UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

A Shock Tracking Technique Based on Conservation in One Space Dimension

De-kang Mao

September 1992

CAM Report 92-42

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90024-1555

A Shock Tracking Technique Based on Conservation in One Space Dimension

Mao De-kang [1]
Department of Mathematics
University of California, Los Angeles
CA 90024, U.S.A.
and
Department of Mathematics
Shanghai University of Science and Technology
Shanghai, China

Abstract

In this paper, which is a continuous work of [20], [21], [22] and [24], we present a shock tracking technique in one space dimension. The main feature of the technique is that it uses the conservativity of the hyperbolic conservation laws rather than Hugoniot condition to track discontinuities. Roughly speaking, the technique is as follows: The computation of a numerical solution on each side of a discontinuity uses information only from the same side. This is able to be done by employing extrapolated data on the same side. From the viewpoint of shock capturing the overall scheme is not conserved; therefore, conservation errors that indicate how much the numerical solution is away from being conserved are formed on every time level. These conservation errors are used to locate the discontinuity positions within grid cells. Numerical analysis of the conservation and of the relation between the conservation errors and discontinuity positions are presented. Handling of interactions of discontinuities is developed. Finally, numerical examples are presented to show the efficiency of the technique.

Key words. shocking tracking, conservation errors, stacking technique, clean-up step.

AMS(MOS) subject classification. 65M06, 65M50, 76L06, 76M20.

[1]Research was supported by ONR Grant No. N00014-86-k-0691, NSF DMS91-03104 and ETH-Zürich.

1. Introduction

We consider the initial value problems of hyperbolic conservation laws, which can be described as

$$u_t + f(u)_x = 0 \tag{1.1a}$$

$$u(x,0) = u_0(x)\ , \tag{1.1b}$$

where $u = (u_1, \cdots, u_m)$ is a state vector and f, the flux, is a vector valued function of m components. For the simplicity of discussion $u_0(x)$ is assumed to be of compact support. The system is hyperbolic in the sense that the $m \times m$ Jacobian matrix

$$A(u) = \frac{\partial f}{\partial u} \tag{1.2}$$

has m real eigenvalues

$$a_1(u) \le a_2(u) \le \cdots \le a_m(u) \tag{1.3}$$

and a complete set of m linear independent right-eigenvectors. A weak solution to (1.1) is a bounded measurable function $u(x,t)$ satisfying

$$\int_0^\infty \int_{-\infty}^\infty (u\phi_t + f(u)\phi_x)dxdt + \int_{-\infty}^\infty u_0(x)dx = 0 \tag{1.4}$$

for all $\phi \in C_0^1((-\infty,\infty) \times [0,\infty))$.

The main difficulty for numerical simulation of (1.1) is that solutions to (1.1) may develop discontinuities, no matter how smooth the initial data are. The loss of smoothness of solutions due to the occurrence of discontinuities causes consistency problem for the numerical schemes with the original partial differential equations.

There are two kinds of approaches of difference approximations for (1.1), namely shock capturing and shock tracking. The shock capturing methods ignore the presence of the discontinuities by applying almost the same numerical schemes everywhere in the flow. It is expected that the discontinuities are resolved by sharp profiles of the numerical solutions. The methods are simple and easy to code and apply. Also it has been shown by numerical experiments and been proved theoretically for some particular cases that if the numerical schemes are conservative, stable and satisfy entropy condition the numerical

solutions will always have correct pictures. Here the "stable" means that the total variations of the numerical solutions are uniformly restricted by some bounds, and the "correct pictures" means that the numerical solutions do not present non-physical discontinuities and all the discontinuities move with correct speeds.

In the last several decades, a lot of efficient shock capturing difference schemes have been constructed, studied and found to be very useful in shock calculation (see [3], [12], [13], [14], [15], [19], [27], [28], [30], [31], [33], [34], [35], [36], and the references cited there). The author should particularly mention ENO schemes developed by Harten, Engquist, Osher, and Chakravarthy and PPM schemes developed by Colella and Woodward. These schemes essentially eliminate spurious oscillations near discontinuities and present very good numerical results.

However, the main drawback of the methods is that they are not able to give exact positions of discontinuities; besides, smear of discontinuities seems to be unavoidable.

Instead of ignoring the presence of discontinuities, the shock tracking methods use lower adaptive grids, the so-called fronts or interfaces, to fit the discontinuities in the numerical solutions. The partial differential equations (1.1a) are solved separately in each region surrounded by the fronts using a method designed for smooth solutions, while the fronts are moved using Rankine-Hugoniot jump conditions.

Early proposals for shock tracking can be traced back to Richtmyer and Morton [29]. Several of its realizations in one space dimension can be found in [16], [18], [26], [32] and [37]. A more challenging task is its realization in two space dimensions due to geometric and dynamic complications. Glimm and his coworkers, e.g., [1], [5], [6], [7], [8], [9], [10], and [11], have developed a very extensive set of tools for front tracking, which have been successfully applied to a wide variety of problems. This package includes procedures to deal with complicated interactions of fronts, Mach triple points, and other such structures.

The shock tracking methods have been proved successful in dealing with essentially piecewise smooth solutions by its very nice numerical results. They present both the numerical solutions and discontinuity positions with high accuracy. However, it seems still to be a problem for these methods to deal with solutions that are not quite piecewise smooth, e.g., solutions

with several spontaneous shocks in a small region. Besides, the methods is complicated in both coding and applications.

Since eight years ago the author has been developing a shock tracking technique that uses the conservativity of (1.1a) rather than Rankine-Hugoniot jump condition to locate the discontinuity positions (see [20] -[24]). In other words, he is trying to use shock capturing's idea to do the shock tracking. The present paper is a continuous work on this technique. Roughly speaking, the technique presented in this paper is as follows. The computation of numerical solutions on each side of discontinuities uses information only from the same side. This is able to be done by employing extrapolated data at the grid points on the other side of the discontinuities. From the shock capturing viewpoint the overall scheme is not conservative; therefore, conservation errors that indicate how much the numerical solutions are away from being conserved are formed on every time level. These conservation errors are used to locate the discontinuity positions within grid cells.

For two reasons the author believes that the tracking based on conservation is better than the tracking based on Hugoniot condition. First, the conservation, unlike Hugoniot condition, is a global feature of (1.1a). Therefore, it is easer to carry out than Hugoniot condition in numerical simulation because to maintain the conservation one does not need to know the detailed structure of the numerical solutions. For example, when handling collisions of discontinuities one does not need to know when and where the collisions actually happen (see section 4). Second, the conservation is more essential than Hugoniot condition in the sense that Hugoniot condition is a description of conservation only for piecewise smooth solutions. Solutions to (1.1) may have very complicated structure and their piecewise smoothness may become questionable. In this case Hugoniot condition is not suitable for describing the solutions. However, the solution are always conserved no matter how complicated they are. Therefore, the tracking based on the conservation is more robust and able to deal with solutions with complicated structures (see the second numerical example in section 6).

The technique presented in this paper, just as expected, is very simple and efficient. It has the following advantages:

1) The technique can be applied to any shock capturing schemes and it works just as an adjustment on the schemes near discontinuities. Therefore, the algorithm can be easily code in an almost shock capturing fashion.

2) The computation proceeds on the regular grid, and no adaptive grid is needed, by which we get rid of completely the small cell problem that troubles most shock tracking methods (see, e.g., [1], [3], and [18]).

3) The handling of discontinuity collisions is quite simple. Due to a so-called "stacking technique" developed in section 4 the handling is also very accurate.

4) The overall scheme is conservative; therefore, it is robust in dealing with small scale structures and spontaneous shocks.

Development of a shock tracking technique based on the conservation in two space dimensions, which is an extension of the tracking technique in one space dimension, is underway ([23] and [25]).

The author should particularly mention Harten's subcell resolution technique [12], which also uses the conservativity of (1.1a) to locate the discontinuity positions within grid cells. Actually, the ideas behind the two techniques are just the same. However, Harten's subcell resolution works in a shock capturing way, while the author's technique works in a shock tracking way. Besides, Harten's subcell resolution is applied only to the second field to improve the computation of contact discontinuities, while the author's technique has been extended to all kinds of discontinuities and their interactions.

The author would also like to mention Sjogreen and Engquist's work [4], from which he understood that the "artificial terms along t direction" in his former paper [20] are actually errors of conservation. So he abandoned the old terminology and adopted the terminology "conservation error" for these quantities, which is more precise.

Recently Colella and I.L. Chern [2] and LeVeque [18] have also developed some conservative shock tracking methods. It seems that maintaining conservation benefits also these tracking methods since their numerical results are quite good and their algorithm are simpler than shock tracking methods that are not conservative. However, they do not use the conservation to locate discontinuity positions. The discontinuity fronts are moved essentially still by Hugoniot condition; therefore, the adaptive grid is employed and particular handling for interaction points of discontinuities is needed.

The format and contributions to the development of the shock tracking technique of this paper are as follows.

We first develop the shock tracking technique for the scalar case of (1.1) in §2 through §4. In §2 we describe how the technique uses extrapolated data to compute a numerical solution on the two sides of a single discontinuity and introduce the conservation errors.

In §3 we investigate the relation between the conservation errors and the discontinuity positions. Under the assumption that the underlying schemes are Godunov type we investigate the relation for cases of any order extrapolation through an approach different from and simpler than that in [22]. The results are generalized and improved than in [22].

In §4 we describe how the technique deals with interactions of discontinuities. The so-called "stacking technique" is developed in this section. We also study the conservation errors in the stacking case and show the conservation of the numerical solution together the conservation errors.

In §5 we apply the technique to the system case of (1.1). A key point of this application is to let information associated with other characteristic fields be able to travel through discontinuities. The application in [22] realize this point by solving Riemann problems related to the original and extrapolated data of the numerical solution. In this paper we develop a so-called "clean-up" step to fulfill this task. As a result, the application is simpler than that in [22].

In §6 we present two numerical examples; particularly, the second one demonstrates the robustness of the technique in dealing with solutions with complicated structures and spontaneous shocks.

§7 is the conclusion.

2. Conservation Errors

In this and the following two sections we assume that both u and f in (1.1) are scalar and f is convex. We now describe the technique for a single discontinuity and introduce the conservation errors.

We assume that the underlying scheme is of Godunov type; i.e.,

$$u_j^{n+1} = u_j^n - \lambda(\hat{f}_{j+1/2}^n - \hat{f}_{j-1/2}^n), \tag{2.1}$$

where the numerical solution u_j^n is an approximation of *cell-average* of the solution and the numerical flux $\hat{f}_{j+1/2}^n$ is an approximation of the flux average

on the interface of adjacent cells. More precisely,

$$u_j^n \simeq \frac{1}{h}\int_{x_{j-1/2}}^{x_{j+1/2}} u(x,t_n)dx \tag{2.2}$$

and

$$\hat{f}_{j+1/2}^n \simeq \frac{1}{\tau}\int_{t_n}^{t_{n+1}} f(u(x_{j+1/2},t))dt. \tag{2.3}$$

We say that the scheme is of $(2k+1)$-point if the numerical flux is a function of $2k$ variables

$$\hat{f}_{j+1/2}^n = \hat{f}(u_{j-k+1}^n,\cdots,u_{j+k}^n) \tag{2.4}$$

and consistent with the flux in (1.1a) in the sense that $\hat{f}(u,\cdots,u) = f(u)$.

We assume that there is a discontinuity in the numerical solution and on the nth time level its position ξ^n is in the cell $[x_{j_1}, x_{j_1+1}]$. This cell, which contains the discontinuity, is called a *critical cell.* Suppose that we have known the discontinuity position ξ^{n+1} on the $(n+1)$th time level at this moment, whose calculation will be described in the following section. There will be three different cases for ξ^{n+1}. First, it is still in the same cell; i.e., $x_{j_1} \le \xi^{n+1} \le x_{j_1+1}$. Second, it moves to the left of the cell; i.e., $\xi^{n+1} \le x_{j_1}$. And third, it moves to the right of the cell; i.e. $x_{j_1+1} \le \xi^{n+1}$ (see Figure 2.1).

One of the main ingredients of the technique is to let the computation of the numerical solution on each side of the discontinuity use information only from the same side. This can be done as follows. First, we extrapolate the numerical solution on the two sides and obtain two sets of extrapolated data, namely $u_{j_1+1}^{n,-},\cdots,u_{j_1+k+1}^{n,-}$ and $u_{j_1-k}^{n,+},\cdots,u_{j_1}^{n,+}$. The data with "$-$" are from the left to the right, while the data with "$+$" are from the right to the left (see Figure 2.2). In the first case, for all $j \le j_1$ the numerical solution is computed as

$$u_j^{n+1} = u_j^n - \lambda(\hat{f}_{j+1/2}^{n,-} - \hat{f}_{j-1/2}^{n,-}), \tag{2.5}$$

while for all $j \ge j_1+1$ it is computed as

$$u_j^{n+1} = u_j^n - \lambda(\hat{f}_{j+1/2}^{n,+} - \hat{f}_{j-1/2}^{n,+}), \tag{2.6}$$

where $\hat{f}_{j+1/2}^{n,-}$ and $\hat{f}_{j+1/2}^{n,+}$ are defined as

$$\hat{f}_{j+1/2}^{n,-} = \hat{f}(u_{j-k+1}^n,\cdots,u_{j_1}^n,u_{j_1+1}^{n,-}\cdots,u_{j+k}^{n,-}) \tag{2.7}$$

and

$$\hat{f}^{n,+}_{j+1/2} = \hat{f}(u^{n,+}_{j-k+1}, \cdots, u^{n,+}_{j_1}, u^{n}_{j_1+1}, \cdots, u^{n}_{j+k}), \tag{2.8}$$

respectively. Meanwhile, the critical cell on the $(n+1)$th time level is still in the same cell $[x_{j_1}, x_{j_1+1}]$. In the second case, for all $j \leq j_1 - 1$ the numerical solution is computed by (2.5), and for all $j \geq j_1 + 1$ it is computed by (2.6). At the grid point $j = j_1$, which is now on the right side of the discontinuity on the $(n+1)$th time level, the numerical solution is computed as

$$u^{n+1}_{j_1} = u^{n,+}_{j_1} - \lambda(\hat{f}^{n,+}_{j_1+1/2} - \hat{f}^{n,+}_{j_1-1/2}). \tag{2.9}$$

Meanwhile, the critical cell on the $(n+1)$th time level is $[x_{j_1-1}, x_{j_1}]$. In the third case, for all $j \leq j_1$ the numerical solution is computed by (2.5), and for all $j \geq j_1 + 2$ it is computed by (2.6). At the grid point $j = j_1 + 1$, which is now on the left side of the discontinuity on the $(n+1)$th time level, the numerical solution is computed as

$$u^{n+1}_{j_1+1} = u^{n,-}_{j_1+1} - \lambda(\hat{f}^{n,-}_{j_1+3/2} - \hat{f}^{n,-}_{j_1+1/2}). \tag{2.10}$$

Meanwhile, the critical cell on the $(n+1)$th time level is $[x_{j_1+1}, x_{j_1+2}]$.

Computed in such a way the numerical solution will have high accuracy up to the discontinuity. However, the overall scheme is not conservative. This is because in the first case in cell $[x_{j_1}, x_{j_1+1}]$, in the second case in cell $[x_{j_1-1}, x_{j_1}]$, and in the third case in cell $[x_{j_1+1}, x_{j_1+2}]$ different numerical fluxes are used in the computation. The computation on the left uses the fluxes with "$-$" in these cells, while the computation on the right uses the fluxes with "$+$". The difference of numerical flux will accumulate and form an error of conservation.

