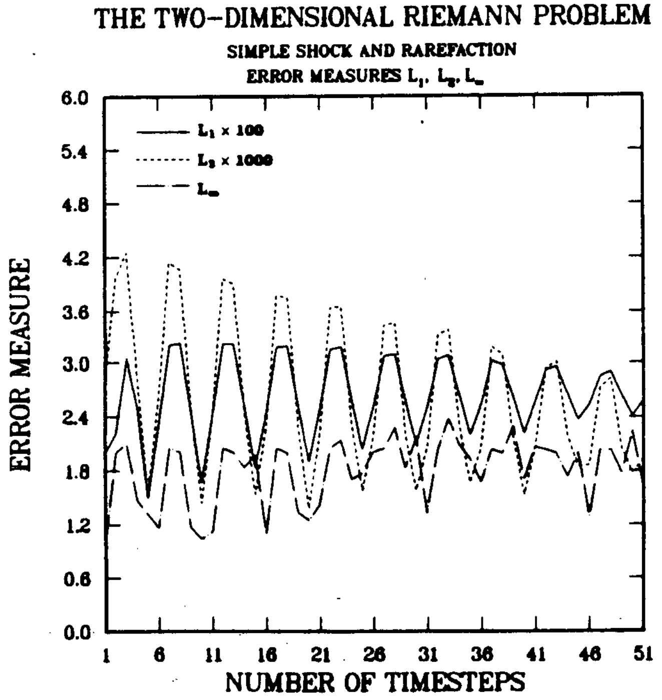

Figure D3

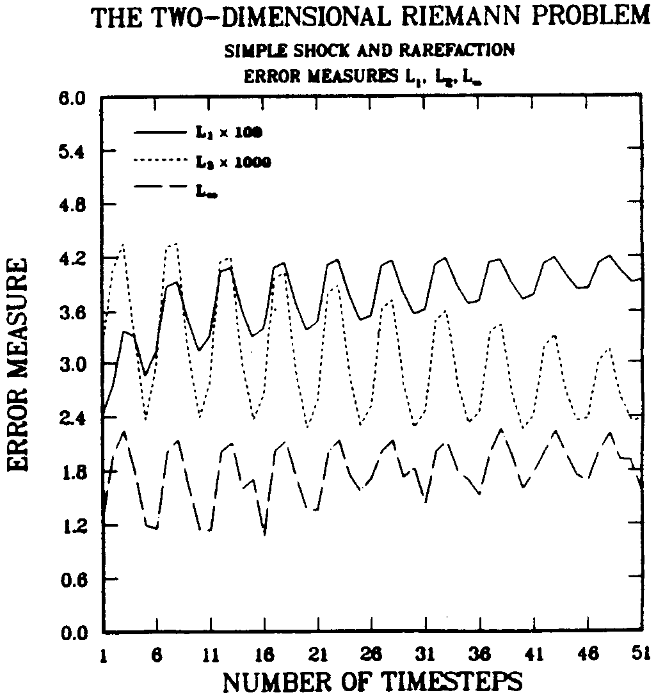

Figure D4

QWIK2D Results for the simple shock and rarefaction problem. Fig. D1 uses P4 on a 100 × 100 grid. Fig. D2 uses P6 on a 100 × 100 grid. Fig. D3 uses P4 on a 50 × 50 grid. Fig. D4 uses P6 on a 50 × 50 grid. Note ordinate scale change on coarser grid.

