JOURNAL OF COMPUTATIONAL PHYSICS 101, 130–139 (1992)

A Front Tracking Method for Conservation Laws in One Dimension

N. H. RISEBRO

Department of Mathematics, University of Oslo, P.O. Box 1053, Blindern, N-0316 Oslo 3, Norway

AND

A. TVEITO

Department of Informatics, University of Oslo, P.O. Box 1080, Blindern, N-0316 Oslo 3, Norway

Received July 13, 1990; revised June 17, 1991

We present a front tracking technique for conservation laws in one dimension. The method is based on approximations to the solution of Riemann problems where the solution is represented by piecewise constant states separated by discontinuities. The discontinuities are tracked until they interact, at this point a new Riemann problem is solved and so on. No finite differences are used. This method is tested on the system of nonstationary gas dynamics defined by the Euler equations, and three test cases are presented.

1. INTRODUCTION

In this paper we present the implementation of a front tracking technique based on a generalization of Dafermos' [2] scheme for the scalar conservation law. We use the front tracking scheme to study the Euler equations of one-dimensional compressible gas dynamics.

The central idea of Dafermos was to approximate the flux function by a piecewise linear function. The advantage of this is that the solution of the Riemann problem in this case is piecewise constant, and one can therefore solve the Cauchy problem with piecewise constant initial data exactly for all time. Subsequent work on Dafermos' method with a scalar conservation law was done by LeVeque [12], Lucier [13], and Holden *et al.* [10]. There has been some work on generalizations of Dafermos' method to systems of conservation laws; Hedstrøm [8] did some numerical experiments on the p-system for a piecewise linear function $p(u)$, and Schwartz and Wendroff [16] used a hybrid method based on the piecewise constant solution to the Riemann problem.

J. Glimm, O. McBryan, and co-workers have for some time been using front tracking as a tool both to study gas-dynamics [4] and problems from reservoir simulation [5]. Their approach does, however, differ from ours in that they approximate the smooth parts of the flow by finite differences.

The basis of front tracking as we use it here, is the approximation of the solution of the Riemann problem by a step function in x/t without modifying the flux functions themselves. For a general initial value problem the initial function is approximated by a step function. Each "step" of this function then defines a Riemann problem which is solved and the solution approximated by a step function in x/t. This yields a set of discontinuities which propagate in the x–t plane until two neighboring discontinuities collide at some time $t_1 > 0$. At this time we solve the newly created Riemann problem determined by the constant states immediately to the left and right of the collision. The solution of this is in turn approximated by a step function, and the approximation can be continued up to the next collision.

It is clear that the approximation of the solution of the Riemann problem is central to this front tracking scheme. We use an approximation which is based on [14], where this front tracking technique was used to show the existence of a weak solution to the initial value problem under the same conditions as Glimm [3]. This approximation represents rarefraction waves by a series of "small steps" and keeps shocks and contact discontinuities at the correct position. The shocks and contact discontinuities are then assigned their correct speed, and the speed of the discontinuites that approximate a rarefraction wave are the characteristic speed on one side of the discontinuity. We stress that in this way no differences are taken, and the scheme contains no artificial viscosity.

The discontinuities in the approximated solution are kept in a list going from left to right, and in addition we keep track of the order of collisions of discontinuities in another list. Thus to implement the front tracking algorithm is more complicated than to implement most finite difference schemes. However, the tracking part is independent of the approximation of the solution of the Riemann problem, so

0021-9991/92 $5.00

once the front tracking machinery is made it can easily be adapted to handle other systems of equations.

In Section 2 we give a description of this front tracking method in general, and in Section 3 we explain how it is used to simulate the compressible nonstationary fluid flow governed by the Euler equations. In Section 4 of this study we present three test cases for the front tracking method: First a closed tube problem proposed by Woodward in [17, 18]; second, a problem where the front tracking method is compared with the Lax–Friedrichs method and with the random choice method. The second problem is chosen so that there will be many interactions of tracked discontinuities, and this seems to be a difficult problem for front tracking. The front tracking method shows great promise with respect to the study of asymptotic behavior for large time, therefore we include a third test example where the initial function is an "almost Riemann problem," i.e., a single smooth but steep transition between constant states.

2. THE FRONT TRACKING METHOD

The front tracking method we present here is a generalization of Dafermos' method for scalar conservation laws. In principle the method is applicable whenever a solution of the Riemann problem is computable, although we have no proof of convergence for arbitrary initial data. But, for strictly hyperbolic systems with sufficiently small total variation of the initial data, it was proved in [14] that the method produces a sub-sequence of approximate solutions which converges towards a weak solution of the problem. More precisely, [14] gives an alternative proof of Glimm's famous existence theorem [3]. The purpose of the present paper is to explore the method as a computational tool.

Consider the following system of conservation laws

$$u_t + f(u)_x = 0, \tag{2.1}$$

where $u = u(x, t) \in \mathbb{R}^n$ and $f: \mathbb{R}^n \to \mathbb{R}^n$ is a smooth function. The Riemann problem for (2.1) is the initial value problem with data of the form

$$u(x, 0) = \begin{cases} u_l & \text{if } x < 0 \\ u_r & \text{if } x > 0. \end{cases} \tag{2.2}$$

If $u(x, t)$ is a solution of (2.1), (2.2) then so is $u(cx, ct)$ for all constants $c > 0$; therefore to be unique the solution of a Riemann problem has to be self-similar, i.e., $u(x, t) = \bar{u}(x/t)$. The solution of the Riemann problem is composed of elementary waves, i.e., rarefraction waves and shock waves (including contact discontinuities), and constant states.