To analyze this conservation error we introduce a new scheme with numerical flux

$$\tilde{f}^{n}_{j+1/2} = \begin{cases} \hat{f}^{n,-}_{j+1/2} & j < j_1 \\ \hat{f}^{n,+}_{j+1/2} & j \geq j_1 \end{cases} \tag{2.11}$$

and write the overall scheme in a conservation-like form by adding some auxiliary terms on the RHS (right hand side) of (2.11). Namely,

$$u^{n+1}_{j} = u^{n}_{j} - \lambda(\tilde{f}^{n}_{j+1/2} - \tilde{f}^{n}_{j-1/2}) + p^{n}_{j+1/2} - p^{n}_{j-1/2} + q^{n+1}_{j} - q^{n}_{j}. \tag{2.12}$$

Obviously, there are infinite choices of these $p^n_{j+1/2}$'s and q^n_j's to realize the overall scheme. However, if we restrict q^n_j and q^{n+1}_j to be non-zero only at the left endpoints of the critical cells on the corresponding time level and $p^n_{j+1/2}$ to be non-zero only in the vicinity of the critical cell and regard q^n_j is known, these $p^n_{j+1/2}$'s and q^{n+1}_j's will be uniquely determined. In the above mentioned three cases their terms that may be non-zero are as follows. In the first case,

$$q^{n+1}_{j_1} = q^n_{j_1} + \lambda(\hat{f}^{n,+}_{j_1+1/2} - \hat{f}^{n,-}_{j_1+1/2}); \tag{2.13}$$

in the second case,

$$\begin{aligned} p^n_{j_1-1/2} &= -q^n_{j_1} + (u^n_{j_1} - u^{n,+}_{j_1}) + \lambda(\hat{f}^{n,-}_{j_1-1/2} - \hat{f}^{n,+}_{j_1-1/2}) \\ q^{n+1}_{j_1-1} &= -p^n_{j_1-1/2}; \end{aligned} \tag{2.14}$$

and in the third case,

$$\begin{aligned} p^n_{j_1+1/2} &= q^n_{j_1} + \lambda(\hat{f}^{n,+}_{j_1+1/2} - \hat{f}^{n,-}_{j_1+1/2}) \\ q^{n+1}_{j_1+1} &= q^n_{j_1} + (u^{n,-}_{j_1+1} - u^n_{j_1+1}) + \lambda(\hat{f}^{n,+}_{j_1+3/2} - \hat{f}^{n,-}_{j_1+3/2}). \end{aligned} \tag{2.15}$$

Although the numerical solution u^n_j is not conserved, according to (2.12) the numerical solution together with q^n_j is conserved; i.e.,

$$\sum_{j=-\infty}^{\infty} (u^{n+1}_j - q^{n+1}_j) = \sum_{j=-\infty}^{\infty} (u^n_j - q^n_j) = \sum_{j=-\infty}^{\infty} (u^0_j - q^0_j). \tag{2.16}$$

As a matter of fact, $q^n_{j_1}$ is just the conservation error formed on the nth time level, which presents how much the numerical solution is away from being conserved.

The original conservation error comes from two sources. First, if the discontinuity starts at the initial time level, then the initial conservation error is able to be determined from the initial data by the formula derived in the following section. Second, if the discontinuity is detected during the computation, then the original conservation error is set to be zero by the consideration of conservation.

From (2.13)-(2.15) we see that when time evolves the conservation error is effected by two factors, namely the accumulation of the flux difference caused by the usage of different numerical fluxes in the same cell and the change of the grid points from one side of the discontinuity to the other side. This

conservation error contains information of the discontinuity position, and in the following section we will discuss how to use it to locate the discontinuity.

3. Locating Discontinuity Position through Conservation Error

In this section we derive the formula by which we can locate the discontinuity positions through the conservation errors. Upon this formula we are able to complete our algorithm.

Since the derivation is based on reconstructions of the solution from its cell-average approximation, a brief description of the reconstruction procedures in Godunov type schemes is needed first.

As well known, a Godunov type scheme employs a reconstruction procedure to recover the solution from its cell-average approximations because the evaluation of the numerical fluxes requires knowledge of the solution itself rather than the cell-averages. In this reconstruction procedure an approximation of the solution is recovered in each cell $[x_{j-1/2}, x_{j+1/2}]$ through the cell-average approximations in this and the nearby cells (see [3], [12], [14], and [34]).

Different Godunov type schemes employ different reconstruction procedures; however, all of them must maintain the conservation in the sense that

$$\frac{1}{h}\int_{x_{j-1/2}}^{x_{j+1/2}} R(x;u^n) = u_j^n, \tag{3.1}$$

where $R(x;u^n)$ is the reconstruction of the solution. We say that a reconstruction is of the rth order if for a smooth solution $u(x,t)$ we have

$$R(x;u^n) = u(x,t) + O(h^r) \tag{3.2}$$

in each cell $[x_{j_1-1/2}, x_{j_1+1/2}]$. The discussion in [12] shows that if the reconstruction is of the rth order, the numerical scheme (2.1) will be the rth order accurate in the sense of cell-average.

Two simple reconstructions are as follows:

1)

$$R(x,u^n) = u_j^n \quad x_{j-1/2} \le x < x_{j+1/2} \tag{3.3}$$

is the first order reconstruction used in Godunov scheme.

2)

$$R(x;u^n) = u_j^n + S_j^n(x - x_j) \quad x_{j-1/2} \le x < x_{j+1/2}, \tag{3.4}$$

where the slope S_j^n is

$$S_j^n = \frac{1}{h}(u_{j+1}^n - u_j^n), \tag{3.5}$$

is a second order reconstruction. However, this reconstruction is never used in any second order Godunov type schemes since it will produce spurious oscillations. The reconstructions used in all second order Godunov schemes have certain kinds of restrictions on the slope to control the oscillation of the numerical solutions.

We now turn to derive the formula that relates the conservation error and the discontinuity position. We still denote the critical cell by $[x_{j_1}, x_{j_1+1}]$, discontinuity position by ξ^n on the nth time level, and the extrapolated data by $u_j^{n,-}$ and $u_j^{n,+}$ on the two sides of the critical cell, respectively. In the following discussion we assume that the exact solution is smooth on the two sides of the discontinuity and the numerical solution is accurate up to the rth order. The later assumption is reasonable since the computation of the numerical solution on the two sides of the discontinuity employs extrapolated data, by which it evades the discontinuity. Therefore, under the first assumption the numerical solution will be very accurate if a good underlying scheme is used.

First we improve the reconstruction of the numerical solution by taking into account the discontinuity in the critical cell. This is done as follows. First on the two sides of the critical cell when the reconstruction requires the data of the cell-averages on the other side. we let it use extrapolated data instead of the original data. The resulting reconstruction is denoted by $\hat{R}(x;u^n)$.

$\hat{R}(x;u^n)$ maintains (3.1) since the reconstruction in each cell $[x_{j-1/2}, x_{j+1/2}]$ still uses the original datum of cell-averages u_j^n in this cell. If both the extrapolation and original $R(x;u^n)$ are of the rth order $\hat{R}(x;u^n)$ maintains also (3.2) separately on the two sides of the critical cell because of the usage of the extrapolated data. As a result, $\hat{R}(x;u^n)$ has a big jump at the grid point $x_{j_1+1/2}$ (see Figure 3.1-(b)). We denote by $\hat{R}_l(x;u^n)$ and $\hat{R}_r(x;u^n)$ the left and right pieces of $\hat{R}(x;u^n)$.

Next we move the big jump from $x_{j_1+1/2}$ to the discontinuity position

ξ^n by either extending or cutting off $\hat{R}_l(x;u^n)$ and $\hat{R}_r(x;u^n)$ within the interval $[\min(\xi^n, x_{j_1+1/2}), \max(\xi^n, x_{j_1+1/2})]$ (see Figure 3.1-(c)). The resulting function, denoted by $\tilde{R}(x;u^n)$, is the improved reconstruction.

Now we are going to derive the formula. As we see that the above movement of jump position will change the integral $\int_{x_{j_1}}^{x_{j_1+1}} \hat{R}(x;u^n)$ by a difference

$$\int_{x_{j_1+1/2}}^{\xi^n} (\hat{R}_r(x;u^n) - \hat{R}_l(x;u^n))dx \tag{3.6}$$

We let this difference equal to $q_{j_1}^n h$ by the consideration of conservation; i.e.,

$$\int_{x_{j_1+1/2}}^{\xi^n} (\hat{R}_r(x;u^n) - \hat{R}_l(x;u^n))dx = q_{j_1}^n h, \tag{3.7}$$

which gives the formula relating the conservation error and the discontinuity position.

When the reconstruction $R(x;u^n)$ is (3.3), (3.7) is

$$(u_{j_1+1}^n - u_{j_1}^n)(\xi^n - x_{j_1+1/2}) = q_{j_1}^n h, \tag{3.8}$$

from which we obtain

$$\xi^n = x_{j_1+1/2} + \frac{q_{j_1}^n h}{u_{j_1+1}^n - u_{j_1}^n}. \tag{3.9}$$

Actually, (3.9) is used in all the numerical examples in section 6. When the reconstruction $R(x;u^n)$ is (3.4), (3.7) is

$$A_{j_1+1/2}^n(\xi^n - x_{j_1+1/2}) + B_{j_1+1/2}^n(\xi^n - x_{j_1+1/2})^2 = q_{j_1}^n h, \tag{3.10}$$

where

$$A_{j_1+1/2}^n = \frac{3}{2}(u_{j_1+1}^n - u_{j_1}^n) - \frac{1}{2}(u_{j_1+2}^n - u_{j_1-1}^n) \tag{3.11}$$

and

$$B_{j_1+1/2}^n = \frac{1}{2h}(u_{j_1+2}^n - u_{j_1+1}^n - u_{j_1}^n + u_{j_1-1}^n). \tag{3.12}$$

(3.10) is a second order algebraic equation, which has a single root in the vicinity of x_{j_1} when the jump is of $O(1)$ and h is small enough.

When the initial condition (1.1b) contains a shock or a contact discontinuity, (3.7) can also be used to calculate their initial conservation errors in the discretization of (1.1b).

PROPOSITION 3.1 *If*

1) there is a discontinuity in the solution of (1.1), whose jump is of $O(1)$,

2) the numerical solution approximates the cell-averages of the exact solution up to $O(h^r)$ *on the two sides of the critical cell,*

3) both the extrapolation and reconstruction $R(x;u^n)$ *are of the* $(r-1)$*th order,*

4) the discretization of the initial data (1.1b) is accurate,

then the discontinuity position obtained by solving (3.7) is the r*th order accurate.*

Proof. Consider $\int_{-\infty}^{\infty} \tilde{R}(x;u^n)dx$, which exists since $u_0(x)$ is of compact support. First we shall prove that

$$\int_{-\infty}^{\infty} (\tilde{R}(x;u^n)dx - u(x,t_n))dx = 0 \tag{3.13}$$

We observe that

$$\begin{aligned} \int_{-\infty}^{\infty} \tilde{R}(x;u^n)dx &= \int_{-\infty}^{x_{j_1+1/2}} \hat{R}_l(x;u^n)dx + \int_{x_{j_1+1/2}}^{\infty} \hat{R}_r(x;u^n)dx \\ &\quad + \int_{x_{j_1+1/2}}^{\xi^n} (\hat{R}_l(x;u^n) - \hat{R}_r(x;u^n))dx \\ &= \sum_{j=-\infty}^{j_0} u_j^n h + \sum_{j=j_0+1}^{\infty} u_j^n h - q_{j_1}^n h. \end{aligned} \tag{3.14}$$

The last step in (3.14) is due to (3.1) and (3.7). Because the overall scheme maintains the conservation of $\sum_{j=-\infty}^{\infty} (u_j^n - q_j^n)$, we have

$$\sum_{j=-\infty}^{\infty} u_j^n h - q_{j_1}^n h = \sum_{j=-\infty}^{\infty} (u_j^0 - q_j^0)h. \tag{3.15}$$

Here if the discontinuity on the initial time level is in the cell $[x_{j_0-1/2}, x_{j_0+1/2}]$, then except $q_{j_0}^0$, all q_j^0's are zero; and if the discontinuity develops later in the computation, then all q_j^0's are zero. Because of assumption 4) the RHS

of (3.15) is equal to $\int_{-\infty}^{\infty} u_0(x)dx$. Due to the conservation of (1.1a)

$$\int_{-\infty}^{\infty} u_0(x)dx = \int_{-\infty}^{\infty} u(x,t_n)dx. \tag{3.16}$$

Combining (3.14)-(3.16) we obtain (3.13).

Denote the exact discontinuity position by s^n. Next we shall prove that

$$|s^n - \xi^n| = O(h^r). \tag{3.17}$$

Suppose $s^n < \xi^n$. We see that

$$\begin{aligned}\int_{-\infty}^{s^n} (\tilde{R}(x;u^n) - u(x,t_n))dx &= \int_{-\infty}^{s^n} (\hat{R}_l(x;u^n) - u(x,t_n))dx \\ = \sum_{j=-\infty}^{j_1-1} (u_j^n h - \int_{x_{j-1/2}}^{x_{j+1/2}} u(x,t_n)dx) &+ \int_{x_{j_1-1/2}}^{s^n} (\hat{R}_l(x;u^n) - u(x,t_n))dx.\end{aligned} \tag{3.19}$$

Due to assumption 2) and 3) both the two terms on the RHS of (3.19) are of $O(h^r)$. By a similar argument we can also show

$$\int_{\xi^n}^{\infty} (\tilde{R}(x;u^n) - u(x;t_n))dx = O(h^r). \tag{3.20}$$

We substitute (3.19) and (3.20) into (3.13) and obtain

$$\int_{s^n}^{\xi^n} (\tilde{R}(x;u^n) - u(x,t_n))dx = O(h^r). \tag{3.21}$$

which means that (3.17) must be true due to assumptions 1) and 2). The case that $s^n > \xi^n$ can be proved in the same way.

Thus completes the proof.

We shall make the following two remarks to complete this section.

Remark 1. The reconstruction used to locate the discontinuity positions should not be the same as the original reconstruction in the underlying Godunov type schemes. In fact, assumption 3) in the theorem shows that the first reconstruction can be one order lower than the second one.

We say that a difference scheme is pointwise if its numerical solution is an approximation to the exact solution at the grid points rather its cell-averages.

Remark 2. If $r = 2$ and the underlying scheme is pointwise then all the above discussion in this section is still true. This is because that a second order approximation of a solution at x_j is also a second order approximation to its cell-average in $[x_{j-1/2}, x_{j+1/2}]$.

4. Algorithm. Interactions of Discontinuities

Now we are able to complete the algorithm of the overall scheme upon the discussion in the previous two sections.

We start with the case of a single discontinuity and still use the same notations as in the last two sections. As shown in section 2 there are three possible cases for the numerical solution in the vicinity of the discontinuity, namely that the critical cell on the next time level either remains in the same cell or moves to the left or right. A key point of the algorithm is to determine which case should happen.

The algorithm in [20] calculates the potential conservation errors on the next time level for all the three cases and then take the one with the least absolute value of the conservation error among them.

The algorithm in [22] calculates the discontinuity position on the next time level by Hugoniot condition; i.e.,

$$\xi^{n+1} = \xi^n + \tau \frac{f(u^n_{j_1+1}) - f(u^n_{j_1})}{u^n_{j_1+1} - u^n_{j_1}}, \tag{4.1}$$

and then makes the option according to ξ^{n+1}. In doing so, the corresponding overall scheme is not conservative; however, we proved that the conservation error for a single discontinuity is uniformly bounded if the solution is piecewise smooth.

In this paper we design the algorithm as follows: First, it computes ξ^{n+1} by (4.1) and makes the option according to it. Then it computes the numerical solution on the next time level; meanwhile, it computes also the conservation error on that level. Finally, it recomputes ξ^{n+1} through the conservation error by (3.7). In doing so, the overall scheme maintains the conservation for the numerical solution.

As one can see, the recomputed ξ^{n+1} may deviate a little bit from its critical cell. This kind of small deviations will also happen in the interactions of discontinuities described later in this section. However, numerical

experiments show that it does not cause any problems. This is because the tracking technique is based on the conservation; therefore, the corresponding numerical results are not sensitive to the discontinuity positions.

In the following discussion when saying discontinuity positions we always mean the recomputed discontinuity positions.

Now we are going to deal with the interactions of discontinuities and develop the so-called "stacking technique". As we shall see, this technique lets the computation proceed still on the regular grid. Just for the simplicity, we shall restrict our discussion only for the case of two discontinuities. The extension of the treatment for cases of several discontinuities are naive.

Assume that there are two critical cells $[x_{j_l}, x_{j_l+1}]$ and $[x_{j_r}, x_{j_r+1}]$ on the nth time level, where $j_l \leq j_r$. First, if the two critical cells are separated, which means $j_l < j_r$, then each of them can be handled in a way as for a single critical cell. The two critical cells may be close to each other, probably they are so close that the grid points between them are not enough to implement extrapolation with the order required by the technique. In this case the order of the extrapolation in this region has to be lowered to the highest order that can be archived with the grid points inside. Although this means that we lose some accuracy, the numerical experiments show that the problem is insignificant.

If the two critical cells approach to each other when the time evolves, then at certain moment, say on the nth time level, they will move into the same cell (see Figure 4.1). We denote the cell by $[x_{j_1}, x_{j_1+1}]$, the left discontinuity position by ξ_l^n, and the right discontinuity position by ξ_r^n, respectively. There are two possible cases, namely $\xi_l^n \leq \xi_r^n$, which means that the two discontinuities have not crossed over each other, and $\xi_l^n > \xi_r^n$, which means that they have crossed over each other.

In the first case we stack the two critical cells in the cell $[x_{j_1}, x_{j_1+1}]$; i.e., each of them is still regarded as an individual critical cell. We shall choose a middle state u_* to connect these two stacked critical cells. It is chosen as follows: In case a in Figure 4.1 $u_* = u_{j_1+1}^{n-1}$, in case b $u_* = u_{j_1}^{n-1}$ and in case c $u_* = \frac{1}{2}(u_{j_1}^{n-1} + u_{j_1+1}^{n-1})$. Algorithmly, we never let case d happen by holding one of the critical cells.