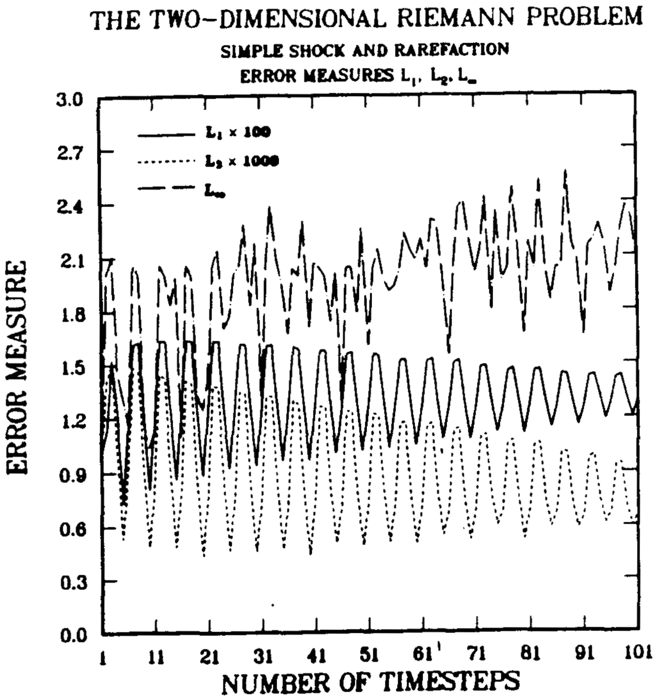

Figure D5

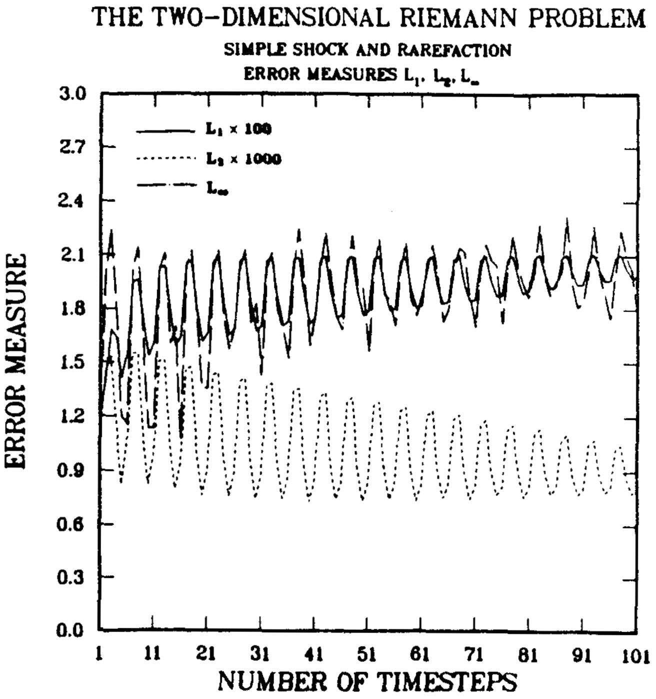

Figure D6

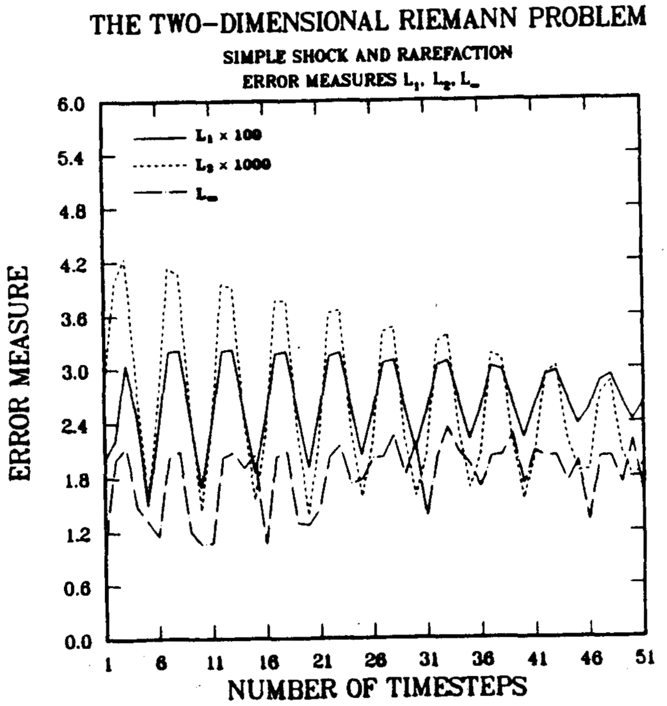

Figure D7

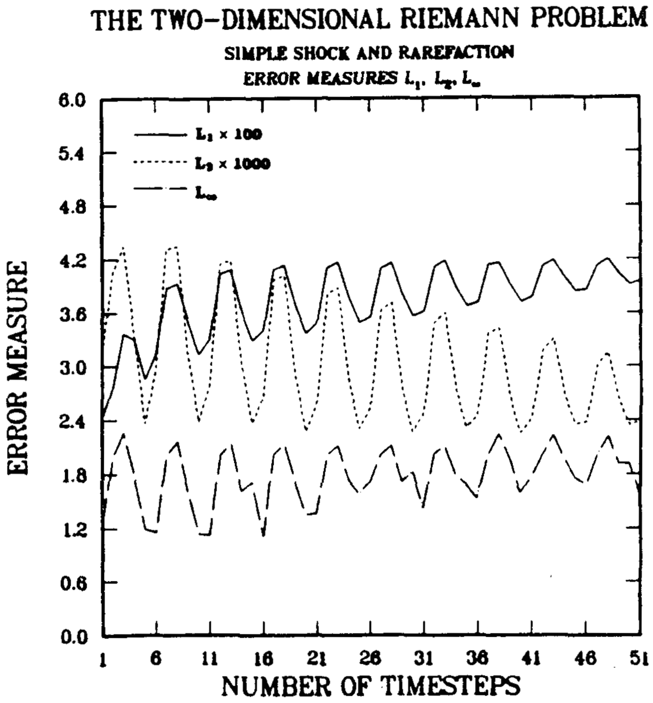

Figure D8

FCT2D/FLIMIT Results for the simple shock and rarefaction problem. Fig. D5 uses P4 on a 100 × 100 grid. Fig. D6 uses P6 on a 100 × 100 grid. Fig. D7 uses P4 on a 50 × 50 grid. Fig. D8 uses P6 on a 50 × 50 grid. Note ordinate scale change on coarser grid.

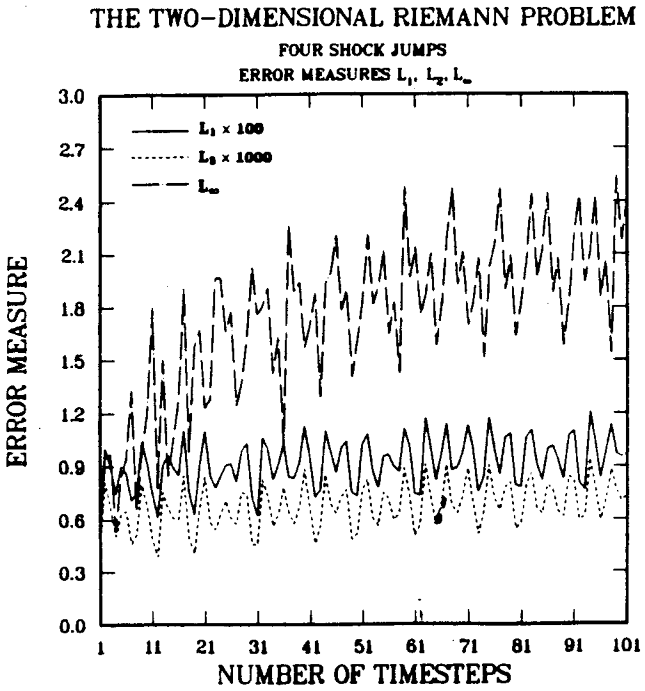

Figure D9

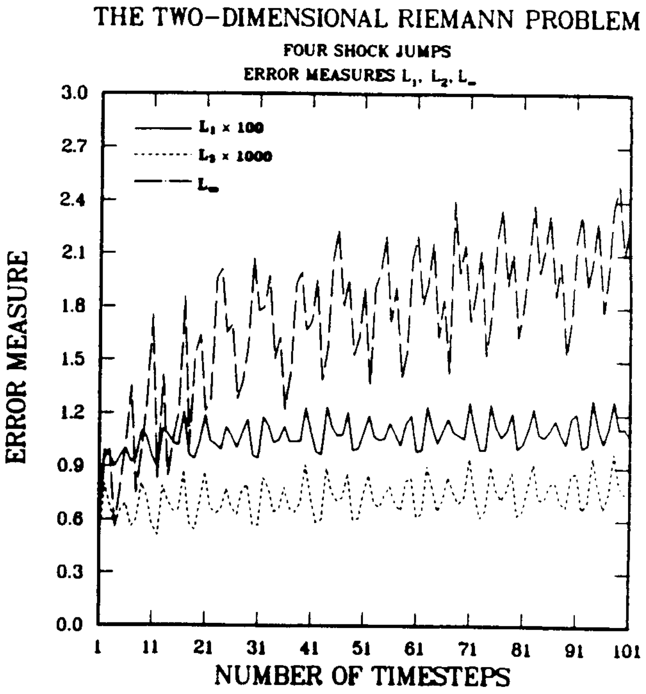

Figure D10

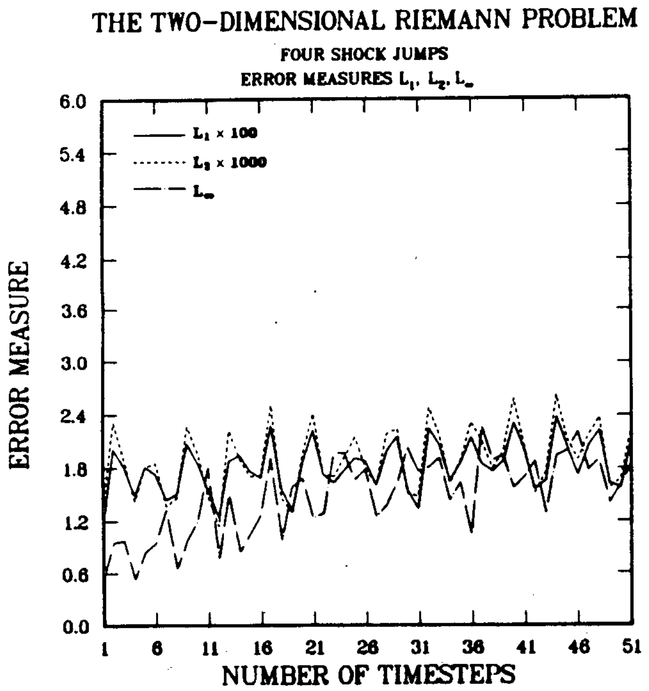

Figure D11

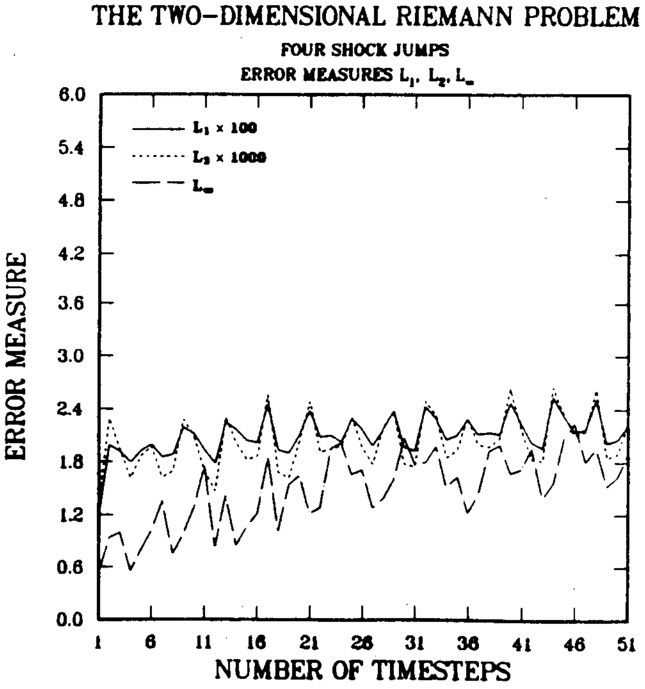

Figure D12

QWIK2D Results for the four simple jump shock problem. Fig. D9 uses P4 on a 100 × 100 grid. Fig. D10 uses P6 on a 100 × 100 grid. Fig. D11 uses P4 on a 50 × 50 grid. Fig. D12 uses P6 on a 50 × 50 grid. Note ordinate scale change on coarser grid.