Rarefraction waves are classical solutions of the system (2.1). For such solutions (2.1) can be rewritten as

$$u_t + df(u)\, u_x = 0, \tag{2.3}$$

where df is the Jacobian matrix of f. For self-similar solutions we obtain

$$(df - (x/t)\, I)\, \bar{u}(x/t) = 0. \tag{2.4}$$

Hence x/t is an eigenvalue of df and $\bar{u}(x/t)$ is the corresponding eigenvector. Let λ be an eigenvalue of df and let r be the corresponding right eigenvector. If $\bar{u}$ is an integral curve of r, i.e.,

$$\bar{u}'(\xi) = r(\bar{u}(\xi)), \tag{2.5}$$

and $\bar{u}$ connects the states u_l and u_r and if λ increases as $\bar{u}$ goes from u_l to u_r, then the path $\bar{u}$ traverses is called a rarefunction curve. Rarefraction curves are directed towards increasing values of λ, and the function

$$u(x, t) = \begin{cases} u_l & \text{for } x/t < \lambda(u_l) \\ \bar{u}(x/t) & \text{for } x/t = \lambda(\bar{u}) \\ u_r & \text{for } x/t > \lambda(u_r) \end{cases} \tag{2.6}$$

is called a simple rarefraction solution of (2.1). The speed of a rarefraction wave is given by $\lambda(\bar{u})$; in particular the initial speed is $\lambda(u_l)$ and the final speed is $\lambda(u_r)$.

Shock waves are weak solutions of (2.1) of the form

$$u(x, t) = \begin{cases} u_l & \text{for } x/t < \sigma \\ u_r & \text{for } x/t > \sigma, \end{cases} \tag{2.7}$$

where σ is the shock speed. In order to be a weak solution, σ has to satisfy the Rankine–Hugoniot relation

$$\sigma(u_l - u_r) = f(u_l) - f(u_r). \tag{2.8}$$

A weak solution is not necessarily unique, therefore an additional entropy condition is imposed on (2.8) in order to obtain uniqueness.

The solution of a Riemann problem consists of a sequence of elementary waves connecting the left state u_l with the right state u_r. We introduce the following notation: $u_1 \xrightarrow{w} u_2$ means that the state u_1 can be connected to u_2 by an elementary wave w. Let $u_1 \xrightarrow{w_1} u_2 \xrightarrow{w_2} u_3$ be two elementary waves, and let v_i^{I} and v_i^{F}, $i = 1, 2$, be the initial and final speed of each wave. If w_i is a shock wave then $v_i^{\mathrm{I}} = v_i^{\mathrm{F}} = \sigma$. The two waves are said to be compatible if

$$v_1^{\mathrm{F}} \leqslant v_2^{\mathrm{I}}. \tag{2.9}$$

If both waves are shocks we impose strict inequality in (2.9). The notation $u_1 \xrightarrow{w_1} u_2 \xrightarrow{w_2} u_3$ means that w_1 and w_2 are compatible, and that u_1 can be connected to u_3 through the

intermediate state u_2. A solution of a Riemann problem thus consists of a sequence of compatible waves

$$u_l = u_0 \xrightarrow{w_1} u_1 \xrightarrow{w_2} u_2 \cdots \xrightarrow{w_N} u_r. \tag{2.10}$$

Such a solution can be computed explicitly for many systems of equations, both for strictly hyperbolic systems [7, 15] and for systems of non-strictly hyperbolic [11] or mixed type [9].

Assume that the solution of the Riemann problem (2.2) for the system (2.1) can be represented in the form (2.10). Let δ be some (small) positive real number. We shall construct a piecewise constant approximation $u_\delta(\xi)$ to the solution, represented by the sequences $\{u_{\delta,j}\}$, $\{\xi_{\delta,j}\}$ in the following manner:

$$u_\delta(\xi) = u_{\delta,i} \quad \text{for} \quad \xi_{\delta,i} < \xi < \xi_{\delta,i+1}. \tag{2.11}$$

Let $\{u_i, w_i, v_i^{\mathrm{I}}, v_i^{\mathrm{F}}\}_{i=0}^{N}$ denote the solution of the Riemann problem given by (2.10). We define $\{u_{\delta,j}\}$ and $\xi_{\delta,j}$ inductively as

$$u_{\delta,0} = u_0 = u_l, \qquad \xi_{\delta,0} = -\infty. \tag{2.12}$$

Assume we have defined the sequences $\{u_{\delta,j}\}_{j=0}^{k}$, and $\{\xi_{\delta,j}\}_{j=0}^{k}$ and that $u_{\delta,k} = u_i$ for some i. The approximation of the wave $u_i \xrightarrow{w_{i+1}} u_{i+1}$ can be divided into two cases:

(1) w_i is a shock wave. In this case $\xi_{\delta,k+1} = \sigma(w_i)$ and $u_{\delta,k+1} = u_{i+1}$.