In the following computation we treat the left critical cell as that the numerical solution on its right is u_* and the right one as that the numerical

solution on its left is u_*. Therefore, during the time we have two discontinuity positions ξ_l^n and ξ_r^n and two conservation errors $q_{j_1,l}^n$ and $q_{j_1,r}^n$ in the same cell. This situation will continue as long as the two discontinuity positions have not crossed over each other.

These two stacked critical cells will be separated again if ξ_l^n and ξ_r^n move to different cells without crossing over each other (see Figure 4.2). If this happen, we shall let the numerical solution at the grid points between the two re-separated critical cells be u_*. Algorithmly, we never let the two critical cells cross over each other by holding one of them, even though its discontinuity position may deviate a little bit from it.

Next we shall show that the overall scheme for these two stacked critical cells is also able to be written in the form of (2.12) with $q_j^n = q_{j,l}^n + q_{j,r}^n$ and $q_j^{n+1} = q_{j,l}^{n+1} + q_{j,r}^{n+1}$. This means that the numerical solution together with the conservation errors are still conserved. To do this we observe the following two initial values on the nth time level.

1)

$$u_{j,l}^n = \begin{cases} u_j^n & j \le j_1 \\ u_* & j > j_1 \end{cases} \tag{4.2}$$

with a critical cell $[x_{j_1}, x_{j_1+1}]$ and conservation errors

$$q_{j,l}^n = \begin{cases} q_{j_1,l}^n & j = j_1 \\ 0 & j \ne j_1. \end{cases}$$

2)

$$u_{j,r}^n = \begin{cases} u_* & j \le j_1 \\ u_j^n & j > j_1 \end{cases} \tag{4.3}$$

also with a critical cell $[x_{j_1}, x_{j_1+1}]$ and conservation errors

$$q_{j,r}^n = \begin{cases} q_{j_1,r}^n & j = j_1 \\ 0 & j \ne j_1. \end{cases}$$

The corresponding numerical solutions on the following time level satisfy

$$u_{j,l}^{n+1} = u_{j,l} - \lambda(\tilde{f}_{j+1/2,l}^n - \tilde{f}_{j-1/2,l}^n) + p_{j+1/2,l}^n - p_{j-1/2,l}^n + q_{j,l}^{n+1} - q_{j,l}^n \tag{4.4}$$

and

$$u_{j,r}^{n+1} = u_{j,r} - \lambda(\tilde{f}_{j+1/2,r}^n - \tilde{f}_{j-1/2,r}^n) + p_{j+1/2,r}^n - p_{j-1/2,r}^n + q_{j,r}^{n+1} - q_{j,r}^n, \tag{4.5}$$

where, according to (2.11), $\tilde{f}^n_{j+1/2,l}$ and $\tilde{f}^n_{j+1/2,r}$ are

$$\tilde{f}^n_{j+1/2,l} = \begin{cases} \hat{f}^{n,-}_{j+1/2} & j < j_1 \\ f(u_*) & j \geq j_1 \end{cases} \tag{4.6}$$

and

$$\tilde{f}^n_{j+1/2,r} = \begin{cases} f(u_*) & j < j_1 \\ \hat{f}^{n,+}_{j+1/2} & j \geq j_1 \end{cases}, \tag{4.7}$$

and $p_{j+1/2,l}$'s, $q^{n+1}_{j,l}$'s, $p^n_{j+1/2,r}$'s and $q^{n+1}_{j,r}$'s are defined in the way described in section 2. Particularly, the nonzero terms in the conservation errors are just the nonzero terms in the conservation errors of the original problems.

It is easy to verify by the definition that on the nth time level

$$u^n_{j,l} + u^n_{j,r} = u^n_j + u_* \tag{4.8}$$

and

$$\tilde{f}^n_{j+1/2,l} + \tilde{f}^n_{j+1/2,r} = \tilde{f}^n_{j+1/2} + f(u_*) \tag{4.9}$$

Since the two critical cells will not cross over each other, it is also easy to verify that

$$u^{n+1}_{j,l} + u^{n+1}_{j,r} = u^{n+1}_j + u_* \tag{4.10}$$

in all cases. Therefore, by adding up (4.4) and (4.5) and substituting (4.8)-(4.10) into it we obtain the conclusion.

This stacking technique can be naively extended to treat several stacked critical cells. In this case we have several discontinuity positions and conservation errors in the same cell. The author would particularly like to point out that here we have developed a new approach to deal with the region where the numerical solution has small scale structures. As well known, the usual way to resolve the details of the small scale structures is to subdivide the grid in this region. However, this approach has the following two shortcomings.

1) Due to the restriction of CFL condition the time step in the subdivided region has to be reduced, which will slow the computation.

2) One needs to take care of the coordination between the subdivided and un-subdivided regions.

Instead of subdividing the grid, the stacking technique developed here stores more than one pieces of information, namely the discontinuity positions

and conservation errors, in the same cell to resolve the details of the small scale structures. In doing so it does not have the shortcomings mentioned above.

Now we are going to deal with the second case in which $\xi_l^n > \xi_r^n$. If this happens on the nth time level, we merge the two critical cells to form a new critical cell and take $q_{j_1}^n = q_{j_1,l}^n + q_{j_1,r}^n$ to be the conservation error for the new one by the consideration of conservation. The discontinuity position ξ^n of the new critical cell is computed by (3.7) through the new conservation error.

Obviously, under the same assumptions and by the same arguments of Proposition 3.1 we are able to prove that the new discontinuity position will be accurate up to a certain order in the later computation. As a verification, we shall observe the situation for piecewise constant solutions and show that in this case the handling of the mergence of two critical cells is equivalent to the ordinary tracking methods based on Hugoniot condition and Riemann problems.

We assume that

1) the numerical solution on the left and right of the stacked critical cells are u_l and u_r, respectively;

2) on the $(n-1)$th time level $\xi_l^{n-1} \leq \xi_r^{n-1}$ and on the nth time level $\xi_l^n > \xi_r^n$.

We shall compute ξ^n using the present tracking technique as well as using Hugoniot condition and solving the Riemann problem, and show the results are the same.

First we compute ξ^n using the present tracking technique. As said before in this section, the discontinuity positions are computed through the conservation errors; therefore, we have

$$q_{j_1,l}^n h = (\xi_l^n - x_{j_1+1/2})(u_* - u_l) \tag{4.11}$$

and

$$q_{j_1,r}^n h = (\xi_r^n - x_{j_1+1/2})(u_r - u_*). \tag{4.12}$$

Then the discontinuity position computed by the technique for the new critical cell satisfies

$$q_{j_1,l}^n h + q_{j_1,r}^n h = (\xi^n - x_{j_1+1/2})(u_r - u_l). \tag{4.13}$$

By substituting (4.11) and (4.12) into (4.13) we obtain

$$\xi^n = \frac{\xi_l^n(u_l - u_*) + \xi_r^n(u_r - u_*)}{u_r - u_l}. \tag{4.14}$$

Second we compute ξ^n using Hugoniot condition and solving the Riemann problem. For the simplicity of discussion we assume that all the ξ_l^{n-1}, ξ_r^n, ξ_r^{n-1} and ξ_r^n are in the same cell $[x_{j_1}, x_{j_1+1}]$ (see Figure 4.3). We draw lines to connect points (ξ_l^{n-1}, t_{n-1}) and (ξ_l^n, t_n) and points (ξ_r^{n-1}, t_{n-1}) and (ξ_r^n, t_n). Since the numerical solution is piecewise constant, the slopes of the two line segments are

$$s_l = \frac{f(u_*) - f(u_l)}{u_* - u_l} \tag{4.15}$$

and

$$s_r = \frac{f(u_r) - f(u_*)}{u_r - u_*}. \tag{4.16}$$

The two line segments must intersect at a point $(\tilde{\xi}, \tilde{t})$ because of the assumption 2, which means that the two discontinuities collide at $\tilde{\xi}$ and the time $\tilde{t}$ to form a new one. It is easy to see that

$$\xi_l^n = \tilde{\xi} + s_l(t_n - \tilde{t}) \tag{4.17}$$

and

$$\xi_r^n = \tilde{\xi} + s_r(t_n - \tilde{t}) \tag{4.18}$$

We observe that the the slope of the new discontinuity, is

$$s = \frac{f(u_r) - f(u_l)}{u_r - u_l}; \tag{4.19}$$

therefore, the discontinuity position ξ^n is

$$\xi^n = \tilde{\xi} + s(t_n - \tilde{t}). \tag{4.20}$$

To compute ξ^n, we multiply (4.17), (4.18) and (4.20) by $u_* - u_l$, $u_r - u_*$ and $u_r - u_l$, respectively, and subtract the first two equalities from the last one. By substituting (4.15), (4.16) and (4.19) into the resulting equality we can also obtain (4.14). Thus the technique is equivalent to the ordinary shocking tracking methods.

5. Application of the Technique to System Case

In this section we shall apply the tracking technique developed in the previous sections to the system described by (1.1)-(1.4).

As well known, the system (1.1) of m un-knowns has m different kinds of characteristics and, therefore, has m different kinds of discontinuities too (see [17]). Due to this reason the critical cells in the system case will also be classified in m kinds according to the discontinuities they contain. We say a single critical cell $[x_{j_1}, x_{j_1+1}]$ on the nth time level is a $k-$critical cell if the solution of the Riemann problem $R(u^n_{j_1}, u^n_{j_1+1})$ with $u^n_{j_1}$ and $u^n_{j_1+1}$ as the left and right states has a strong $k-$discontinuity. The stacked critical cells can be classified accordingly.

For a single $k-$critical cell $[x_{j_1}, x_{j_1+1}]$, we solve the Riemann problem $R(u^n_{j_1}, u^n_{j_1+1})$ to find out the moving speed of the $k-$discontinuity, and then use it to calculate the predict position of ξ^{n+1}, by which we are able to make the option for the computation of the numerical solution. We still compute the numerical solution on the two sides of the critical cell using the extrapolated data on the same sides and compute the conservation error through (2.13), (2.14) or (2.15), which are now in vector forms.

Since there are m different characteristic fields, the computed conservation error may contain information that belongs to the fields other than the $k-$field. This information of the other fields has to be cleaned from the conservation error and go to the numerical solution by the consideration of conservation. By doing so, the information on one side of the discontinuity that associates with the characteristics on its other side is able to travel through it to the other side. In the following of this section, a so-called "clean-up" step is developed to fulfill this task.

For a single critical cell, we solve the Riemann problem $R(u^n_{j_1}, u^n_{j_1+1})$, whose solution consists of m waves, discontinuities or centered simple waves, and $m-1$ middle states $u^*_1, u^*_2, \cdots, u^*_{m-1}$. We denote

$$e_l = u^*_l - u^*_{l-1}, \quad 1 \le l \le m, \tag{4.1}$$

where $u^*_0 = u^n_{j_1}$ and $u^*_m = u^n_{j_1+1}$. Then we linearly decompose the conservation error through the $\{e_l\}_{l=1,m}$, namely

$$q^n_{j_1} = \alpha_1 e_1 + \alpha_2 e_2 + \cdots + \alpha_m e_m, \tag{4.2}$$

and then let

$$q^n_{j_1} := \alpha_k e_k. \tag{4.3}$$

The discontinuity position ξ^n will be computed by (3.9) with $u^n_{j_1} = u^*_l$ and $u_{j_1+1} = u^*_{l+1}$.

As we see, if the solution is of two constants u_l and u_r connected with a $k-$discontinuity, then the flux difference vector $f(u_r) - f(u_l)$ is parallel to the solution jump vector $u_r - u_l$ according to Hugoniot condition. As a result, the conservation error of the critical cell is always parallel to $u_r - u_l$ too. Under this consideration we pick the kth term in (4.2), which is the only term parallel to $u_r - u_l$ in the decomposition, as the conservation error.

The rest terms in (4.2) will go to the numerical solution in the fashion that

$$u^n_{j_1} := u^n_{j_1} - \sum_{l=1}^{k-1} \alpha_l e_l \tag{4.4}$$

and

$$u^n_{j_1+1} := u^n_{j_1+1} - \sum_{l=k+1}^{m} \alpha_l e_l. \tag{4.5}$$

In doing so, the information associated with the characteristics on the left or right of the discontinuity goes to the numerical solution on the left or right, respectively. At the meantime $\sum_{j=-\infty}^{\infty} (u^n_j - q^n_j)$ is maintained to be conserved.

The developed clean-up step has only the first order accuracy if the solution is piecewise smooth. This is because it is designed assuming that the numerical solution are piecewise constant on the two sides of the critical cells. Nevertheless, the numerical experiments show that the loss of the accuracy is insignificant. The higher order clean-up steps are likely able to be developed on the basis of the discussion in the previous section; however, they are still expected.

For stacked critical cells, the clean-up step is constructed accordingly as follows: When there is an another critical cell stacked in the same cell on the left with a middle state u_* connected with the concerned critical cell, we clean up the conservation error in the same way as for a single critical cell except that $u^n_{j_1}$ is replaced by u_*. In this case the sum $\sum_{l=1}^{k-1} \alpha_l e_l$ in (4.4) will

go to u_* in the way of (4.4). At the meantime we let this sum also go to the conservation error of the left stacked critical cell in the way that

$$q^n_{j_1,l} := q^n_{j_1,l} + \sum_{l=1}^{k-1} \alpha_l e_l, \tag{4.6}$$

where $q^n_{j_1,l}$ is the conservation error of the left stacked critical cell. The last step is due to that u_* is not counted in $\sum_{j=\infty}^{\infty}(u^n_j - q^n_j)$; therefore, this sum has to be stored also somewhere in the conservation error to maintain the conservation for the numerical solution together with the conservation errors. The case that there is a critical cell stacked in the same cell on the right is handled in the same way, only the sum $\sum_{l=k+1}^{m} \alpha_l e_l$ in (4.5) will go to the conservation error of the concerned critical cell.

When two critical cells stacked in the cell $[x_{j_1}, x_{j_1+1}]$ are merged, the clean-up step is constructed as follows.

For the simplicity of discussion we assume that there is no other critical cells than these two ones stacked in the cell. First we also solve the Riemann problem $R(u^n_{j_1}, u^n_{j_1+1})$ and decompose the conservation error, which is now the sum of the conservation errors of the merged critical cells, through $\{e_l\}_{l=1,m}$. Next we check the strength of each wave resulting from the Riemann problem in the order that l increases, namely from the left to the right.

If the $l-$wave is a shock or contact discontinuity and strong enough according to some criterion, we set a critical cell for it in $[x_{j_1}, x_{j_1+1}]$ and pick the term $\alpha_l e_l$ in (4.2) as its conservation error. In this case if there is going to be other critical cells stacked in the same cell on its right, we set u^*_{l+1} to be the middle state on its right.

If the wave is a centered simple wave, or a shock or contact discontinuity but not strong enough, we let its corresponding term $\alpha_l e_l$ in (4.2) go to the numerical solution or conservation error in the way as follows: If there is no critical cell in the same cell on the left, then this term will go to $u^n_{j_1}$ in the way that

$$u^n_{j_1} := u^n_{j_1} - \alpha_l e_l. \tag{4.7}$$

Otherwise, if there is a critical cell in the same cell on the left, then this term will go to the right middle state of the left critical cell in the way of (4.7).

Meanwhile, it will also go to the conservation error of the left critical cell in the way that

$$q^n_{j_1,l} := q^n_{j_1,l} + \alpha_l e_l \tag{4.8}$$

by the consideration of conservation, where $q^n_{j_1,l}$ is the conservation error of the left critical cell.

As we see, if all the waves resulting from a Riemann problem $R(u_l, u_r)$ are shocks and contact discontinuities, then in the linear decomposition of $f(u_r) - f(u_l)$ through $\{u^*_{l+1} - u^*_l\}_{l=1,m}$, namely

$$f(u_r) - f(u_l) = \alpha_1(u^*_1 - u^*_0) + \alpha_2(u^*_2 - u^*_1) + \cdots + \alpha_m(u^*_m - u^*_{m-1}), \tag{4.9}$$

the coefficient of the lth term α_l is just the speed of the l-wave, where u^*_l is the lth middle state with $u^*_0 = u_l$ and $u^*_m = u_r$. This is because of Hugoniot condition and the fact that $\{u^*_{l+1} - u^*_l\}_{l=1,m}$ are linearly independent.

In section 4 we showed that in the scalar case, when the solution is piecewise constant the handling of the mergence of two critical cells is equivalent to the ordinary tracking methods. In the system case, by noticing the above mentioned fact we are also able to show this equivalence following the same argument in section 4.

The extension to the case of several critical cells stacked in the same cell is naive.