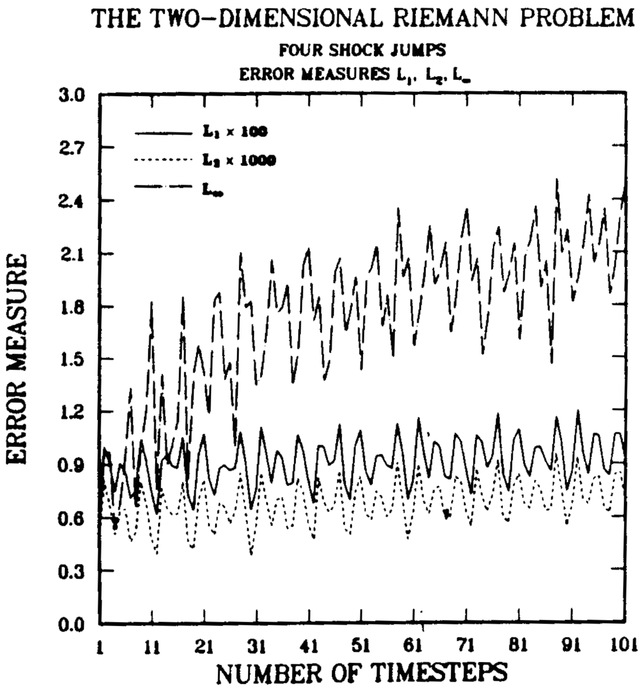

Figure D13

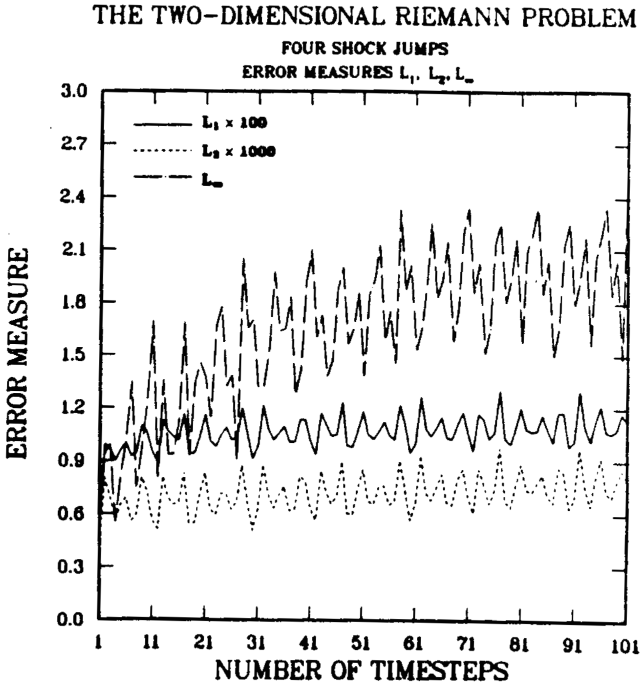

Figure D14

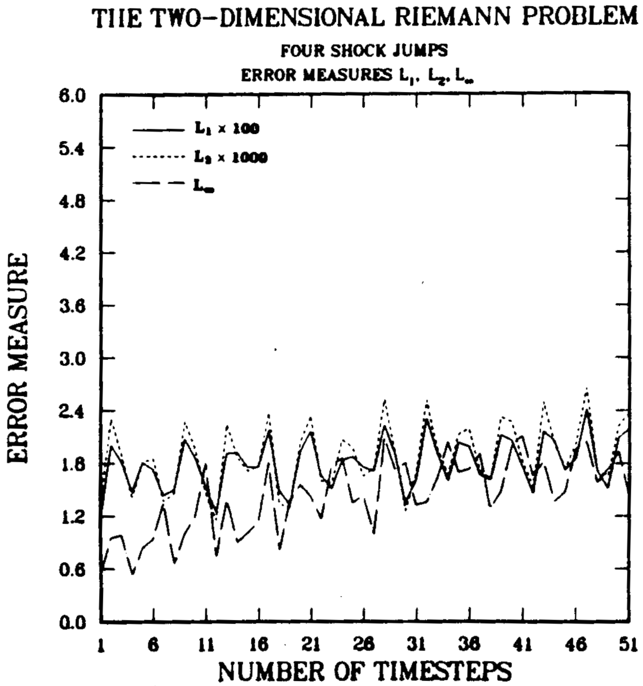

Figure D15

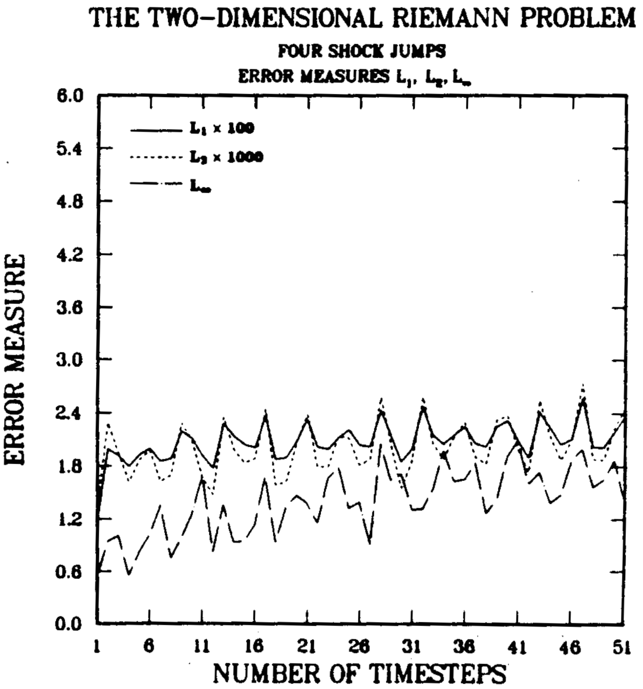

Figure D16

FCT2D/FLIMIT Results for the four simple jump shock problem. Fig. D13 uses P4 on a 100 × 100 grid. Fig. D14 uses P6 on a 100 × 100 grid. Fig. D15 uses P4 on a 50 × 50 grid. Fig. D16 uses P6 on a 50 × 50 grid. Note ordinate scale change on coarser grid.