(2) w_i is a rarefaction wave. Let

$$M = \left\lceil \frac{\lambda(u_{i+1}) - \lambda(u_i)}{\delta} \right\rceil + 1$$
$$\delta' = \frac{\lambda(u_{i+1}) - \lambda(u_i)}{M}, \tag{2.13}$$

where $\lceil y \rceil$ is the largest integer smaller than or equal to y, and let $\bar{u}$ be the rarefraction curve defined by (2.5). We can assume $\bar{u}$ to be parameterized by λ. Now let

$$u_{\delta,k+l} = \bar{u}(\lambda(u_i) + l\delta') \tag{2.14}$$

and

$$\xi_{\delta,k+l} = \lambda(u_{\delta,k+l}) \tag{2.15}$$

for $l = 1, \ldots, M$, cf. Figs. 2.1 and 2.2.

We now say that (2.11) is a *δ-approximation* to the solution of the Riemann problem. This approximation has the properties that

$$\{u_i\} \subset \{u_{\delta,j}\} \tag{2.16}$$

$$\sup_{\xi} |u_\delta(\xi) - u(\xi)| = O(\delta). \tag{2.17}$$

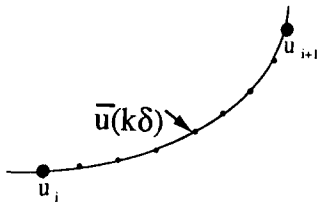

FIG. 2.1. The approximation of a rarefraction wave in state space.

Observe that by (2.16) all the intermediate states in the solution of the Riemann problem are kept in the approximation. This is due to the separate treatment of each elementary wave. We believe that this property of the approximation is crucial in order for it to be able to reflect the complexity of the solution. Equation (2.17) holds, since u and u_δ will be equal except in a rarefaction fan; here $u_\delta(\xi) = u_{\delta,i}$ for some i and $|u_\delta - u| \leqslant |u_{\delta,i} - u_{\delta,i+1}| = O(\delta)$.

This δ-approximation defines a number of *fronts*. A front is an object with the following attributes

$$\textbf{front}: \{u_l, s, (x_0, t_0), (x_1, t_1), \text{family, pointers}\}.$$

Here u_l is the state to the left of the discontinuity, s is the speed of the discontinuity, (x_0, t_0) is the point in x, t space, where the discontinuity originates, and (x_1, t_1) is the point in which it terminates. A front is terminated when it collides with one of its neighboring fronts. The family of a front indicates the wave type it approximates. For strictly hyperbolic systems it is defined to be i if $v_{i-1}^{\mathrm{F}} \leqslant s < v_{i+1}^{\mathrm{I}}$. The pointers are used to store the fronts in a suitable data structure. The fronts are organized from left to right in a so-called x-list. Each front has a pointer to its left and right neighbors. This structure makes it easy to remove and add fronts. We also have a t-list. This list organizes the fronts with respect to the collision times t_1. The first front in the list has the smallest value of t_1. This front has a pointer to the front with the second smallest t_1 and so on.

We are interested in the Cauchy problem for (2.1) with

$$u(x, 0) = u_0(x). \tag{2.18}$$

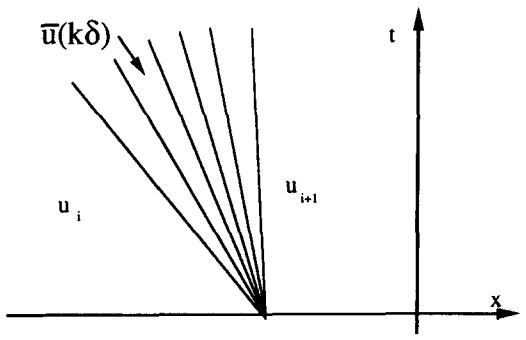

FIG. 2.2. The approximation of a rarefraction wave in x–t space.

If u_0 is a piecewise constant function with a finite number of discontinuities, we can construct a δ-approximation to the solution of each of the Riemann problems defined at the discontinuities of u_0. We now have a system of fronts, each of which can be propagated independently until one of them interacts (collides) with one (or more) of its neighbors. Then we solve the Riemann problem defined by the left state of the left colliding front and the right state of the right colliding front. We make a δ-approximation to this solution, and again we have a system of fronts which can be tracked until the next collision. In this way the solution can be advanced in time, see Fig. 2.3. Now we can state the algorithm used to compute our numerical approximation:

Generate step function approximation to initial function
Generate initial fronts
Time := 0.0
do while *Time* < *TotalTime*
  CollisionTime := *FirstCollisionTime*
  if *CollisionTime* < *TotalTime* **then**
    Solve Riemann problem at CollisionPoint
    Update the CollisionTimes
    Update the CollisionList
  else
    Advance all fronts to TotalTime
  endif
  Time := *CollisionTime*
enddo

This front tracking method is fast compared to fixed grid methods, since it automatically will focus computational effort where interactions are occurring, i.e., where the solution has a complex behavior. This effect is especially apparent when few interactions are taking place as we show in Example 3 in Section 4.

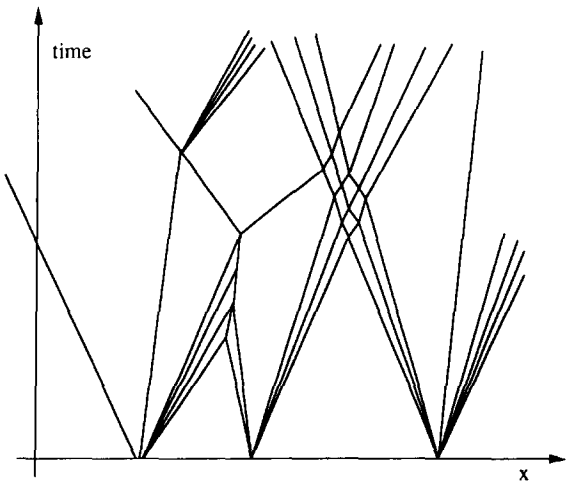

FIG. 2.3. A system of discontinuities in x–t space.