6. Numerical Examples

In this section we shall present two numerical examples for the Euler equations of gas dynamics for polytropic gas. The Euler equations are

$$u_t + f(u)_x = 0, \tag{6.1a}$$

$$u = (\rho, m, E)^T, \tag{6.1b}$$

$$f(u) = qu + (0, p, qp)^T \tag{6.1c}$$

$$p = (\gamma - 1)(E - \frac{1}{2}pq^2), \tag{6.1d}$$

where ρ, q, p and E are the density, velocity, pressure and total energy, respectively, $m = \rho q$ is the momentum and γ is the ratio of specific heats. The eigenvalues of the Jacobian matrix $A(u) = \partial f/\partial u$ are

$$a_1(u) = q - c, \quad a_2(u) = q, \quad a_3(u) = q + c, \tag{6.2}$$

where $c = (\gamma p/\rho)^{\frac{1}{2}}$ is the sound speed.

The underlying scheme is a second order TVD scheme with a two-step Runge-Kutta time discretization described in [31]; i.e.,

$$\begin{aligned} u_j^{n+1/2} &= u_j - \lambda(\hat{f}^n_{j+1/2} - \hat{f}^n_{j-1/2}), \\ \bar{u}_j^{n+1} &= u_j^{n+1/2} - \lambda(\hat{f}^{n+1/2}_{j+1/2} - \hat{f}^{n+1/2}_{j-1/2}), \\ u_j^{n+1} &= \frac{1}{2}(u_j^n + \bar{u}_j^{n+1}) \end{aligned} \tag{6.3}$$

and a TVD numerical flux $\hat{f}$ satisfying

$$\frac{1}{h}(\hat{f}^n_{j+1/2} - \hat{f}^n_{j-1/2}) = f_x|_{x=x_j} + O(h^2). \tag{6.4}$$

The mesh ratio is set to satisfy $\lambda \max_j(u_j - c_j, u_j, u_j + c_j) \leq 0.5$.

The data structure for the critical cells is as follows: A doubly linked list is used for the critical cells. Each critical cell is an element of this list with pointers to the left and right neighboring critical cells. With such a data structure it is easy to insert new critical cells. Whether the neighboring critical cells are stacked in the same cell or separated is judged by checking their x indexes.

A special advantage of this shock tracking technique is that it does not require adaptive grid, the whole computation proceeds still on the regular grid. Thus, the algorithm of the overall scheme is much simpler than ordinary shock tracking methods. It is able to be programmed in an almost shock capturing fashion by regarding the technique as an adjustment on the underlying scheme in the vicinity of discontinuities.

The algorithm consists of the following steps:

1) Compute the numerical solution without considering the critical cells.

2) Recompute the numerical solution near the critical cells using the extrapolated data on the same side and regarding that all the critical cells do not move.

3) Compute the conservation errors regarding that all the critical cells do not move.

4) Compute the predict discontinuity position in each critical cell by Hugoniot condition and determine whether the critical cell should move or not.

5) For the critical cells that should move, recompute the numerical solution nearby and the corresponding conservation errors in the way described in section 2 if they are single critical cells, or recomputed the corresponding middle states and conservation errors in the way described in section 4 if they are stacked with other critical cells in the same cells.

6) Clean up the conservation errors and recomputed the discontinuity positions in the critical cells with the cleaned conservation errors in the way described in section 5.

EXAMPLE 1. The initial data are

$$u_0 = \begin{cases} u_l & 0 \le x < 0.1 \\ u_m & 0.1 \le x < 0.9 \\ u_r & 0.9 \le x \le 1 \end{cases} \tag{6.5}$$

where

$$\begin{aligned} &\rho_l = \rho_m = \rho_r = 1, \\ &q_l = q_m = q_r = 0, \\ &p_l = 10^3, p_m = 10^{-2}, p_r = 10^2. \end{aligned} \tag{6.6}$$

A solid wall boundary condition is applied to the two ends $x = 0$ and $x = 1$. This is the blast wave problem suggested by Colella and Woodward in [36]. We refer readers to [36] and [13] for the details and comparisons of various numerical results of this problem.

The numerical results at the finally time $t = 0.038$ are presented in Figure 6.1-(a) to Figure 6.1-(c), where (a) is of the density, (b) is of the velocity, and (c) is of the pressure. The circles represent the numerical solution computed by the overall scheme with the shock tracking technique with 400 grid points. The solid lines present the numerical solution computed by a second order ENO scheme with 800 grid points for the comparison. Figure 6.1-(d) presents the tracked discontinuities. We refer readers to the contour figure of the numerical solution in the $x - t$ plane presented in [36] for the comparison with Figure 6.1-(d).

EXAMPLE 2. The initial data are

$$\rho_0(x) = 1,$$
$$q_0(x) = 0,$$
$$p_0(x) = \begin{cases} 460 & 0 \le x \le 0.1775 \\ 10 + 50(9-l) & 0.05(l-1) < x - 0.1775 \le 0.05l \\ & 1 \le l \le 8 \\ 10 & 0.5775 < x \le 0.1. \end{cases} \tag{6.7}$$

The same solid wall boundary condition as in the previous example is applied to the end $x = 1$, and the following symmetric boundary condition is applied to the end $x = 0$,

$$u^n_{-j} = u^n_j \quad j = 1, 2, \cdots, k, \tag{6.8}$$

where k is the semi-length of the stencil of the numerical flux.

We see that the initial pressure contains nine big jumps, each of which is of 50. As a result, the solution to the problem has nine strong left shocks and a contact discontinuity starting from the initial time level. When the time evolves, they interact on each other and finally all the left shocks merged into a strong left shock (see Figure 6.2-(d)).

We shall test our shock tracking technique on this problem in two different ways to demonstrate its robustness in dealing with solutions with small scale structure and capturing spontaneous shocks.

First we set critical cells for all the discontinuities on the initial time level and track them afterwards. The numerical results at the final time $t = 0.026$ are presented in Figure 6.2-(a) to Figure 6.2-(c), where (a) is of the density, (b) is of the velocity, and (c) is of the pressure. The circles represent the numerical solution computed by the overall scheme with the shock tracking technique with 200 grid points. The solid lines present the numerical solution computed only by the underlying scheme with 1600 grid points for the comparison.

The numerical results on the 200-point grid computed only by the underlying scheme is not very good since the peak in the density profile around $x = 0.8$ does not shoot up well.

Figure 6.2-(d) presents the tracked discontinuities and Figure 6.2-(e) presents a picture of the critical cells, which is drawn by linking the left and right endpoints of the critical cells on the neighboring time level relat-

ing to the same discontinuities by line segments, respectively. The final left shock position at the time $t = 0.026$ is 0.8136.

Figure 6.2-(f) presents a picture of the critical cells in the region $0.55 \leq x \leq 0.75$ and $0.015 \leq t \leq 0.02$, i.e. the region marked by dash lines in Figure 6.1-(e). Because of the small scale structure of the solution the region is crowded with critical cells and stacking and mergence of critical cells happen a lot.

There are some small oscillations in the velocity and pressure profiles. These oscillations must come from the shock tracking technique since the TVD underlying scheme is supposed not to produce oscillations. An investigation of how to maintain the TVD property for the technique is underway. It is seems that care needs to be taken also for rarefaction waves, particularly when shocks of the same type or shocks and contact discontinuities interact on each other and strong rarefaction waves are generated.

Next we do not set the discontinuities on the initial time level; instead, we choose a tolerance number ε and use it to detect discontinuities in the computation. When in a cell $[x_j, x_{j+1}]$ on the nth time level the jump of the pressure is greater than ε, we solve the Riemann problem $R(u_j^n, u_{j+1}^n)$. When the solution of $R(u_j^n, u_{j+1}^n)$ contains a shock with the jump of the pressure greater than ε, we set a critical cell for this shock in the cell. At the time that the critical cell is set the conservation error is 0.

We test for $\varepsilon = 15$, 20 and 25, and the numerical results are presented in Figure 6.3, Figure 6.4 and Figure 6.5, respectively. All the (a) figures are of density profiles at the time $t = 0.026$, all the (b) figures are of velocity profiles at the same time level, and all the (c) pictures are of the tracked discontinuities. The final left shock positions are 0.8141 for $\varepsilon = 15$, 0.8143 for $\varepsilon = 20$, and 0.8142 for $\varepsilon = 25$. We see that the difference among the final left shock positions in the above four cases (including the first case of tracking the discontinuities from the initial time level) is of $O(h^2)$, which agrees with the analysis in section 3.

7. Conclusion

The shock tracking technique based on conservation presented in this paper is efficient and robust in deal with solutions with complicated structures and spontaneous shocks. The technique is also very simple since it does not

require adaptive grid and the computation proceeds still on regular grid. This makes the coding and application of the technique quite easy. Things that need more investigation are 1) how to maintain the TVD property when it is applied to TVD underlying schemes, 2) how to improve the order of accuracy when it is applied to the system case, and 3) how to extend the technique to two space dimensions. 1) and 3) are underway now.

Acknowledgments

The author thanks Professor A. Harten and Professor S. Osher for the helpful discussion with them. The author thanks the members of Mathematics Department in UCLA for their hospitality and support. Most of the numerical analysis and computation of this work was completed when the author was a visiting scholar invited by Professor S. Osher in UCLA. Finally the author thanks Professor R. Jeltsch and the members of the Seminar für Angewandte Mathematik in ETH for their hospitality and support. This paper was written out when the author was visiting ETH invited by Professor R. Jeltsch.

References

[1] I-L Chern, J. Glimm, U. McBryan, B. Plohr and S. Yaniv, J. Comput. Phys. **62**, 83 (1986).

[2] I-L Chern and P. Colella, *A conservative front tracking method for hyperbolic system of conservation laws*, preprint, (1986), to appear in J. comput. Phys.

[3] P. Colella and P.R. Woodward, J. Comput. Phys. **54**, 174 (1984).

[4] B. Sjogreen and B. Engquist, In Proceedings of the Third International Conference on Hyperbolic Problems, Uppsala, Sweden, 1990, edited by B. Engquist and B. Gustafsson, (Studentlitteratur, Sweden, 1991), p. 848.

[5] J. Glimm, Comm. Pure Appl. Math. **41**, 569 (1988).

[6] J. Glimm, J.Grove, B. Lindquist, O. McBryan and G. Tryggvason, SIAM J. Sci. Stat. Comput. **9**, 61 (1988).

[7] J. Glimm, O. McBryan, R. Menikoff, and D. Sharp, SIAM J. Sci. Stat. Comput. **7**, 230 (1986).

[8] J. Glimm and McBryan, Adv. Appl. Math. **6**, 422 (1985).

[9] J. Glimm, C Klingenberg, O. McBryan, B. Plohr, D. Sharp and S. Yaniv, Adv. Appl. Math. **6**, 259 (1985).

[10] J. Glimm, E. Isaacson, D. Marchesin and O. McBryan, Adv. Appl.Math. **3**, 91 (1981).

[11] J. Grove, Adv. Appl. Math. **10**, 201 (1989).

[12] A. Harten, J. Comput. Phys. **83**, 148 (1989)

[13] A. Harten, B. Engquist, S. Osher and S.R. Chakravarthy, J. Comput. Phys. **71**, 148 (1987)

[14] A. Harten and S. Osher, SIAM, J. Numer. Anal. **24**, 179 (1987).

[15] A. Harten, J.M. Hyman and P.D. Lax, Comm. Pure Appl. Math. **29**,297 (1976).

[16] W.D. Henshaw, J. Comput. Phys. **68**, 25 (1987).

[17] P.D. Lax, J. Comm. Pure Appl. Math. **10**, 537 (1957).

[18] R.J. LeVeque and K.M. Shyue, *Shock tracking based on high resolution wave propagation methods*, Technical Report No.92-3, University of Washington, Seattle, (1992).

[19] R.J. LeVeque, SIAM J. Numer. Anal. **22**, 1051 (1985).

[20] D. Mao, J. Comput. Phys. **92**, 422 (1991).

[21] D. Mao, In Proceedings of the Third International Conference on Hyperbolic Problems, Uppsala, Sweden, 1990, edited by B. Engquist and B. Gustafsson, (Studentlitteratur, Sweden, 1991).

[22] D. Mao, *A treatment of discontinuities in finite difference methods*, UCLA CAM Report 91-19 (1991), to appear in J. Comput. Phys.

[23] D. Mao, *A treatment of discontinuities in finite difference methods in two space dimension*, UCLA CAM Report 91-30 (1991), to appear in J. Comput. Phys.

[24] D. Mao, J. Comput. Math. No. 3, 356 (1985) (in Chinese).

[25] D. Mao, *Towards a shock tracking technique based on conservation in two space dimensions*, in preparation.

[26] Moretti, *Thoughts and afterthoughts about shock computations*, Report No. PIBAL-72-37, Polytechnic Insitute of Brooklyn, 1972.

[27] S. Osher, SIAM Numer. Anal. **21**, 217 (1984).

[28] S. Osher and S. Chakravarthy, SIAM Numer. Anal. **21**, 955 (1984).

[29] R. Richtmyer and K. Morton, *Difference methods for initial value problems*, Interscience, New York, 1967.

[30] P.L. Roe, J. Comput. Phys. **43**, 357 (1981).

[31] Chi-Wang Shu, Math. Comp. **49**, 105 (1987).

[32] B.K. Swartz and B. Wendroff, Appl. Numer. Math. **2**, 385 (1986).

[33] P.K. Sweby, SIAM J. Numer. Anal. **21**, 995 (1984).

[34] B. van Leer, J. comput. Phys. **23**, 276 (1977).

[35] B. van Leer, J. Comput. Phys. **14**, 361 (1974).

[36] P. Woodward and P. Colella, J. Comput. Phys. **54**, 115 (1984).

[37] Y.L. Zhu, X.C. Zhong, B.M. Chen, and Z.M.Zhang, *Difference methods for initial-boundary-value problems and flows around bodies*, Science Press, Beijing China, 1980.

Captions for Figures

Figure 2.1 :

The three cases for discontinuity position ξ^{n+1}. (a) ξ^{n+1} remains in $[x_{j_1}, x_{j_1+1}]$, (b) ξ^{n+1} moves to the left, and (c) ξ^{n+1} moves to the right.

Figure 2.2 :

Numerical solution on the nth time level has a jump in cell $[x_{j_1}, x_{j_1+1}]$. Solid line circles denote the original data of the numerical solution and dash line circles denote the extrapolated data from the two sides of the jump.

Figure 3.1 :

Reconstructions of numerical solution. (a) $R(x; u^n)$, which does not take into account the discontinuity. It smears the discontinuity. (b) $\hat{R}(x; u^n)$, which takes into account the discontinuity. It has a big jump at $x_{j_1+1/2}$. (c) $\tilde{R}(x; u^n)$, which is obtained by moving the big jump in $\hat{R}(x; u^n)$ to discontinuity position ξ^n. The change of integral $\int_{x_{j_1}}^{x_{j_1+1}} \hat{R}(x; u^n)dx$, i.e. the shaded area in the picture, is equal to conservation error $q_{j_1}^n$.

Figure 4.1 :

Cases of two critical cells moving into the same cell. Upward brace denotes the left critical cell while the downward brace denotes the right one. (a) The left critical cell remains in the same cell while the right one moves to the left. (b) The left critical cell moves to the right while the right one remains in the same cell. (c) The left and right critical cells move to the right and left, respectively, into the same cell. (d) The left and right critical cells move to the right and left, respectively, and cross over each other. This case does not happen in algorithm.

Figure 4.2 :

Cases of two stacked critical cells being separated. Upward brace denotes the left critical cell while the downward brace denotes the right one. (a) The left critical cell moves to the left while the right one remains in the same cell. (b) The left critical cell remains in the same cell while the right one moves to the right. (c) The left and right critical cells move to the left and right,

respectively. (d) The left and right critical cells move to the right and left, respectively. This case does not happen in algorithm.

Figure 4.3 :

Two discontinuities start from ξ_l^{n-1} and ξ_r^{n-1}, collide at point (ξ^*, t^*), and form a new discontinuity. ξ_l^n and ξ_r^n are the two discontinuity positions before the mergence of the two corresponding critical cells. ξ^n is the position of the newly formed discontinuity.

Figure 6.1 :

Example 1. Numerical solution at time $t = 0.038$. (a) Density. (b) Velocity. (c) Pressure. (d) Tracked discontinuities.

Figure 6.2 :

Example 2. Numerical solution at time $t = 0.026$. Discontinuities are set at the initial time level. (a) Density. (b) Velocity. (c) Pressure. (d) Tracked discontinuities. (e) Critical cells. (f) Critical cells in region $(0.55, 0.75) \times (0.015, 0.02)$.