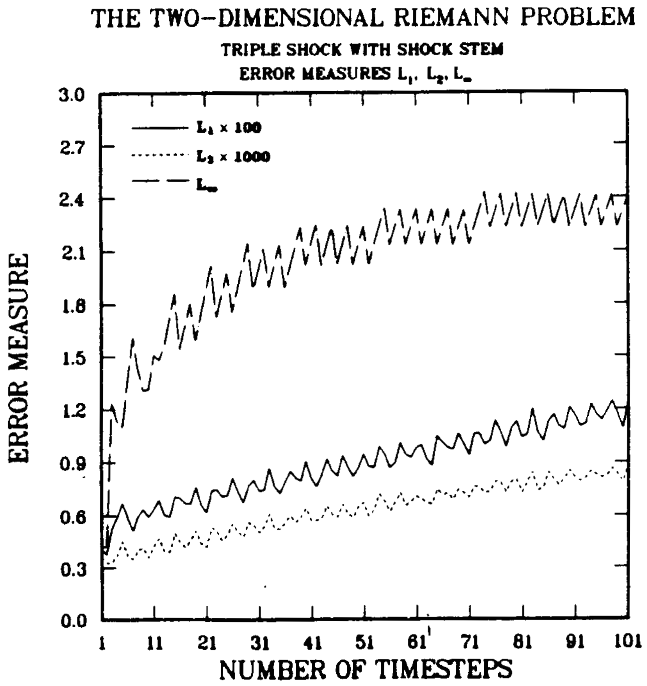

Figure D17

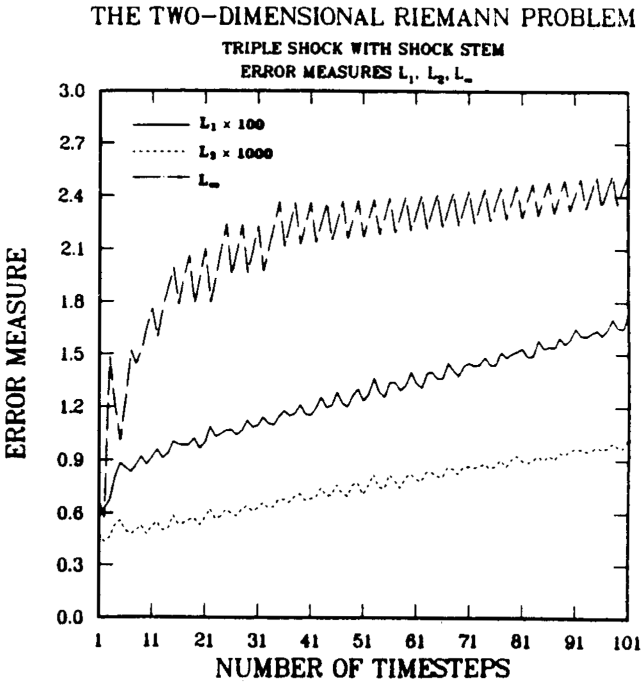

Figure D18

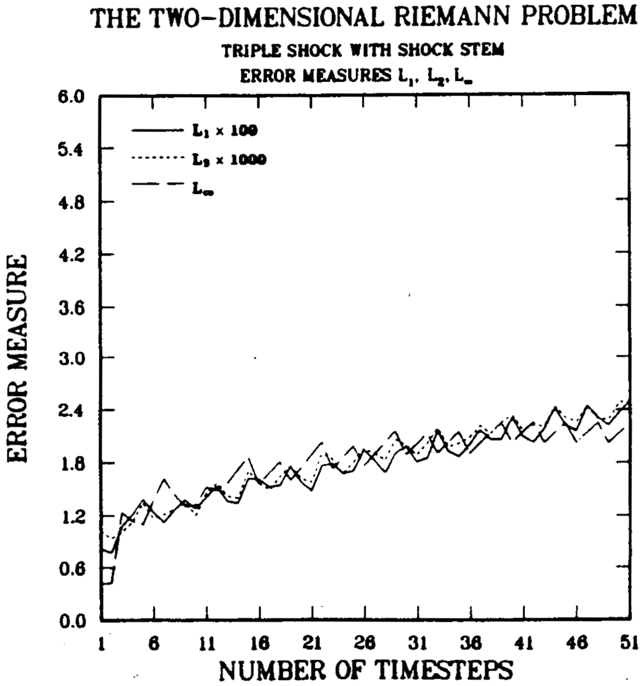

Figure D19

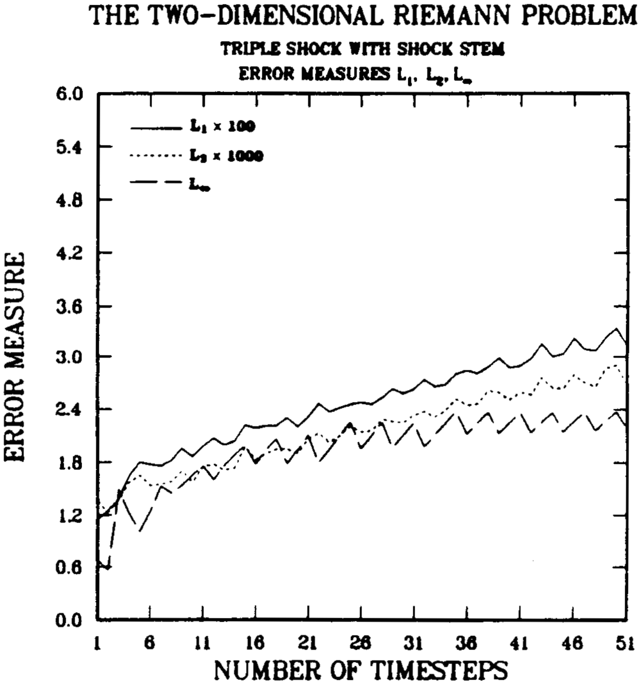

Figure D20

QWIK2D Results for the triple shock with shock stem problem. Fig. D17 uses P4 on a 100 × 100 grid. Fig. D18 uses P6 on a 100 × 100 grid. Fig. D19 uses P4 on a 50 × 50 grid. Fig. D20 uses P6 on a 50 × 50 grid. Note ordinate scale change on coarser grid.