Remark. When updating the solution in this way, we in general have no theoretical limit on the number of fronts that may arise. Therefore we must have some mechanism which limits the growth of the number of fronts. In order to do this, we have chosen to ignore small fronts in the δ-approximation to the solution of the Riemann problem. We ignore fronts which represent a discontinuity of magnitude less than some $c(\delta)$, where $c \to 0$ as $\delta \to 0$. In general we have no justification for ignoring such fronts, but for the system of gas dynamics the numerical experiments showed that this resulted in an accurate and efficient method.

3. APPLICATION TO GAS DYNAMICS

We consider the equations of one-dimensional gas dynamics in Eulerian coordinates:

$$\begin{aligned} \rho_t + (\rho u)_x &= 0 \\ (\rho u)_t + (p + \rho u^2)_x &= 0 \\ [\rho(\tfrac{1}{2}u^2 + i)]_t + [\rho u(\tfrac{1}{2}u^2 + i) + pu]_x &= 0. \end{aligned} \tag{3.1}$$

Here p is the pressure, u is velocity, ρ is the density, and i is the internal energy. We assume the gas to be polytropic so $i = c_v T$, where c_v is a positive constant and T is the temperature. We also assume the gas to be ideal so that $p = R\rho T$, where R is a positive constant. Finally, we assume that some γ-gas law holds so that $p = k \exp(S/c_v)\, \rho^\gamma$, where S is the entropy and k is a positive constant. Since a solution to (3.1) may contain discontinuities, we interpret (3.1) in the sense of distributions.

An integral component of the front tracking method is the solution of the Riemann problem. For the system of gas-dynamics, this solution is well known and has been discussed extensively, see, e.g., [14 and the references therein; 7], where detailed instructions for constructing a solution are given, as well as a comparison between several solution methods. Therefore we only outline the structure of the solution. The Riemann problem is the special initial value problem for (3.1), where the initial functions take the form:

$$(p, u, \rho)(x, 0) = \begin{cases} (p_l, u_l, \rho_l) & \text{for } x < 0 \\ (p_r, u_r, \rho_r) & \text{for } x > 0. \end{cases} \tag{3.2}$$

There are three elementary waves in the solution of the Riemann problem: shocks, rarefaction waves, and contact discontinuities. The solution generally contains a leftward moving shock or rarefaction wave, a contact discontinuity, and a rightward moving shock or rarefaction. In fact, given (p_l, u_l, ρ_l) and (p_r, u_r, ρ_r) we can construct two curves in the (p, u) plane, u_1 and u_2, passing through (p_l, u_l), which have the property that if (p_r, u_r) is on the curve u_1, the

Riemann problem is solved by a leftward moving wave and possibly a contact discontinuity. If (p_r, u_r) is on the curve u_2 the Riemann problem is solved by a rightward moving wave and possibly a contact discontinuity. Furthermore, if $u_l > u_r$ the wave is a shock wave, and if $u_l < u_r$ the wave is a rarefaction wave. These curves divide the (p, u) plane into four regions, and the pattern of the solution (e.g., shock–contact–rarefaction) only depends upon which region (p_r, u_r) is in. Similarly there are curves $\tilde{u}_1$ and $\tilde{u}_2$ through (p_r, u_r). The Riemann problem is solved by finding the unique point (p_m, u_m), where $\tilde{u}_2$ intersects u_1. The pressure and velocity do not change across the contact discontinuity and the density can be determined when (p_m, u_m) is known. Let ρ_{ml} and ρ_{mr} be the densities to the left and right of the contact discontinuity, respectively. If the backward moving wave is a shock (rarefaction) we say that (p_l, u_l, ρ_l) is connected to (p_m, u_m, ρ_{ml}) by a shock (rarefaction) and similarly for the forward moving wave. For our purposes we can regard the solution of the Riemann problem to be known with infinite accuracy, although the determination of the middle state usually requires some iterative procedure.

Thus given the solution of the Riemann problem we will, following the general algorithm outlined above, make an approximation to use as a building block in our numerical algorithm. Let δ be some small positive number. If (p_i, u_i, ρ_i) is connected to (p_m, u_m, ρ_{mi}) by a shock, we leave the discontinuity as it is. If the states are connected by a rarefaction wave we generate constant states on the curve u_i (or $\tilde{u}_i$) in (p, u, ρ) space as in the previous section, cf. (2.13), (2.14), and (2.15). These constant states are generated as follows. First assume that the state to the left is (p_l, u_l, ρ_l); this will be our first constant state. We then follow the curve u_1 towards (p_m, u_m, ρ_{ml}), a distance of δ. This point on the curve will be our next constant state. The speed of the discontinuity between these two constant states will be the characteristic speed to the left of the discontinuity: $u - c$, where c is the *sound speed* given by $c^2 = \gamma p/\rho$. We continue with steps of size δ on the curve u_1 until we are closer (measured in arc-length along the curve) to (p_m, u_m, ρ_{ml}) than δ; (p_m, u_m, ρ_{ml}) is then the last constant state. If the state on the left is (p_m, u_m, ρ_{mr}) the wave is a forward moving rarefaction wave. In this case we start from (p_r, u_r, ρ_r) and choose our constant states on $\tilde{u}_2$. Otherwise the procedure is symmetrical. We leave the contact discontinuity in the middle as it is. This approximation we call a δ-approximation to the solution of the Riemann problem.