Figure 6.3 :

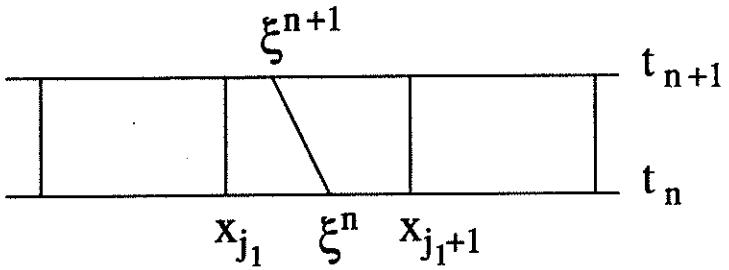
Example 2. Numerical solution at time $t = 0.026$. Discontinuities are detected with tolerance number $\varepsilon = 15$. (a) Density. (b) Velocity. (c) Tracked discontinuities.

Figure 6.4 :

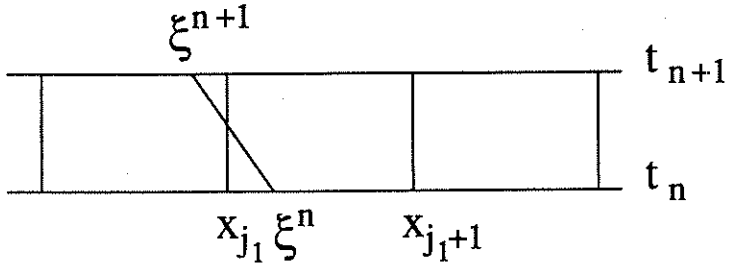
Example 2. Numerical solution at time $t = 0.026$. Discontinuities are detected with tolerance number $\varepsilon = 20$. (a) Density. (b) Velocity. (c) Tracked discontinuities.

Figure 6.5 :

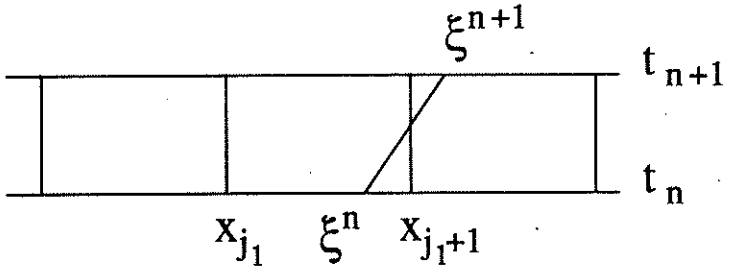
Example 2. Numerical solution at time $t = 0.026$. Discontinuities are detected with tolerance number $\varepsilon = 25$. (a) Density. (b) Velocity. (c) Tracked discontinuities.



(1)



(2)



(3)

Figure 2.1

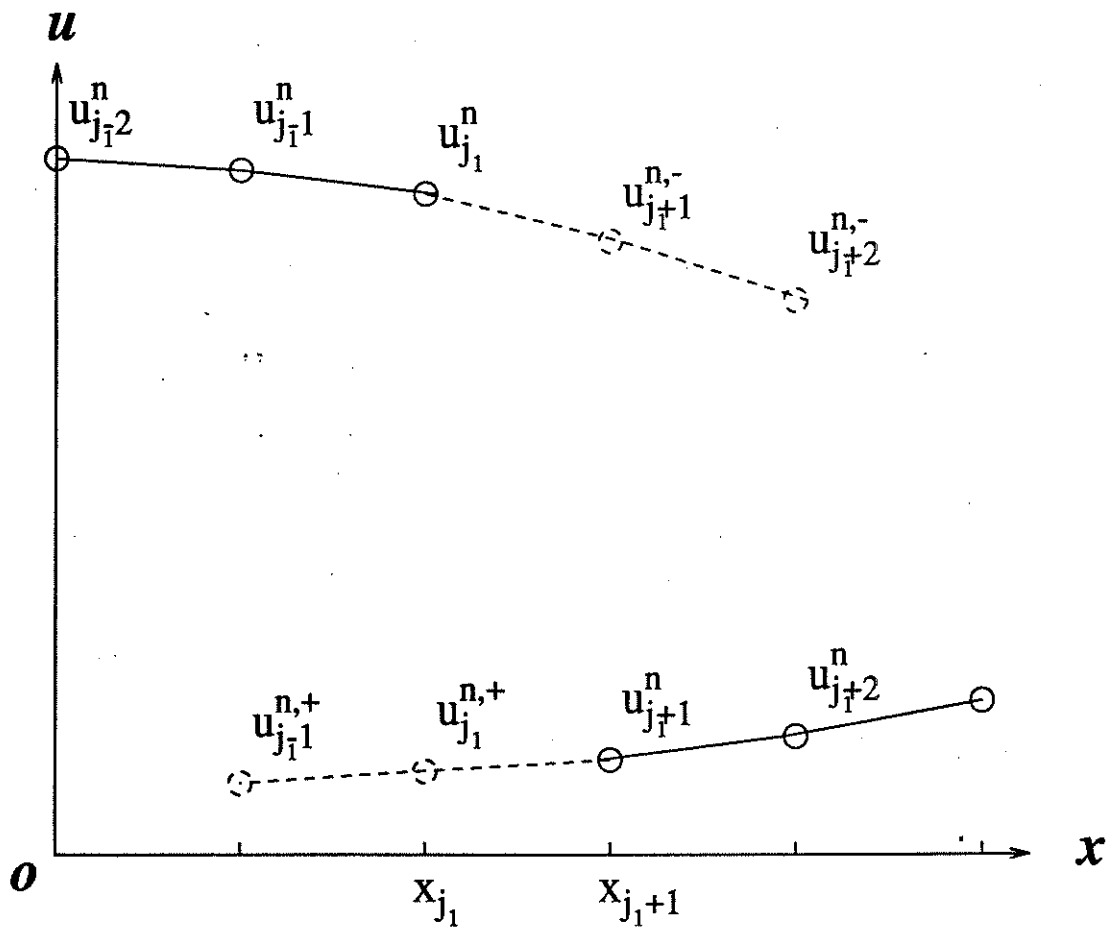

u
$u^n_{j_1-2}$
$u^n_{j_1-1}$
$u^n_{j_1}$
$u^{n,-}_{j_1+1}$
$u^{n,-}_{j_1+2}$
$u^{n,+}_{j_1-1}$
$u^{n,+}_{j_1}$
$u^n_{j_1+1}$
$u^n_{j_1+2}$
o
x_{j_1}
x_{j_1+1}
x

Figure 2.2

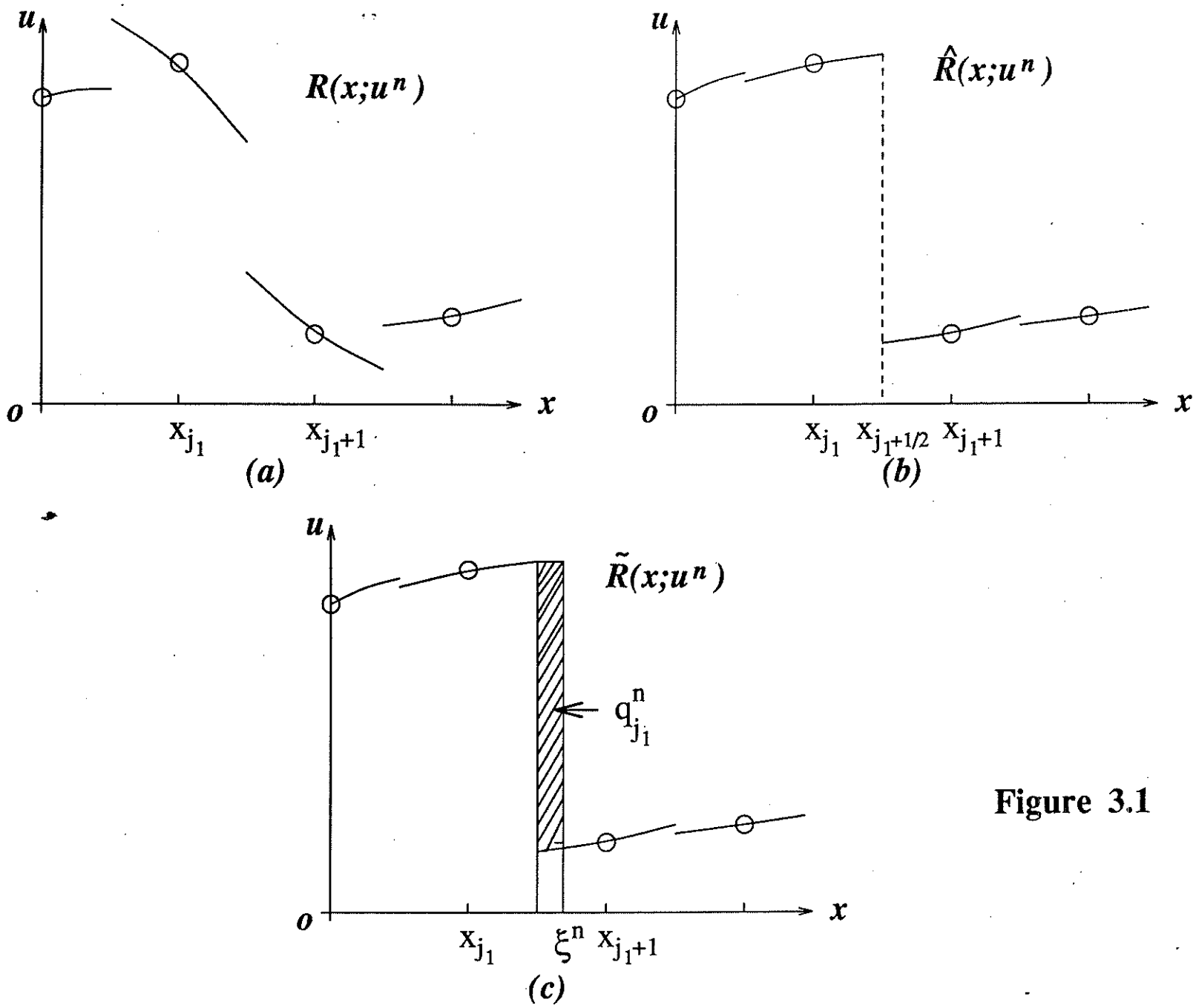
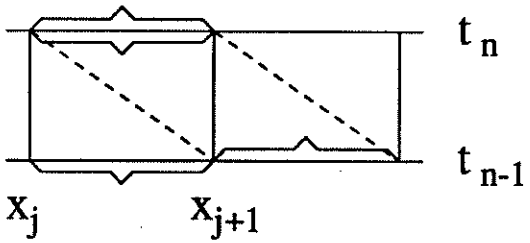
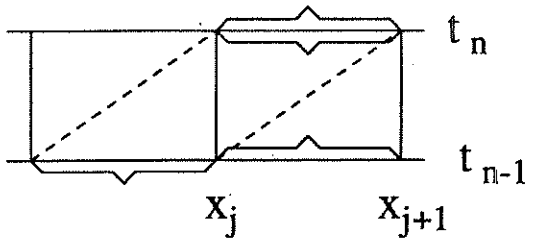

u
R(x;u^n)
o
x
x_{j_1}
x_{j_1+1}
(a)
u
R̂(x;u^n)
o
x
x_{j_1}
x_{j_1+1/2}
x_{j_1+1}
(b)
u
R̃(x;u^n)
q^n_{j_1}
o
x
x_{j_1}
ξ^n
x_{j_1+1}
(c)

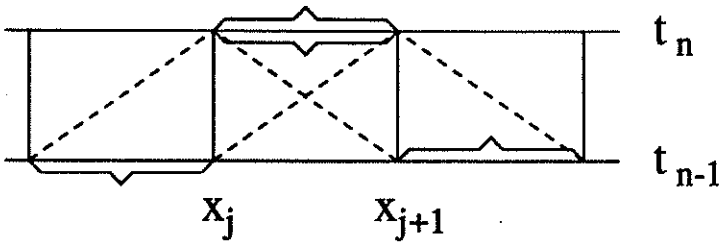
Figure 3.1



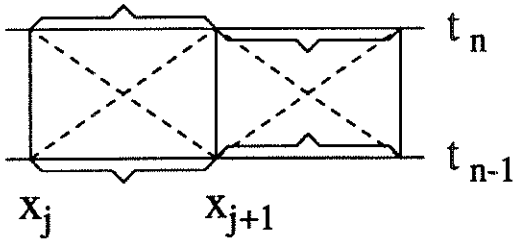
(a)



(b)

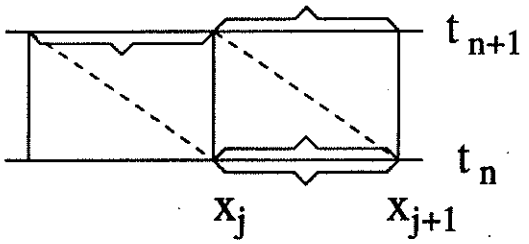


(c)

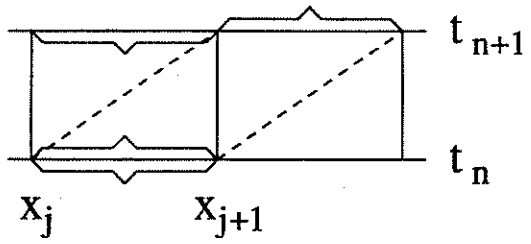


(d)

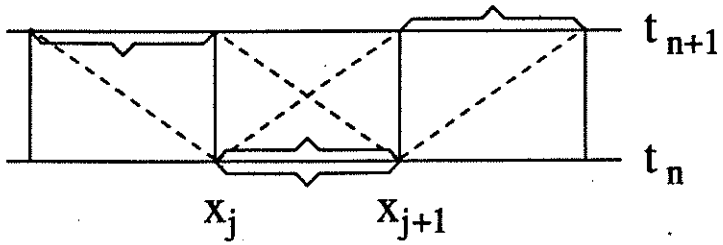
Figure 4.1



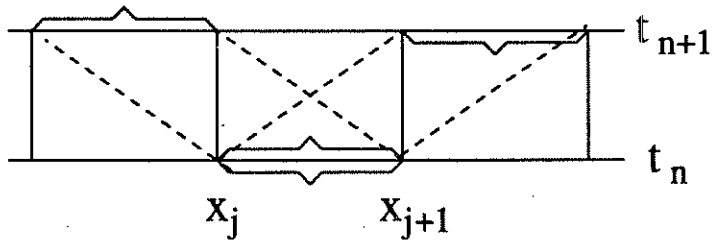
(a)



(b)



(c)



(d)

Figure 4.2

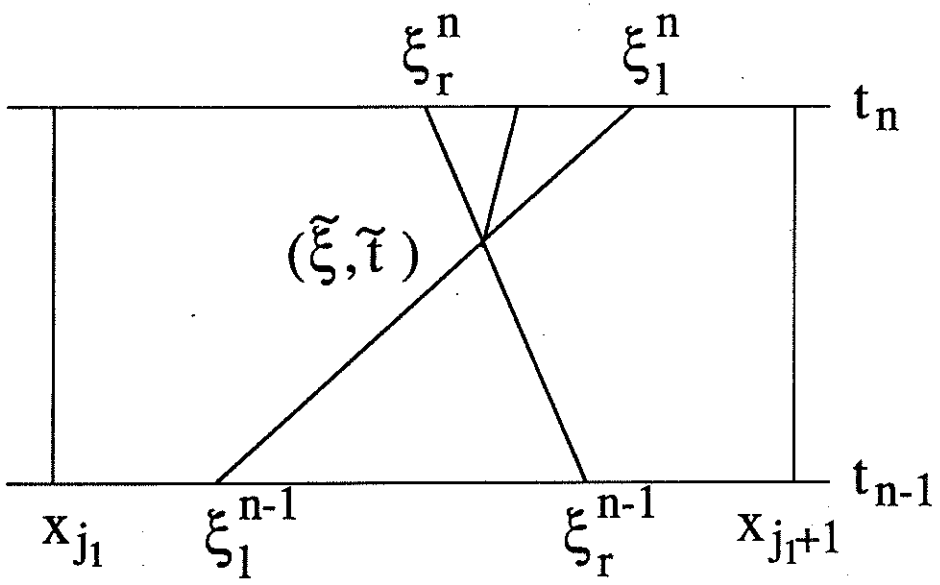


Figure 4.3

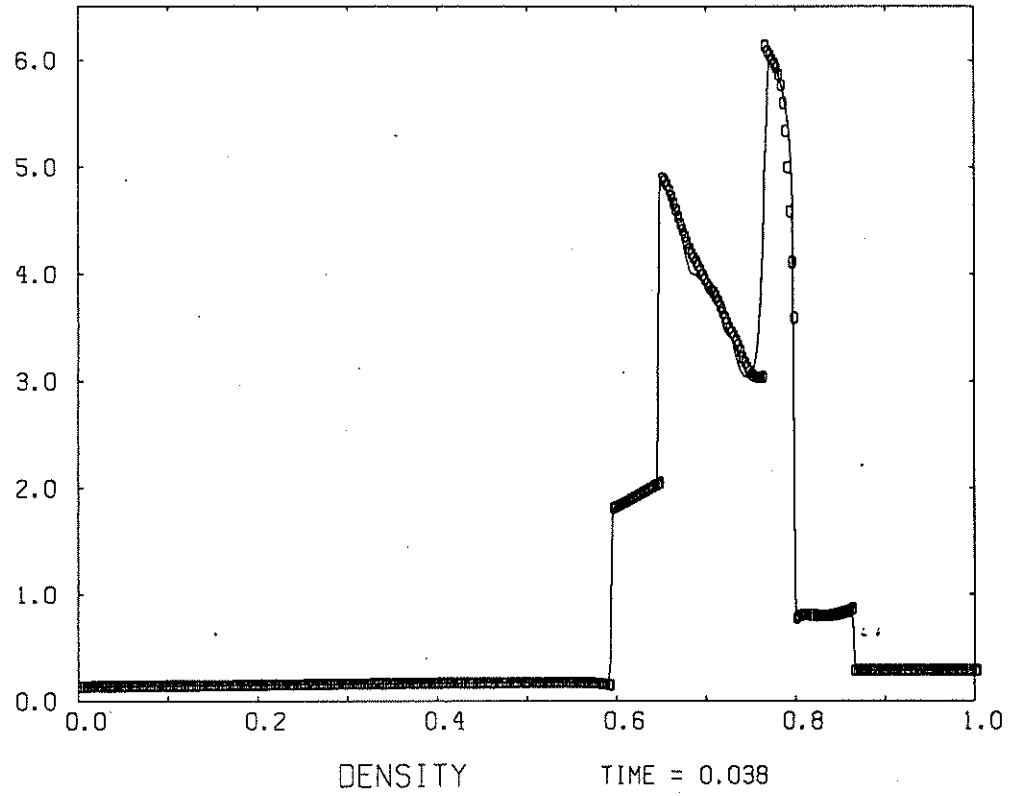


FIGURE 6.1-(a)