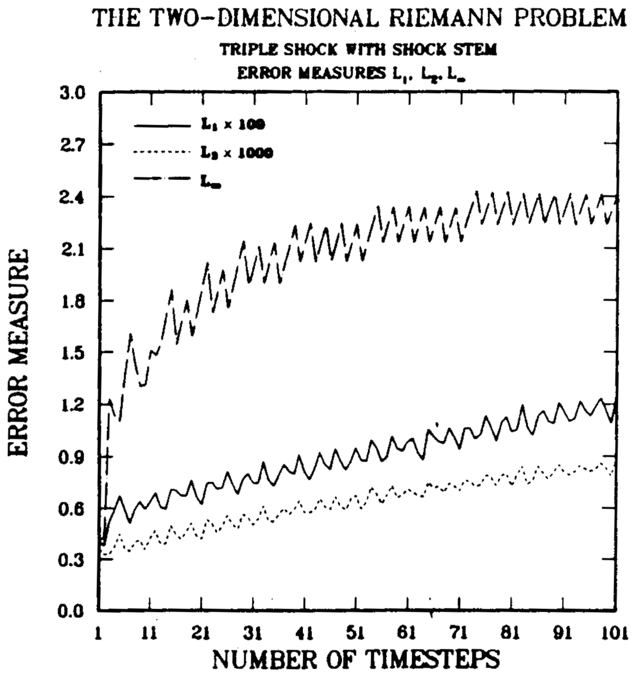

Figure D21

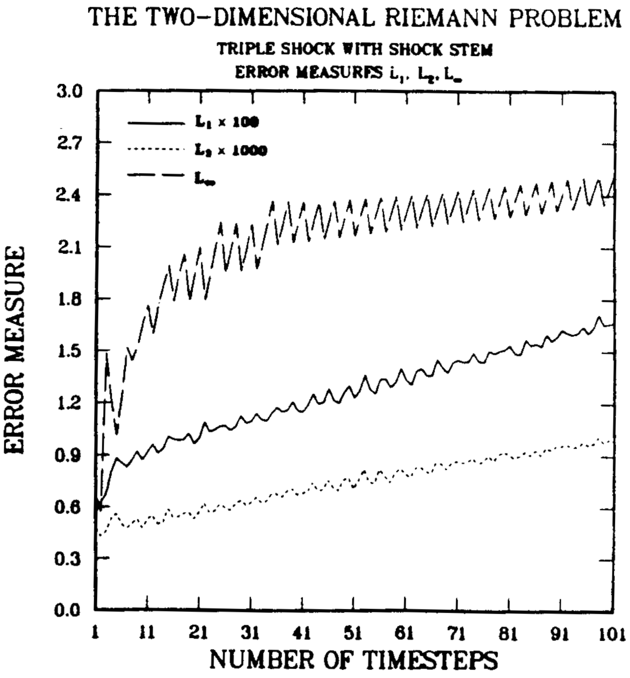

Figure D22

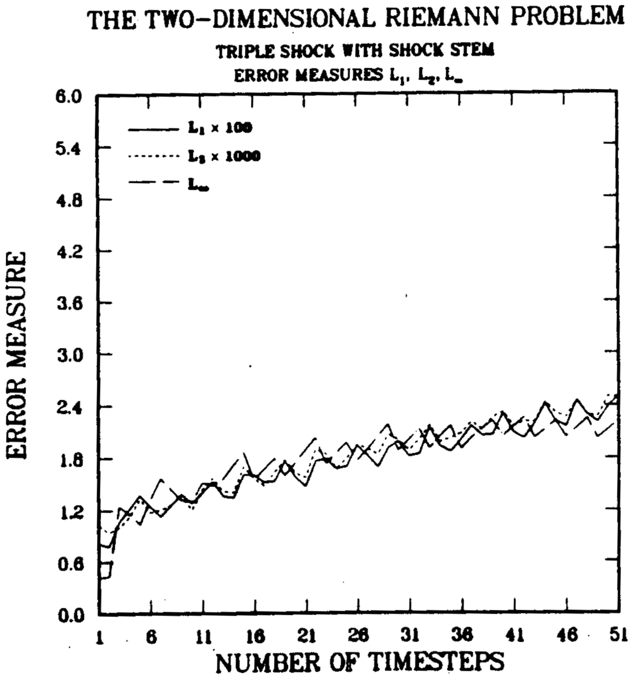

Figure D23

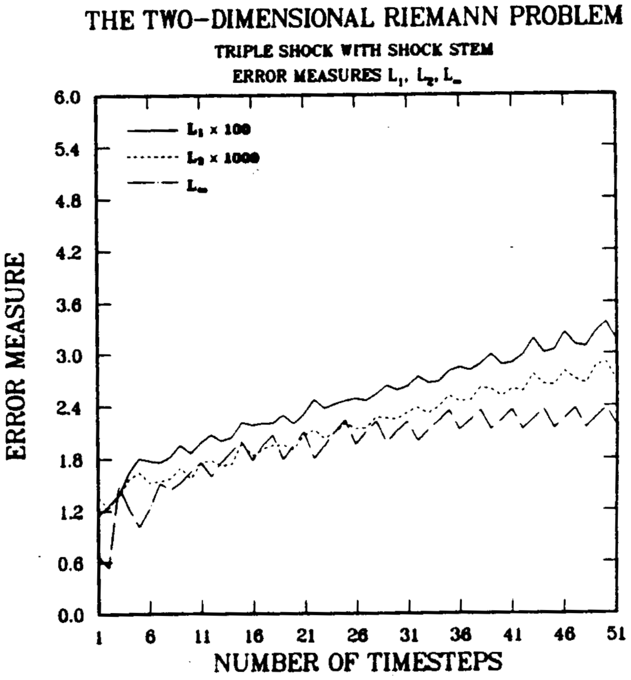

Figure D24

FCT2D/FLIMIT Results for the triple shock with shock stem problem. Fig. D21 uses P4 on a 100 × 100 grid. Fig. D22 uses P6 on a 100 × 100 grid. Fig. D23 uses P4 on a 50 × 50 grid. Fig. D24 uses P6 on a 50 × 50 grid. Note ordinate scale change on coarser grid.