As in the previous section, a δ-approximation consists of a number of *fronts*. Each front has a state; (p, u, ρ) which is the constant state immediately to the left of it, a speed, and a family. The family of a front is 1 if the left and right states of the front lie on a u_1 curve; similarly the family is 2 if the states lie on a $\tilde{u}_2$ curve. The family of a contact discontinuity is 0.

Our strategy is now to approximate the initial data as piecewise constant, solve the Riemann problems at $t = 0$, then make δ-approximations to the solutions of the initial Riemann problems, and finally to track each front until it interacts with one of its neighboring fronts. Here we can solve a new Riemann problem with states defined by the states of the colliding fronts, make a δ-approximation to this solution, and track the fronts until the next collision. In this way we update the solution in time. In order to do this we equip our fronts with some more structure. A front has a starting point (x, t), as well as a collision time t_1 which is the time it will collide with one of its neighbors. If the front will not collide this parameter is set to be ∞. To order these collision times each front also has a pointer to the front which has the first collision time after the fronts own collision time. Finally each front has a strength which is the distance along the u_i curve between the state to the left of the front and the state of the right of the front.

The step function approximation to the initial data is also related to δ. We assume our initial data to be piecewise continuous and constant outside some bounded interval. Let $U = (p, u, \rho)$, and let the initial function take the value U_l to the left of this interval and the value U_r to the right of it. Fix $x_0 = -\infty$, $U_0 = U_l$, and let the sequences $\{U_i\}_{i=1}^N$ and $\{x_i\}_{i=1}^N$ be defined by

$$x_i = \inf_{x > x_{i-1}} \{|U_0(x) - U_{i-1}| \geq \delta\}, \qquad U_i = U_0(x_i). \tag{3.3}$$

Redefine $U_N = U_r$. A collision is said to have a strength equal to the product of the strengths of the colliding fronts. When creating a δ-approximation to the solution of a Riemann problem after a collision of fronts, we will not create new frons of families different from the families of those that collide if the collision strength is less than the minimum of $(\delta/2, \delta^2)$.

Most of the CPU time (more than 95%) used by the front tracking method is spent solving Riemann problems. It is therefore important to have fast Riemann solvers available. We have used the Riemann solver described and recommended by Gottlieb and Groth in [7].

4. EXAMPLES

In this section we present three test examples where the front tracking method has been used. We want to explore some of the properties of the front tracking method, such as the convergence rate and conservation of mass end energy, and to compare it with other numerical methods.

In case of a strictly hyperbolic conservation law with initial data of small total variation, the front tracking method produces a sequence which converges in L_1-norm [14]. Hence we use the relative difference in L_1 between a

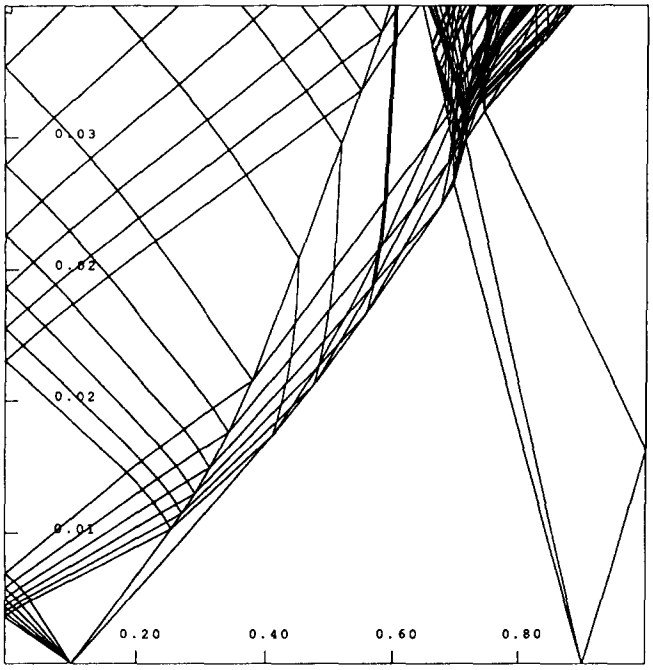

FIG. 4.1. Problem #1. All fronts from $t=0$ to $t=0.038$ for the approximation using $\delta = 100.0$.

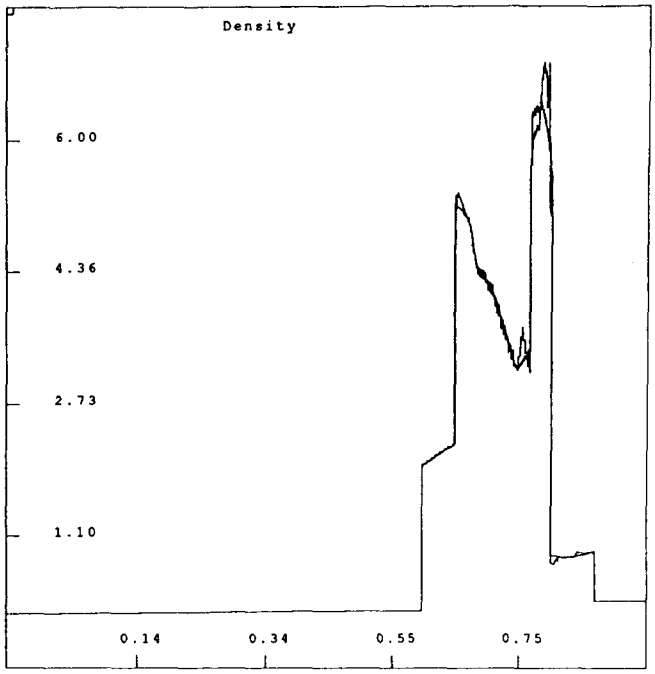

FIG. 4.2. Problem #1. The reference solution and the approximation with $\delta = 20.0$ at $t = 0.038$.