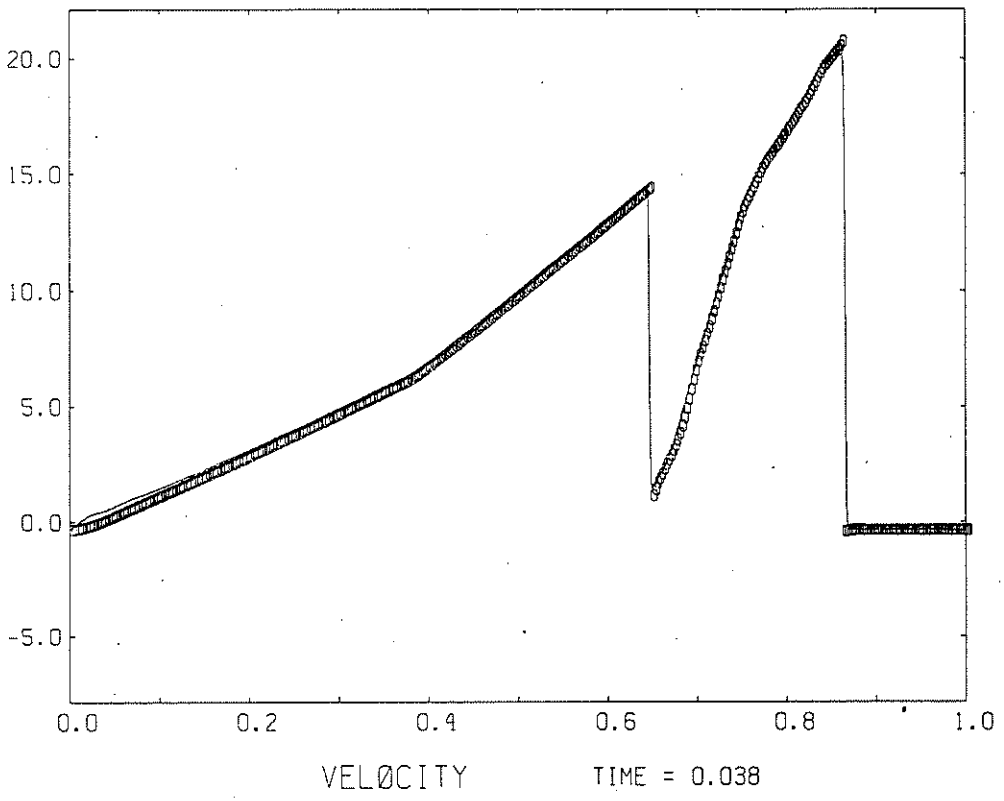


FIGURE 6.1-(b)

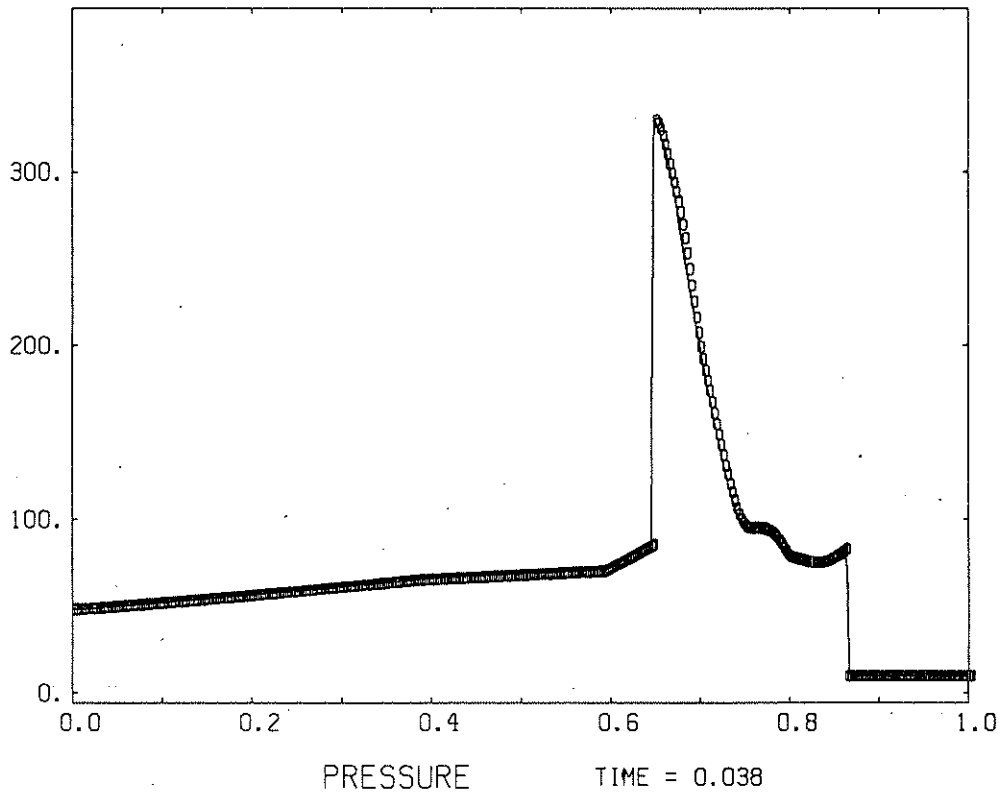


FIGURE 6.1-(c)

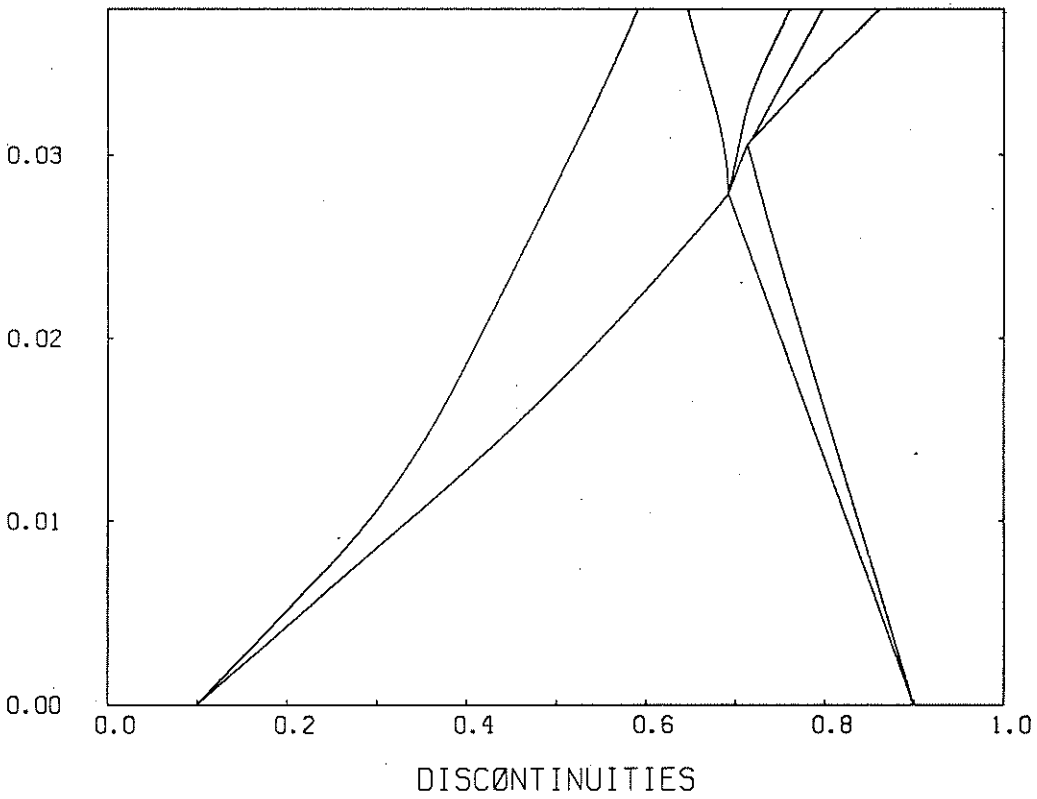


FIGURE 6.1-(d)

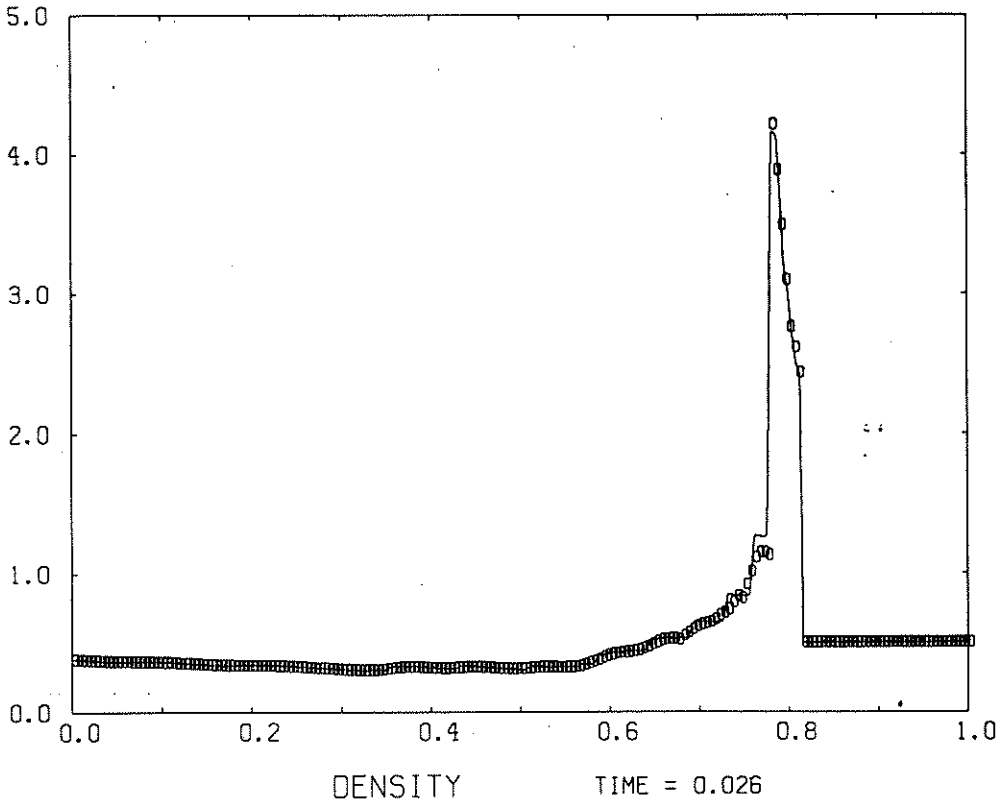


FIGURE 6.2-(a)

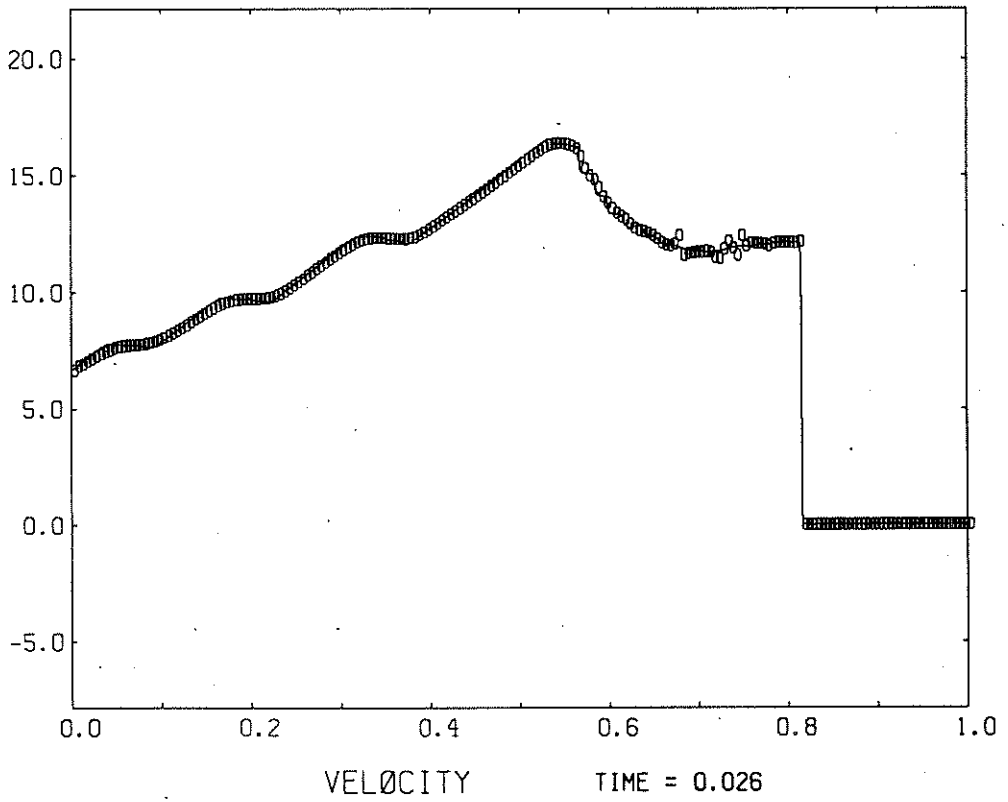


FIGURE 6.2-(b)

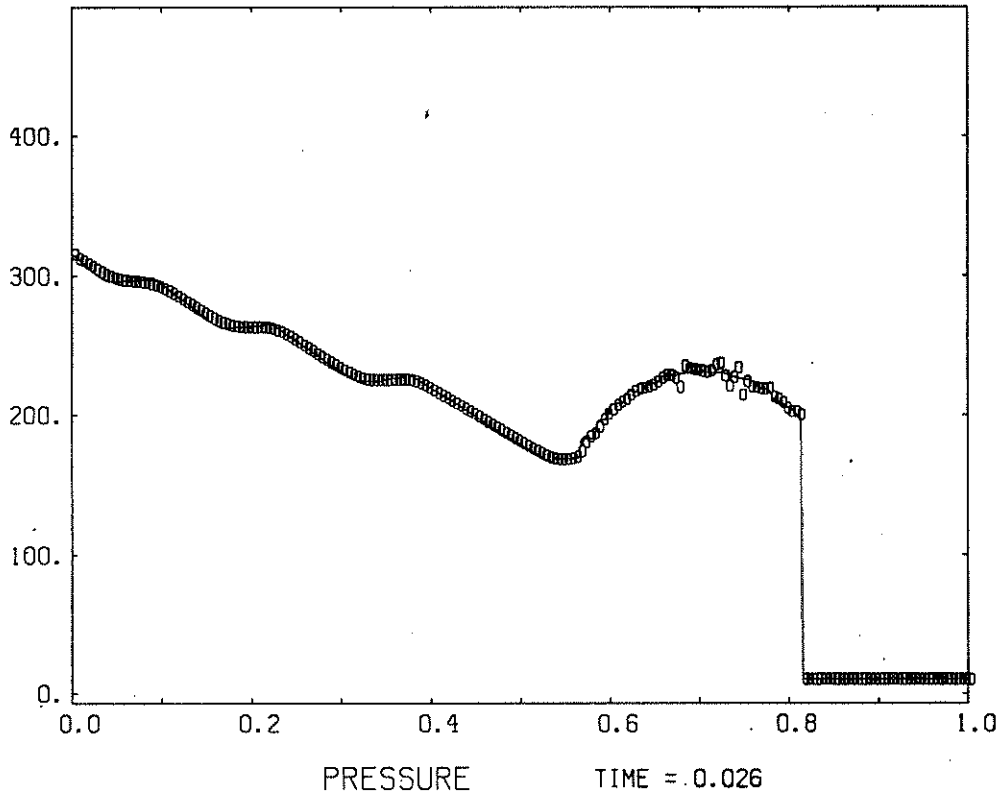


FIGURE 6.2-(c)