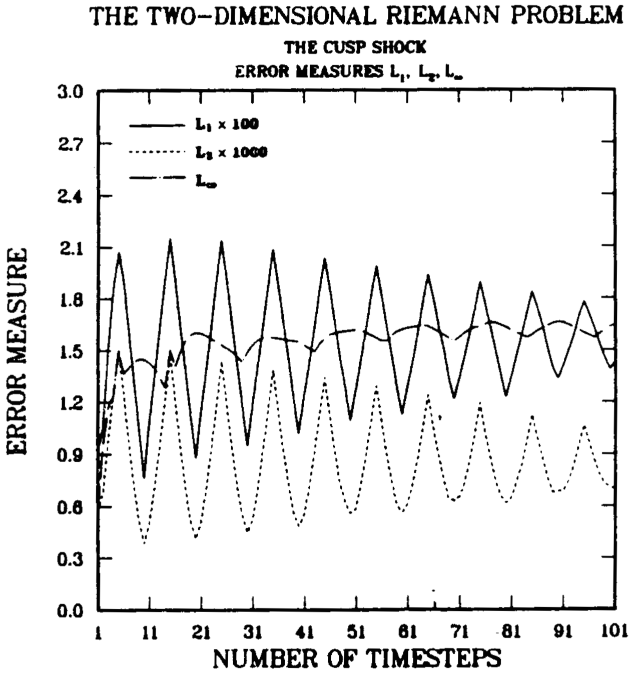

Figure D25

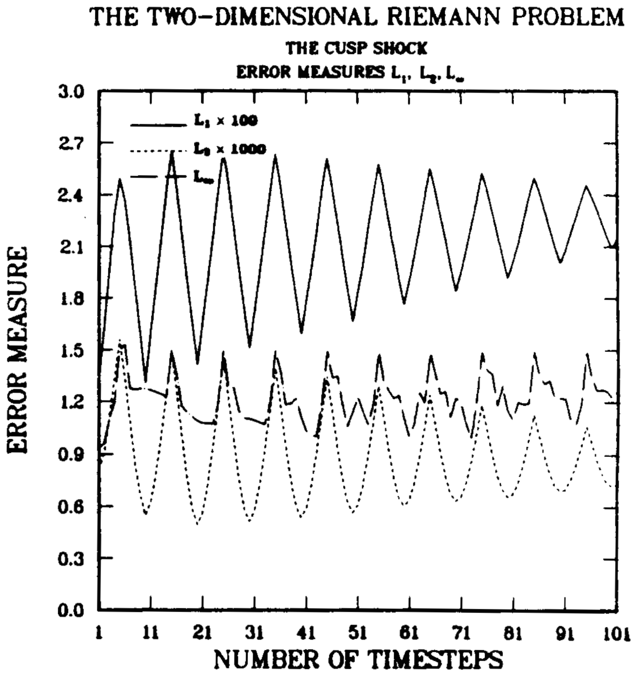

Figure D26

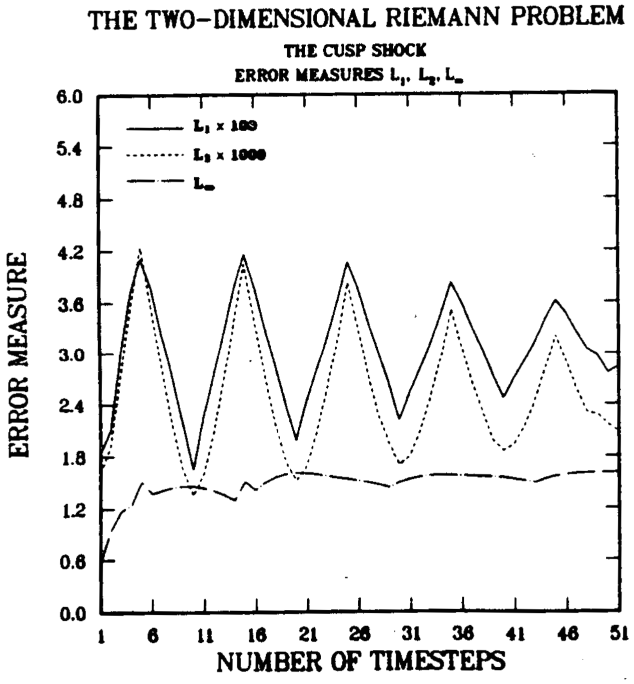

Figure D27

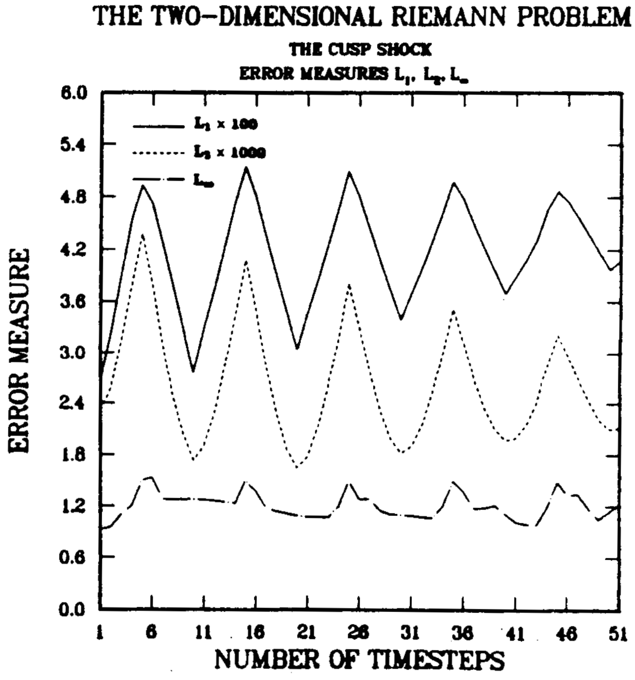

Figure D28

QWIK2D Results for the cusp shock problem. Fig. D25 uses P4 on a 100 × 100 grid. Fig. D26 uses P6 on a 100 × 100 grid. Fig. D27 uses P4 on a 50 × 50 grid. Fig. D28 uses P6 on a 50 × 50 grid. Note ordinate scale change on coarser grid.

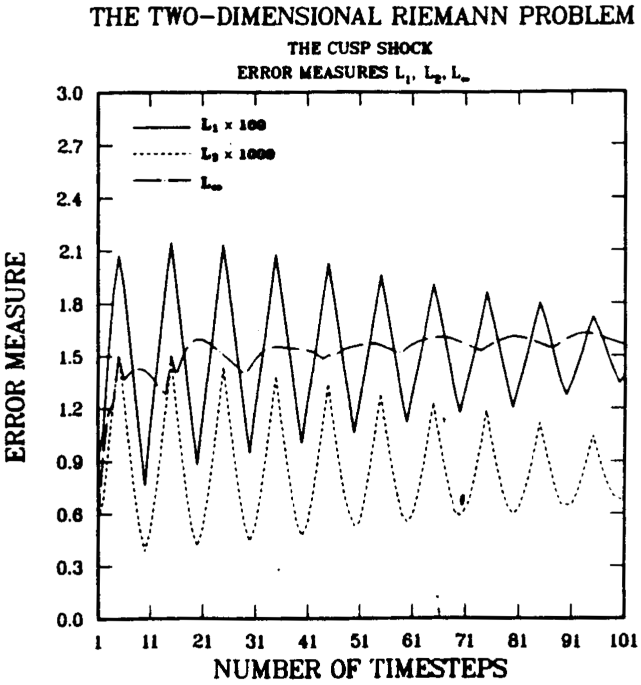

Figure D29

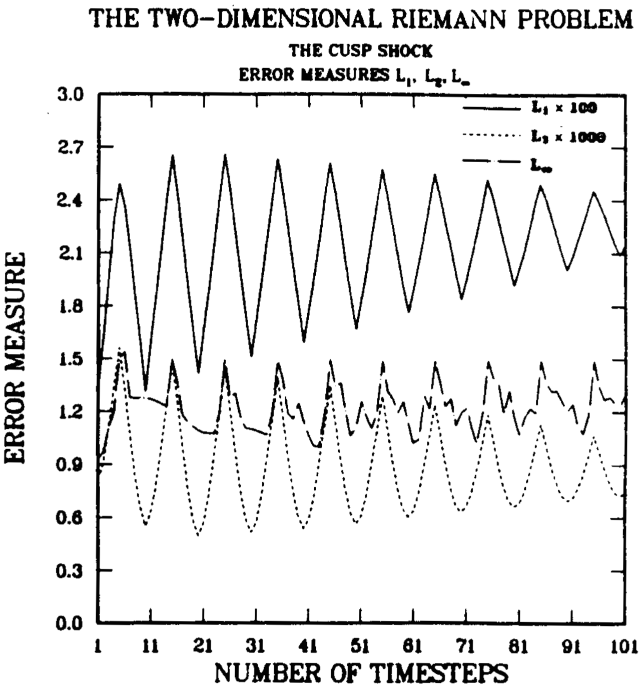

Figure D30

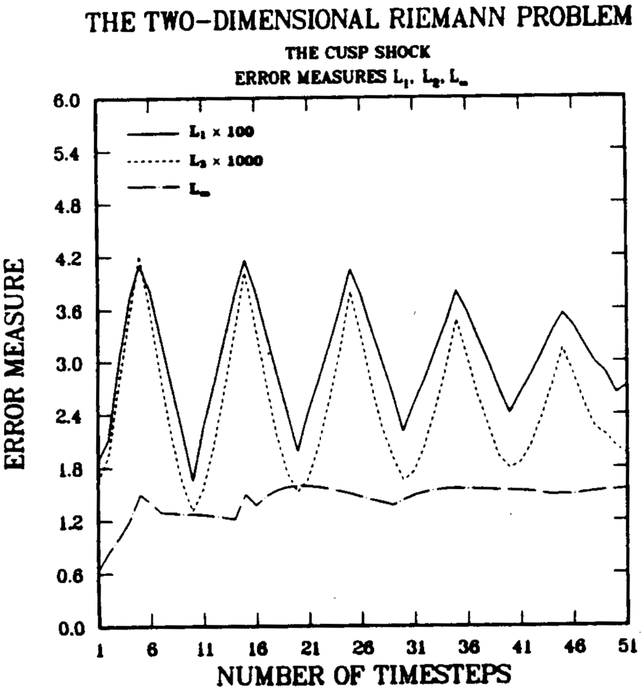

Figure D31

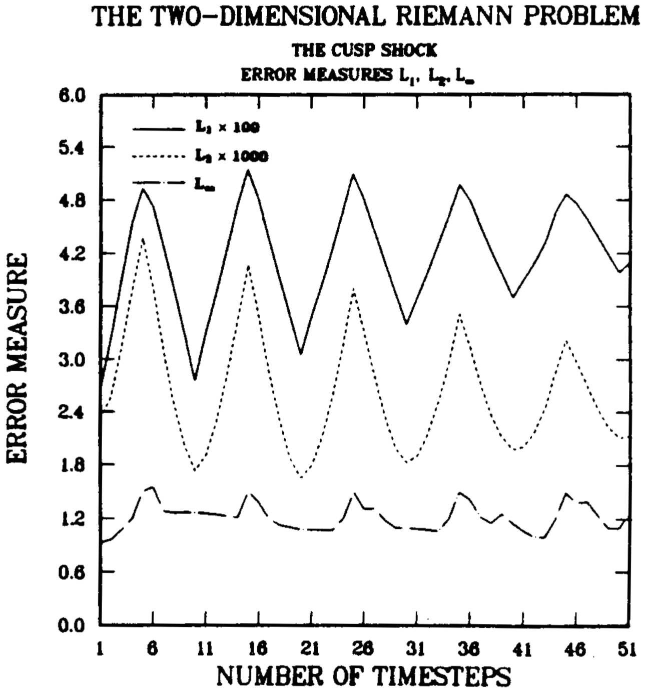

Figure D32

FCT2D/FLIMIT Results for the cusp shock problem. Fig. D29 uses P4 on a 100 × 100 grid. Fig. D30 uses P6 on a 100 × 100 grid. Fig. D31 uses P4 on a 50 × 50 grid. Fig. D32 uses P6 on a 50 × 50 grid. Note ordinate scale change on coarser grid.