reference solution and the approximate solution as measure of the error. Let $\|f\|_1 = \int |f|\, dx$ and define the relative error

$$\varepsilon = \frac{\|p_\delta - p\|_1}{\|p\|_1} + \frac{\|u_\delta - u\|_1}{\|u\|_1} + \frac{\|c_\delta - c\|_1}{\|c\|_1}. \tag{4.1}$$

Here c is the sound speed. The quantities with subscript δ are the approximate solutions.

EXAMPLE 1. This example is taken from Woodward and Colella [17, 18] and seems to be a standard test problem in numerical gas dynamics. The initial condition consists of three constant states of a γ gas law, $\gamma = 1.4$. The gas is initially at rest between closed walls at $x=0$ and $x=1$. The density is everywhere unity, but the pressure varies:

$$p(0) = \begin{cases} 1000.0 & \text{for } x < 0.1 \\ 0.01 & \text{for } 0.1 < x < 0.9 \\ 100.0 & \text{for } x > 0.9. \end{cases} \tag{4.2}$$

Initially two shock waves develop and interact, while two rarefaction develop and are reflected off the walls at each end of the tube. These reflections again interact with each of the shock waves and a very complicated pattern is quickly established. (See also [18] for a qualitative description of the solution.) In Fig. 4.1 we see all fronts for an approximation using $\delta = 100.0$ from $t=0.0$ to $t=0.038$.

The reference solution to this problem used $\delta = 0.8$. At $t=0.038$ approximately 5,000,000 collisions of fronts have occurred. The reference solution seems to agree well with the solution reported in [18]; both the peaks and the position of the discontinuities all agree. In Fig. 4.2 we see the reference solution and an approximation using $\delta = 20.0$. Note that the positions of all major shocks are virtually identical in the two solutions. In Table I we show δ, the L_1 error, the error in mass conservation, the error in energy conservation, and the CPU time used by the front tracker

TABLE I

Front Tracking

δ	CPU	Error	Mass err %	Energy err %
50.0	5.2	0.165	0.71	0.92
40.0	10.8	0.098	0.51	0.60
30.0	17.8	0.079	0.38	0.47
25.0	26.4	0.069	0.39	0.41
20.0	37.0	0.056	0.30	0.29
15.0	86.7	0.037	0.25	0.30
12.5	134.8	0.030	0.24	0.25
10.0	193.7	0.026	0.19	0.23
7.5	391.1	0.020	0.16	0.20
7.0	399.9	0.018	0.14	0.19
6.0	690.3	0.017	0.12	0.14
5.0	810.3	0.014	0.09	0.11

Note. Based on a standard regression analysis we compute: $\varepsilon = O(\delta^{1.02})$, standard error estimate for the exponent 0.03; CPU time $= O(\varepsilon^{-2.1})$, standard error estimate for the exponent 0.06.

for this problem. Note that although the front tracker is not conservative, the errors in the mass balance and the energy balance remain small, and correlates well with the L_1 error. Based on standard regression analysis we have calculated exponents r_1 and r_1 such that $\varepsilon = O(\delta^{r_1})$ and CPU time $= O(\varepsilon^{r_2})$. These are given at the bottom of Table I.

EXAMPLE 2. This example was constructed in order to compare the front tracking method with two other methods: Glimm's method and Lax–Friedrichs' method. Since the front tracking method will be least efficient when there are many interactions, we constructed an initial value problem which gives many interactions of fronts in a short time. For the Lax–Friedrichs method and the random choice method we used a uniform grid for the space variable and calculated the optimal time step at each time level according to the CFL condition. For a description of Glimm's method we refer the reader to Chorin [1] or Gottlieb [6].