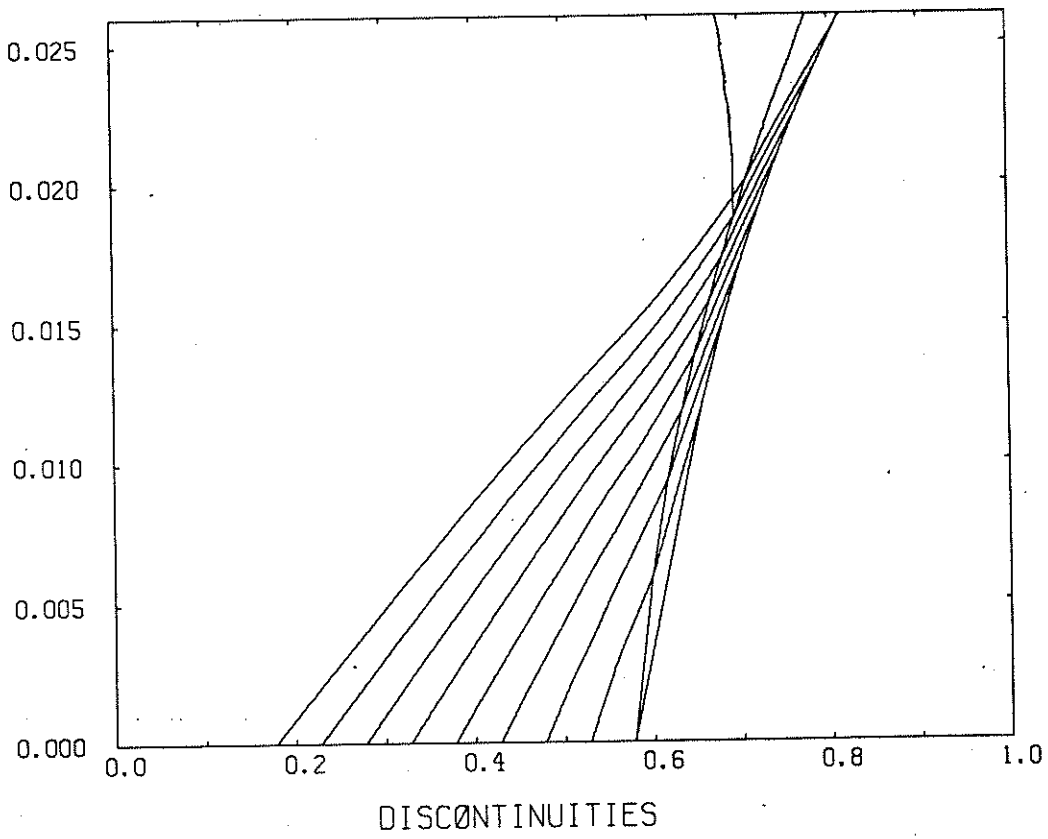

0.025
0.020
0.015
0.010
0.005
0.000
0.0
0.2
0.4
0.6
0.8
1.0
DISCØNTINUITIES

FIGURE 6.2-(d)

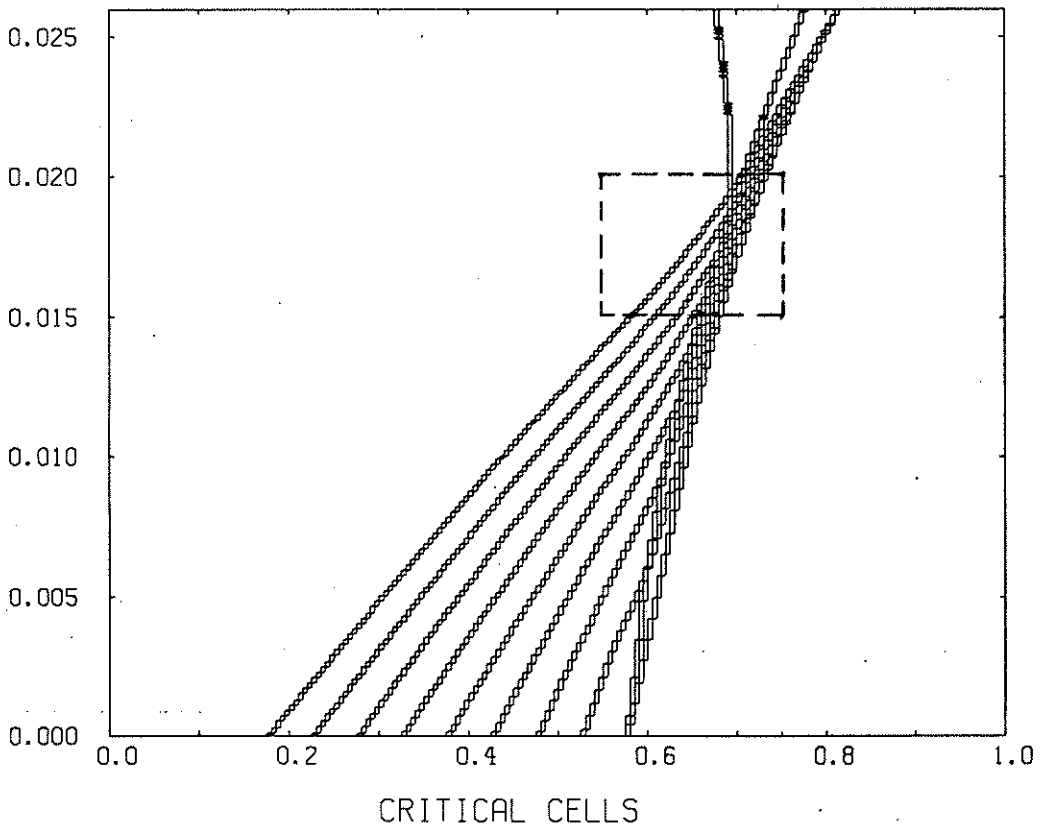


FIGURE 6.2-(e)

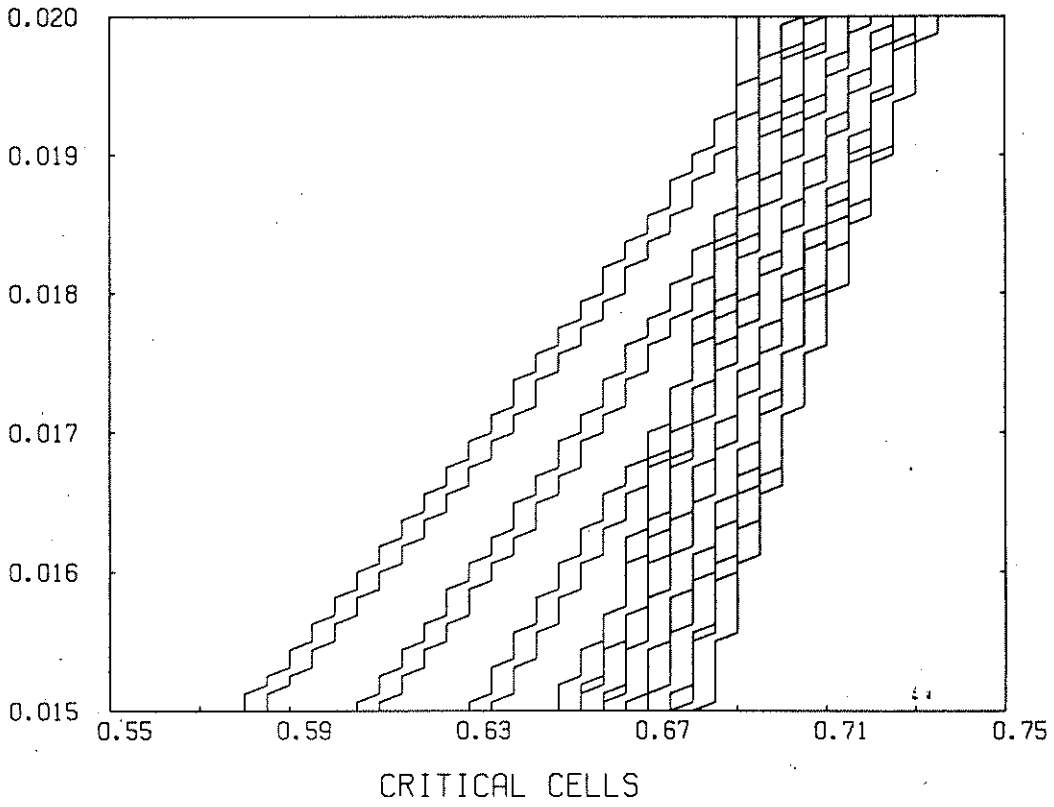

0.020
0.019
0.018
0.017
0.016
0.015
0.55
0.59
0.63
0.67
0.71
0.75
CRITICAL CELLS

FIGURE 6.2-(f)

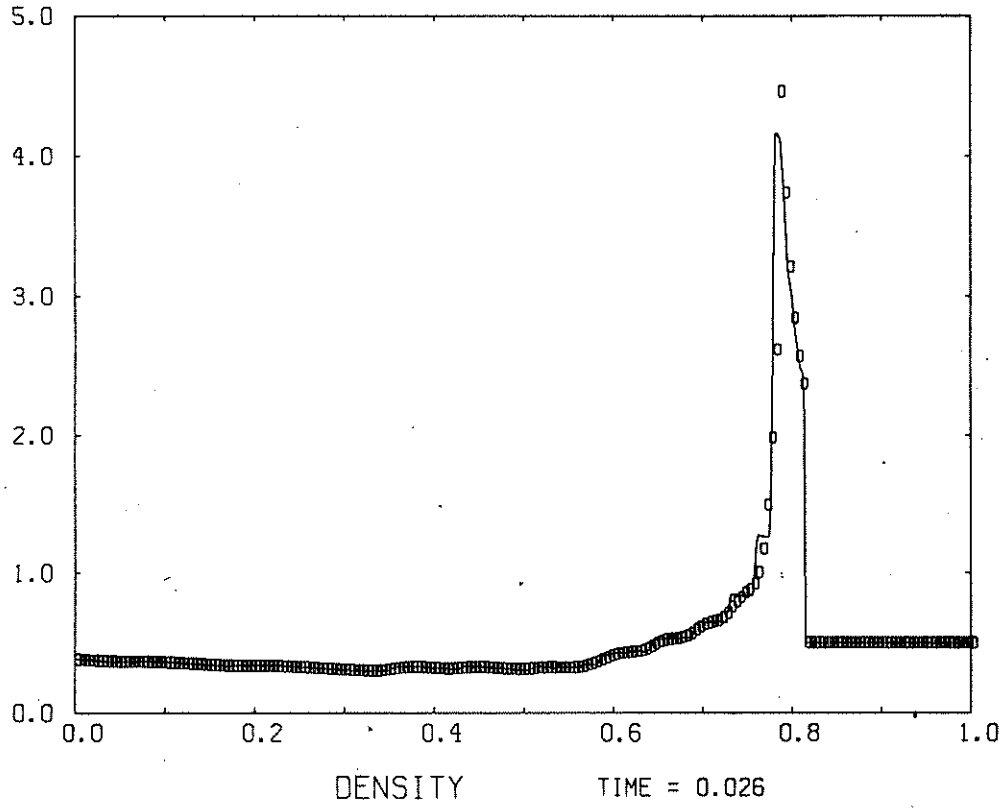


FIGURE 6.3-(a)

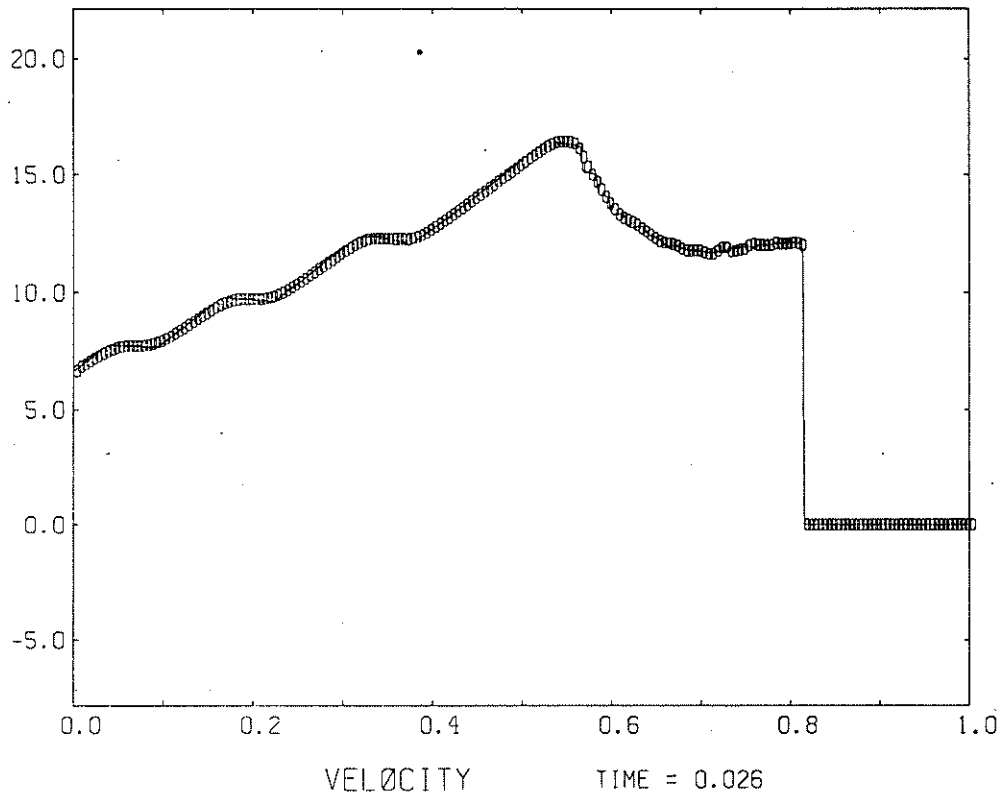


FIGURE 6.3-(b)

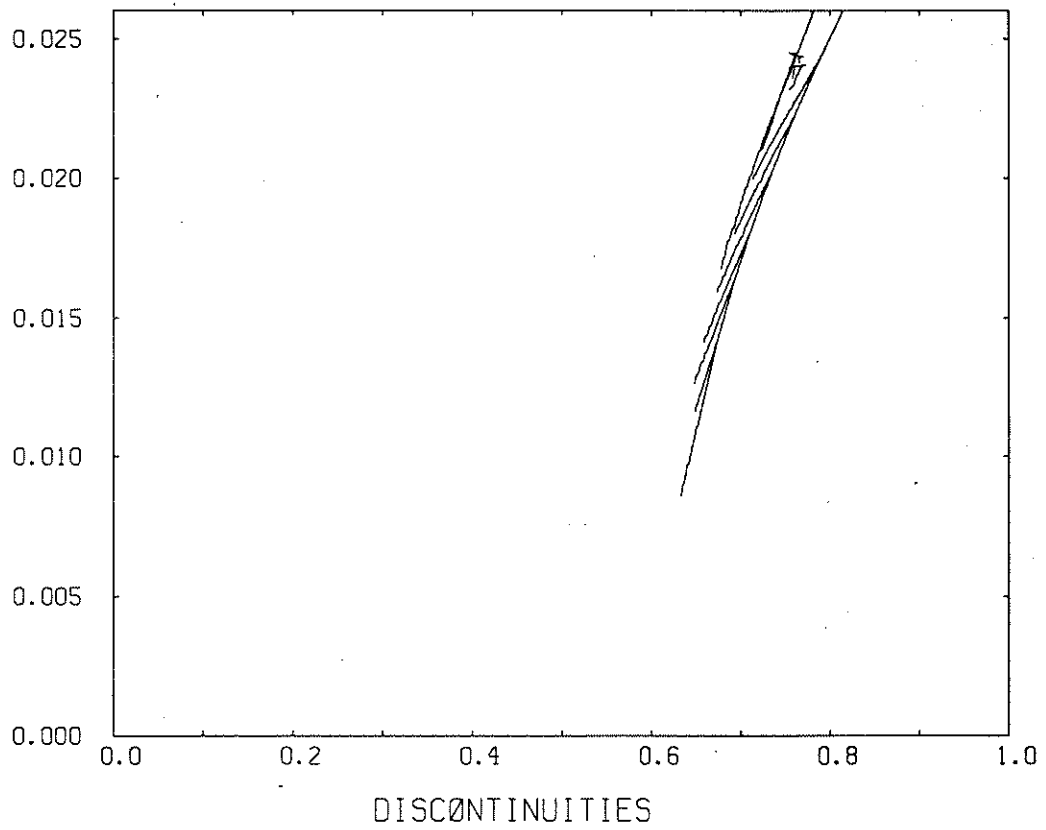

0.025
0.020
0.015
0.010
0.005
0.000
0.0
0.2
0.4
0.6
0.8
1.0
DISCØNTINUITIES

FIGURE 6.3-(c)