Appendix E

Results from Varying the FLIMIT Parameters PRLIM, JPRLIM, and FOLD

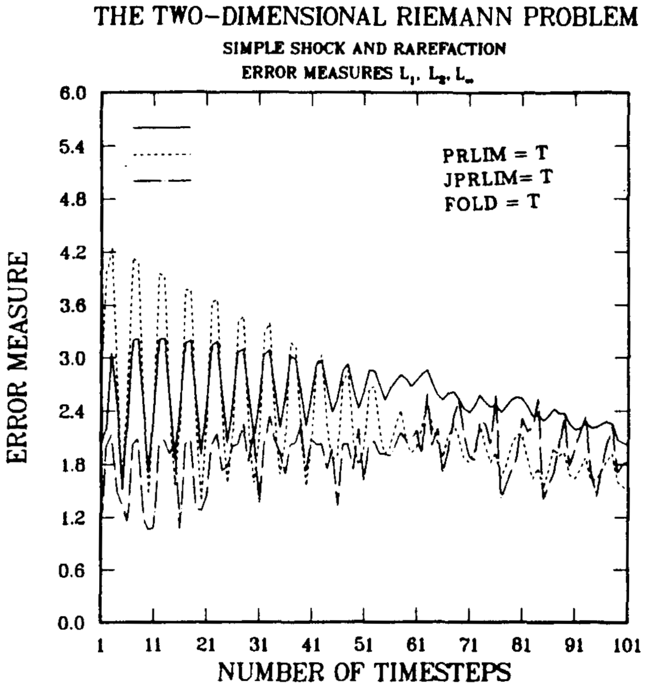

Figure E1

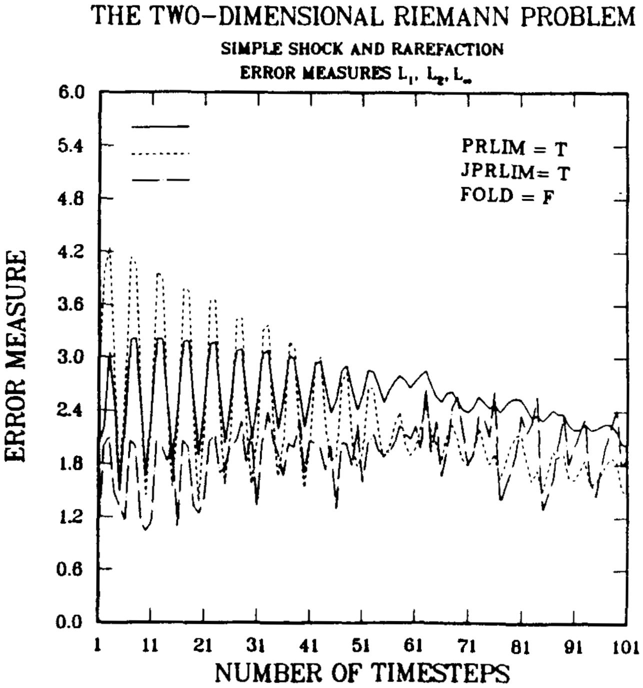

Figure E2

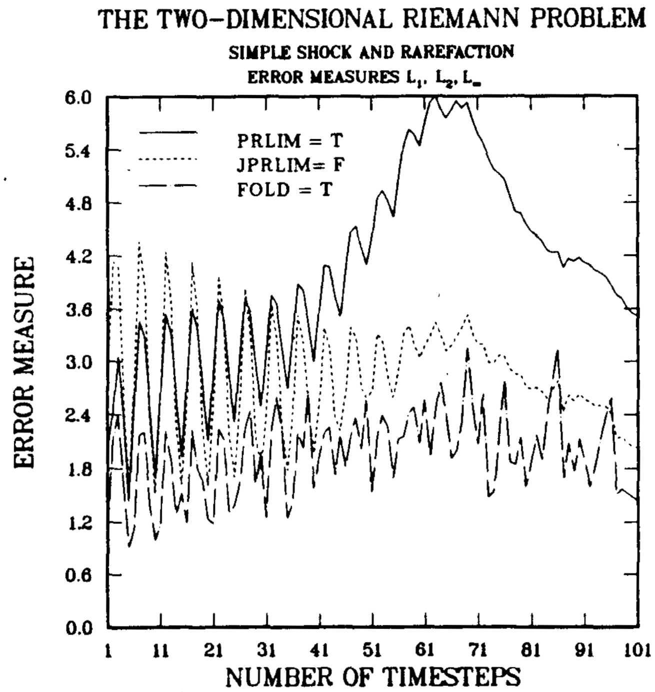

Figure E3

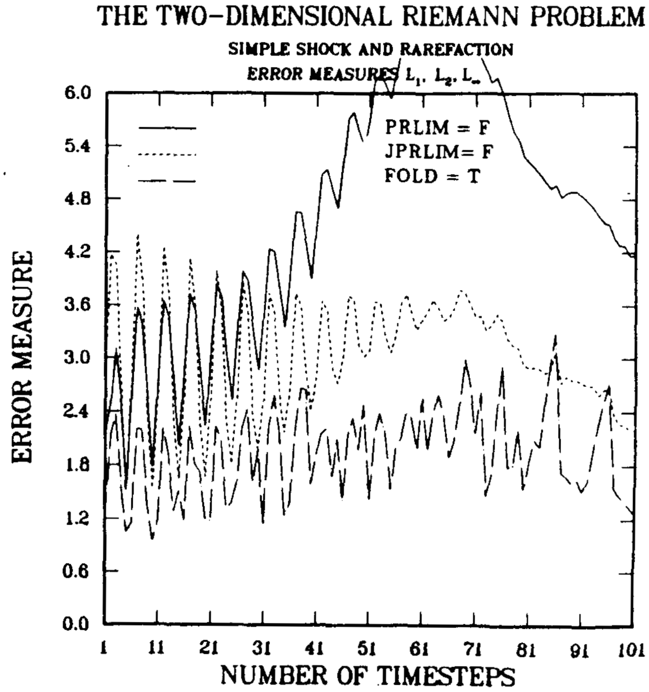

Figure E4

FCT2D/FLIMIT Results (the 2-D Zalesak limiter) from varying the the user-specified variables PRLIM, JPRLIM, and FOLD for all possible combinations. Results are for the simple shock and rarefaction problem using P4 on a 50 × 50 grid.