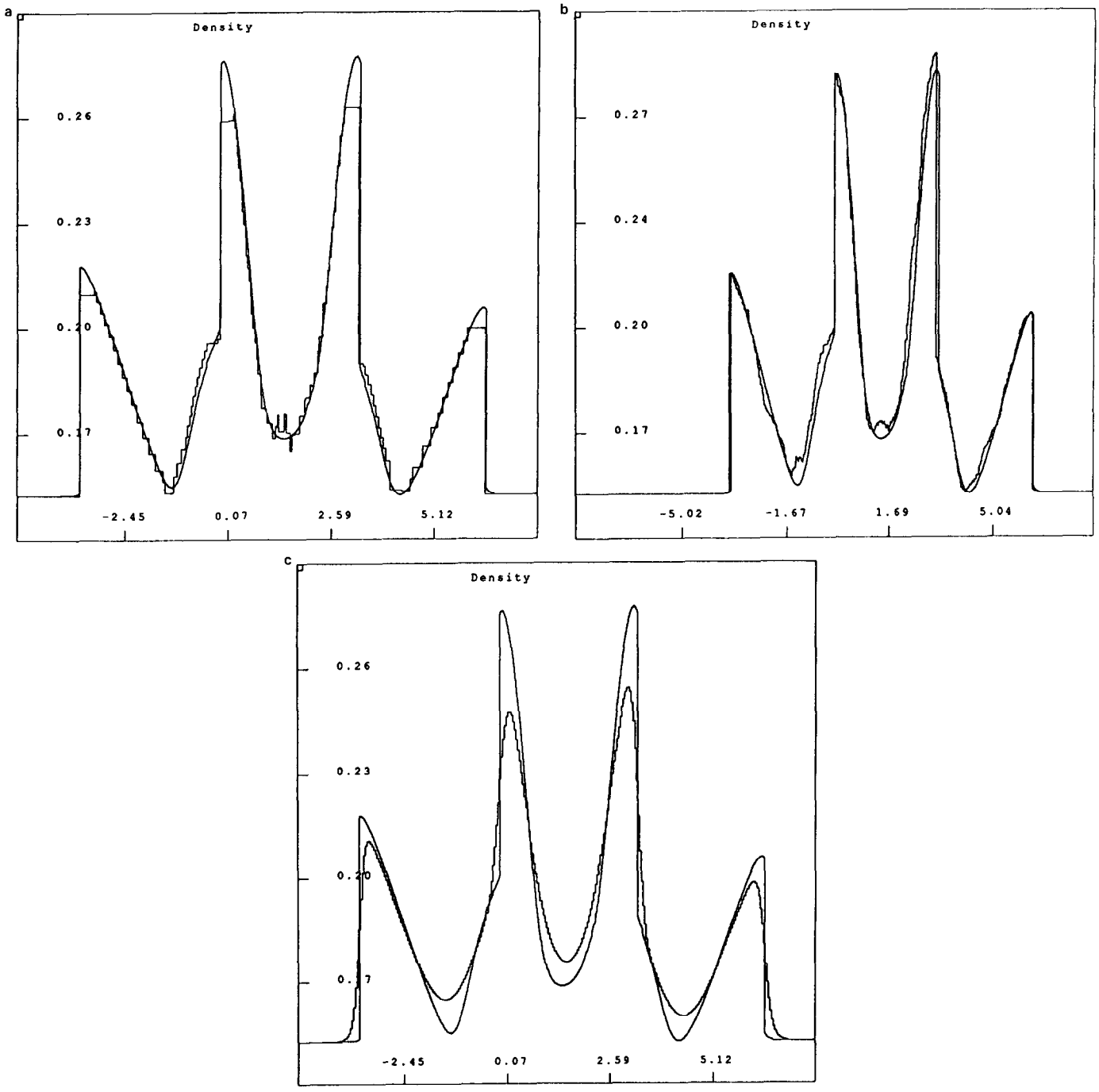

FIG. 4.3. (a) Problem #2. The reference solution and front tracking, $\delta = 1.0$, CPU time = 10.1; (b) The reference solution and Glimm's method, $\Delta x = 0.02$, CPU time = 219.2; (c) The reference solution and Lax–Friedrichs method, $\Delta x = 0.04$, CPU time = 13.5.

The problem consisted of an open shock tube with initial velocity everywhere zero, the initial sound speed everywhere 10.0, and the pressure distribution was given by

$$p_0(x) = \begin{cases} f(-7) & \text{for } x < -7 \\ f(x) & \text{for } -7 < x < 7 \\ f(7) & \text{for } x > 7, \end{cases} \tag{4.3}$$

where

$$f(x) = 11 + 20e^{-5x^2} + 15e^{-5(x-3)^2}. \tag{4.4}$$

Also for this problem we used a γ-gas law with $\gamma = 1.4$. We calculated this solution up to $t = 0.25$. As a reference solution we used the front tracking method with $\delta = 0.03$. This run had more than 9,000,000 collisions of fronts. In Fig. 4.3 we see the density of the approximate solutions of the three methods: (a) for front tracking, (b) for Glimm's method, and (c) for the Lax–Friedrichs method, as well as the density of the reference solution, at $t = 0.25$. We see that a leftward and a rightward moving wavegroup have formed, and that the solution contains four strong shocks. In Table II we present CPU times and errors for the three methods. Both Glimm's method and the front tracker have roughly the right discontinuities in the right places, but the front tracker only uses a fraction of the CPU time of Glimm's method. The front tracker does not have the same peak values as the reference solution, which is due to the approximation of the initial value function. The Lax–Friedrichs method is reasonably fast, but has poor resolution of the shocks. To investigate the "convergence rates" of the different methods, we used regression analysis to calculate exponents r such that CPU time $= O(\varepsilon^r)$ for each method. These indicate that the three methods have roughly the same r's, but note that in order to obtain an error of, e.g., 0.06, the front tracking method needs a CPU time of 21 while the Lax–Friedrichs method needs 622 and Glimm's method does not reach this level of accuracy despite a CPU time of 700.