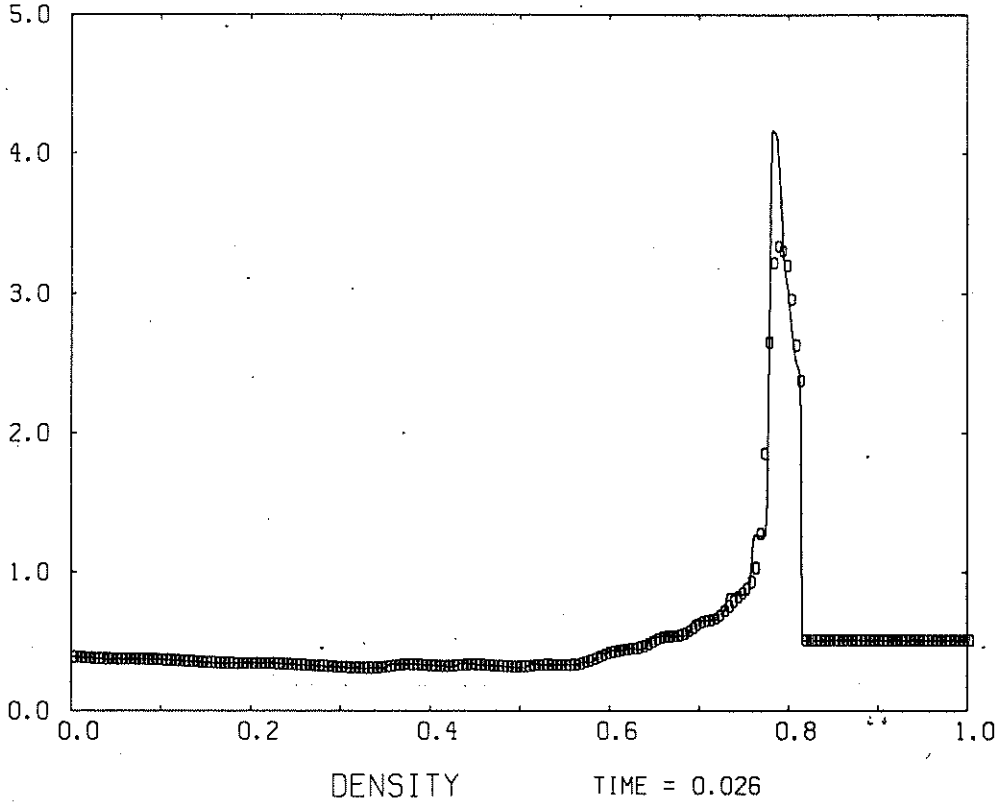


FIGURE 6.4-(a)

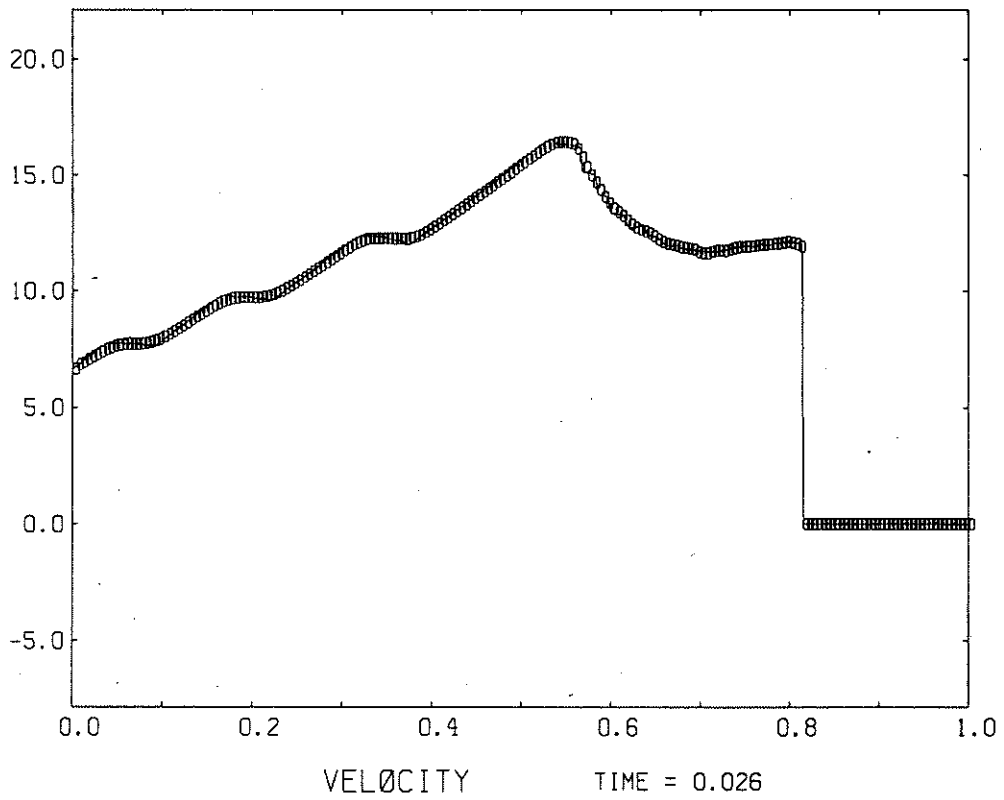


FIGURE 6.4-(b)

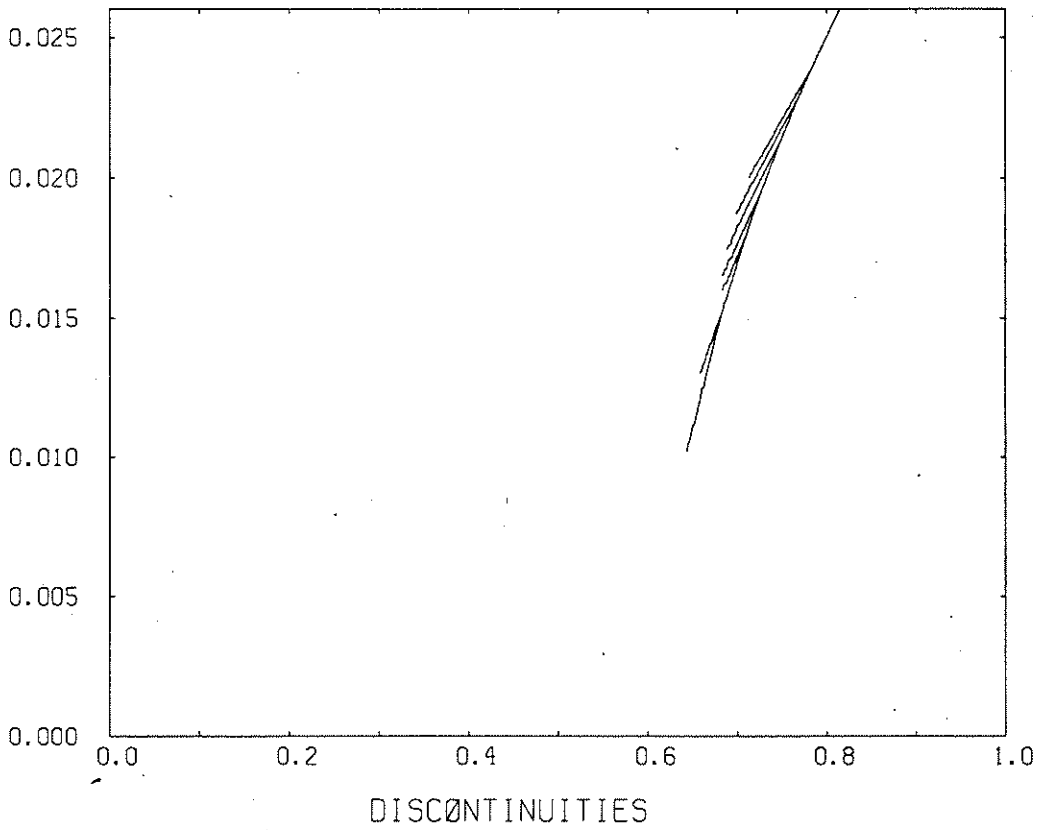

0.025
0.020
0.015
0.010
0.005
0.000
0.0
0.2
0.4
0.6
0.8
1.0
DISCØNTINUITIES

FIGURE 6.4-(c)

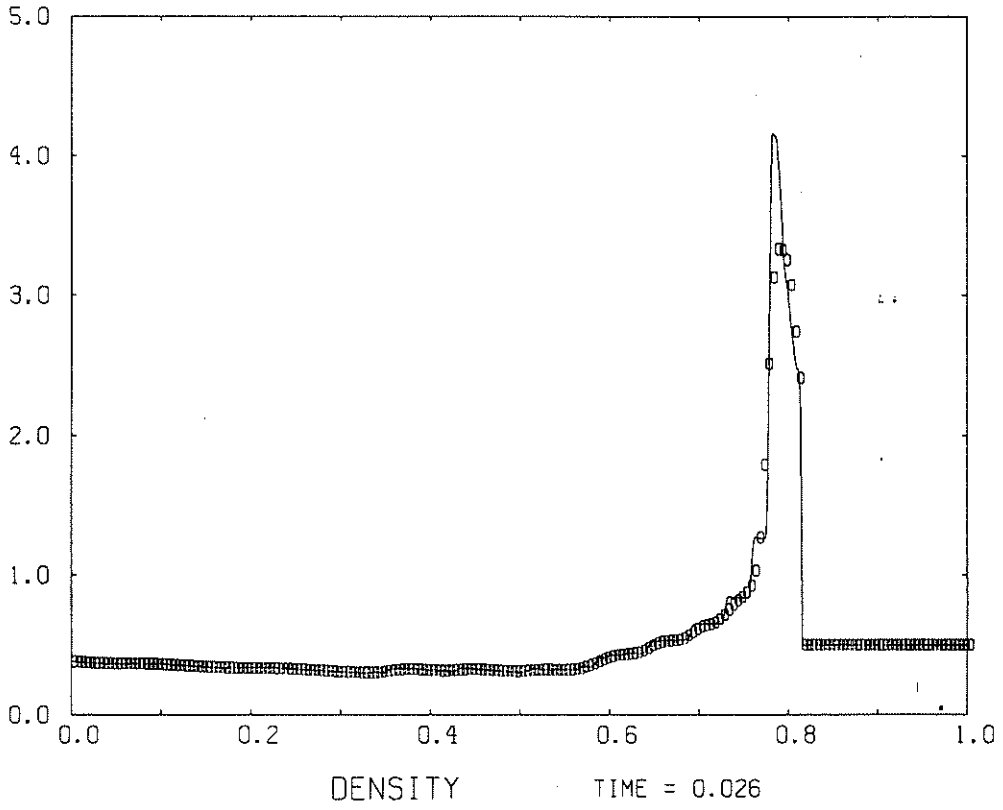


FIGURE 6.5-(a)

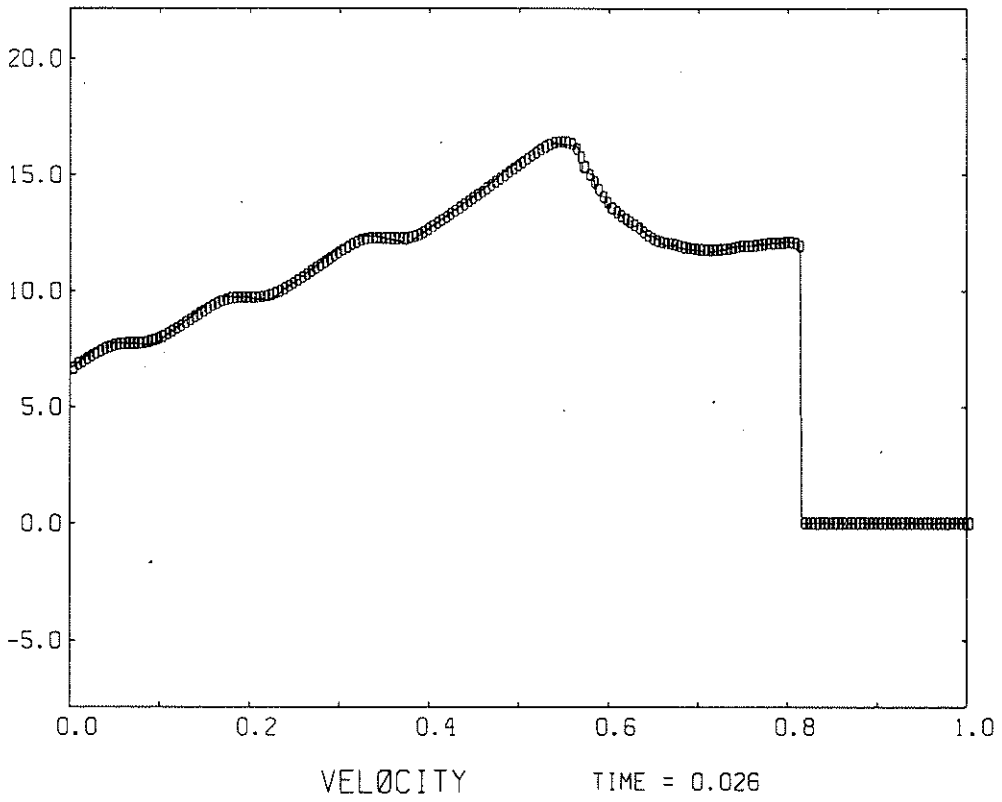


FIGURE 6.5-(b)

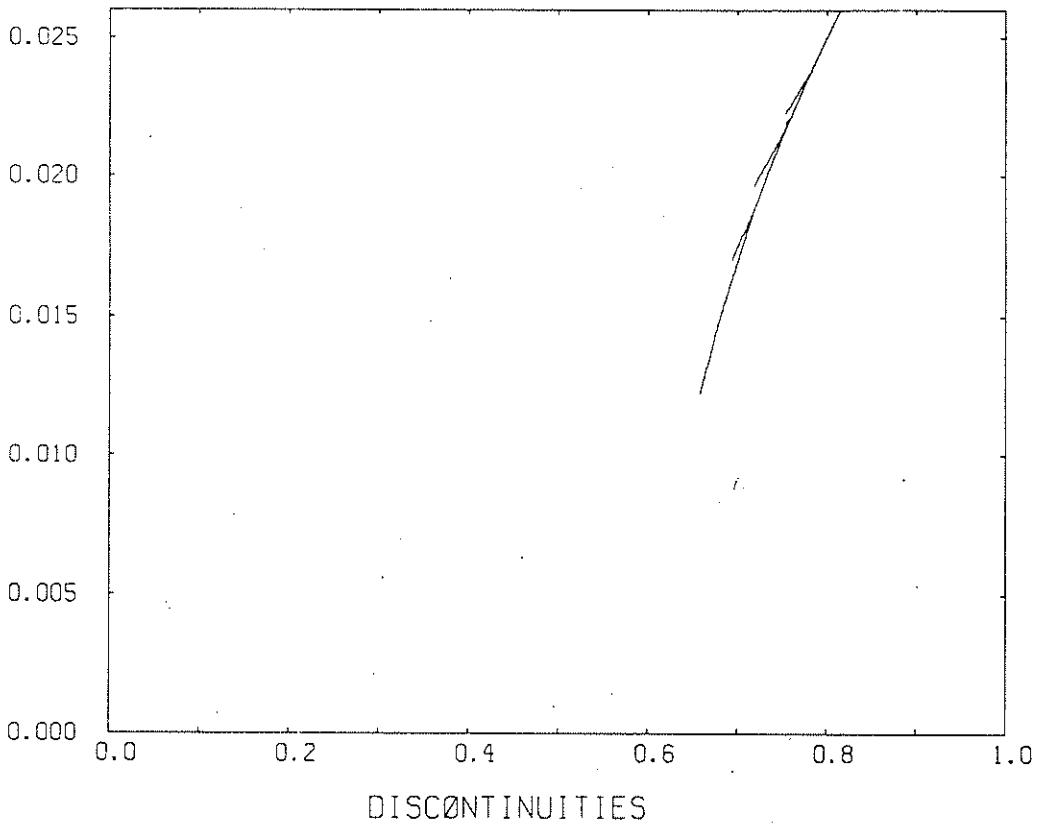

0.025
0.020
0.015
0.010
0.005
0.000
0.0
0.2
0.4
0.6
0.8
1.0
DISCØNTINUITIES

FIGURE 6.5-(c)