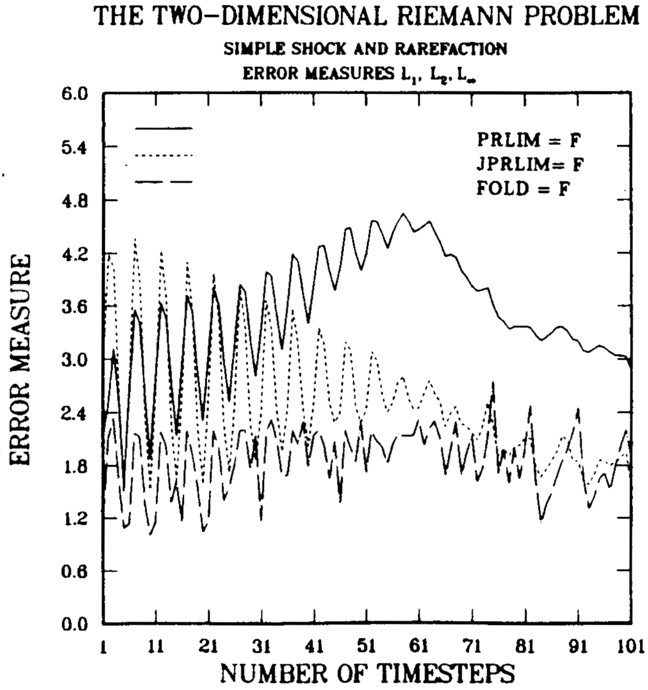

Figure E5

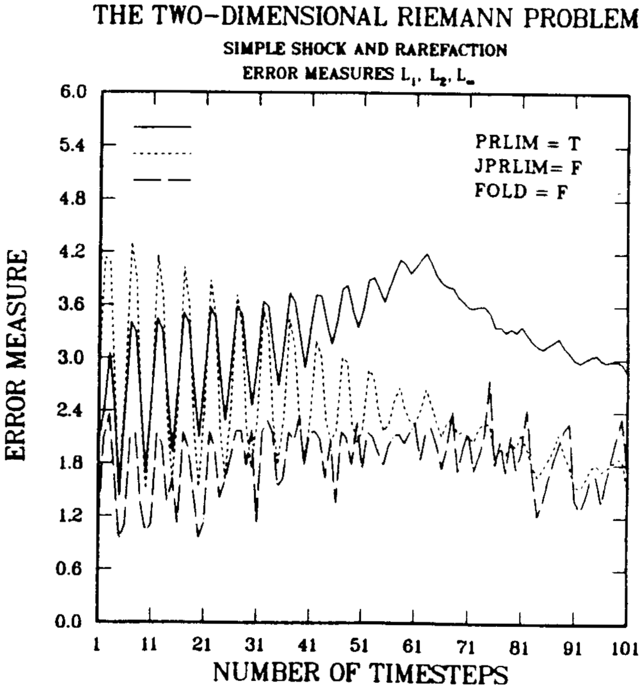

Figure E6

FCT2D/FLIMIT Results (the 2-D Zalesak limiter) from varying the the user-specified variables PRLIM, JPRLIM, and FOLD for all possible combinations. Results are for the simple shock and rarefaction problem using P4 on a 50 × 50 grid.

13 Appendix F – Marker Particle Algorithm

Appendix F

Marker Particle Transport Algorithm

In this Appendix we provide a description of the marker particle transport algorithm that was used in the flow instability studies. Quasiparticle methods, such as PIC [55], MAC [56], and VORTEX [57], are well established computational tools for tracking phenomena, particularly when numerical diffusion effects are to avoided. Similar to these methods, we consider a finite number of marker particles which are Lagrangian advected by the local flow velocities without any interparticle collisional effect. Much of the following algorithm was taken from references [58] and [59].

First, we define a set of marker particles that can be temporally varied and for the kth member of this set we assign a current spatial location, $\mathbf{r}_k^n = (x_k^n, y_k^n)$, and a local velocity, $\mathbf{v}_k^n = (u_k^n, v_k^n)$. The superscript n signifies a current time for all temporally varied quantities. The set of marker particles is overlaid on a computational fixed grid of uniform spacing, $(\Delta x, \Delta y)$, and physical boundary locations enclosing the domain are defined as x_r, x_l, y_r, y_l.

For all marker particles we store previous time values of the marker velocity and time advance the location of particles using an Adams-Bashforth method:

$$\mathbf{r}_k^{n+1} = \mathbf{r}_k^n + \Delta t(\frac{3}{2}\mathbf{v}_k^n - \frac{1}{2}\mathbf{v}_k^{n-1}).$$

Additionally, we make boundary modifications. For example, in the x direction, periodic constraints can be enforced by a correction $x_k^{n+1} = x_r - x_l + x_k^{n+1}$ when $x_k^{n-1} - x_l < 0$ and when $x_r - x_k^{n+1} < 0$ then $x_k^{n+1} = x_l - x_r + x_k^{n-1}$. These modifications imply that a particle crossing a physical boundary reappears at the opposite boundary. Similar corrections are used for the other direction and these formulaes can be modified to treat symmetric or reflective boundary constraints.

With a new location for each particle, we then determine the fixed computational cell location containing the particle and the lower left grid location of this cell is given by $i_k = x_k^{n+1}/\Delta x + 1$ and $j_k = y_k^{n+1}/\Delta y + 1$. With this information, we then determine bilinear area ratios that are used for interpolation. In reference to Figure [F.1], A_1, A_2, A_3 and A_4 are given as:

$$A_1 = (y_{j_k+1}^g - y_k^{n+1})(x_{i_k+1}^g - x_k^{n-1})/\Delta x\Delta y$$

$$A_2 = (y_{j_k+1}^g - y_k^{n-1})(x_k^{n+1} - x_{i_k}^g)/\Delta x\Delta y$$

$$A_3 = (y_k^{n-1} - y_{j_k}^g)(x_k^{n+1} - x_{i_k}^g)/\Delta x\Delta y$$

$$A_4 = (y_k^{n+1} - y_{j_k}^g)(x_{i_k+1}^g - x_k^{n-1})/\Delta x\Delta y$$

Where the g supercript refers to fixed grid quantities. With these relationships, marker particle velocities are determined using the fixed grid hydrodynamic calculations as interpolated to the marker particle according to:

$$\mathbf{v}_k^{n+1} = A_1 \mathbf{v}_{i_k,j_k} + A_2 \mathbf{v}_{i_k-1,j_k} + A_3 \mathbf{v}_{i_k+1,j_k+1} + A_4 \mathbf{v}_{i_k,j_k+1} .$$

When a marker particle carries with it assigned quantities and become active in the hydrodynamic calculations, the transported quanitity is interpolated onto the fixed grid coordinate system using bilinear interpolation:

$$f_{i_k,j_k} = A_1 f_k \; ; \quad f_{i_k+1,j_k} = A_2 f_k \; ;$$

$$f_{i_k+1,j_k+1} = A_3 f_k \; ; \quad f_{i_k,j_k+1} = A_4 f_k$$

All of the above computations are revised during each time step of the hydrodynamics calculations.

Distribution:

Dr. Patrick J. Roache
4925 Kathryn Circle SE
Albuquerque, NM 87108

Naval Research Laboratory (2)
Washington, D.C. 20375
J. P. Boris
D. L. Book

Los Alamos National Laboratory (2)
P.O. Box 1663
Los Alamos, NM 87545
F. Harlow
J. Travis

H. Krier
Department of Mechanical Engineering
University of Illinois
Champaign, IL 61820

P. B. Butler
Department of Mechanical Engineering
University of Iowa
Iowa City, Iowa 52242

Lawrence Livermore National Laboratory (3)
P.O. Box 808
Livermore, CA 94550
E. Lee, L-368
C. M. Tarver, L-324
P. Woodward

P. Colella
Lawrence Berkeley Laboratory
University of California
Livermore, CA 94720

K. Kim
Naval Surface Weapons Center
White Oak Laboratory
Silver Springs, MD 20910

D. Kooker
Army Ballistics Research Laboratory
DRDAR-BLI
Aberdeen Proving Ground, MD 21005

P. Dawson
Department of Mechanical Engineering
Cornell University
Ithaca, NY 14853

T. L. Boggs
Aerothermochemistry Division, 388
Naval Weapons Center
China Lake, CA

W. R. Espander
RDA Associates
Albuquerque, NM

1131 P. A. Taylor
1500 W. Herrmann
1510 J. W. Nunziato
1511 G. G. Weigand
1511 R. E. Benner
1511 R. Givler (3)
1511 G. R. Montry
1512 J. C. Cummings
1512 A. J. Russo
1512 J. E. Shepherd
1512 J. R. Torczynski
1513 D. W. Larson
1513 M. R. Baer (25)
1513 R. J. Gross (30)
1513 C. E. Hickox
1513 S. N. Kempka
1513 J. A. Schutt
1513 Dayfile
1520 D. J. McCloskey
1530 L. W. Davison
1531 S. L. Thompson
1533 P. Yarrington
1533 M. E. Kipp
1534 J. R. Asay

1540 W. C. Luth
1631 W. L. Oberkampf
1632 W. D. Sundberg
1633 S. McAlees
1633 V. Behr
1636 J. K. Cole
1636 F. G. Blottner
2513 J. E. Kennedy
2513 S. F. Roller
2646 A. H. Treadway
3141 D. K. Brown (5)
3151 W. L. Garner (3)
6332 R. Beraun
6425 A. W. Reed
8201 R. J. Gallagher
8231 W. D. Wilson, Actg.
8240 C. W. Robinson
8244 C. M. Hartwig
8244 C. E. Hackett
8245 R. J. Kee
8245 G. H. Evans
8245 W. L. Hermina
8245 S. Paolucci
8362 M. Dyer
8363 B. R. Sanders

FOR DOE/TIC
3154-4 J. Hernandez (28)