EXAMPLE 3. The front tracking method is well suited to study asymptotic behaviour. In order to investigate this we have chosen the third example such that the initial data are "close to" a Riemann problem. The initial sound speed and velocity are 10.0 and 0.0, respectively, and the initial pressure has the distribution

$$p_0(x) = \begin{cases} 210 & \text{for } x < -25 \\ 100(1 - \tanh(x)) + 10 & \text{for } -25 < x < 40 \\ 10 & \text{for } x > 40, \end{cases} \tag{4.5}$$

We calculated the solution to this problem up to $t = 2.0$. At

TABLE IIa

Front Tracking

δ	CPU	Error
1.0	10.18	0.118
0.9	13.30	0.095
0.7	21.90	0.058
0.6	42.06	0.045
0.5	69.02	0.043
0.4	121.74	0.040
0.3	267.00	0.031
0.2	799.66	0.020

Note. Based on a standard regression analysis we compute: $\varepsilon = O(\delta^{1.0})$, standard error estimate for the exponent 0.1; CPU time $= O(\varepsilon^{-2.5})$, standard error estimate for the exponent 0.3.

TABLE IIb

Glimm's Method

Δx	CPU	Error
0.10	6.7	0.708
0.09	8.0	0.616
0.08	13.1	0.726
0.07	23.5	0.512
0.06	26.8	0.410
0.05	43.5	0.592
0.04	61.7	0.372
0.03	96.9	0.197
0.02	219.2	0.231
0.01	696.9	0.144

Note. Based on a standard regression analysis we compute: $\varepsilon = O(\Delta x^{0.7})$, standard error estimate for the exponent 0.1; CPU time $= O(\varepsilon^{-2.4})$, standard error estimate for the exponent 0.4. The number of grid blocks is $14.0/\Delta x$.

TABLE IIc

Lax–Friedrichs Method

Δx	CPU	Error
0.10	2.22	0.398
0.09	2.72	0.367
0.08	3.48	0.332
0.07	4.42	0.303
0.06	6.10	0.268
0.05	8.98	0.234
0.04	13.54	0.200
0.03	55.24	0.159
0.02	233.24	0.115
0.01	347.44	0.070
0.008	622.48	0.061
0.006	1403.46	0.031

Note. Based on a standard regression analysis we compute: $\varepsilon = O(\Delta x^{0.81})$, standard error estimate for the exponent 0.03; CPU time $= O(\varepsilon^{-2.4})$, standard error estimate for the exponent 0.1. The number of grid blocks is $14.0/\Delta x$.

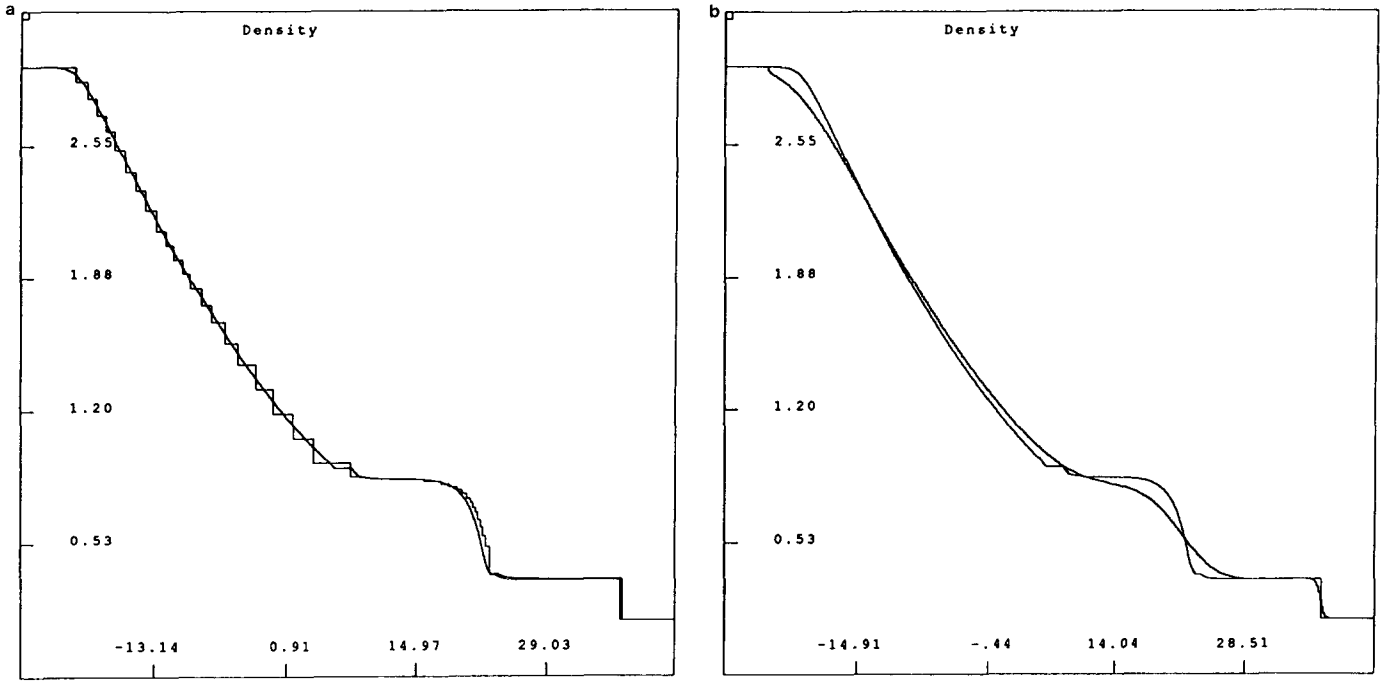

FIG. 4.4. (a) Problem #3. The reference solution and front tracking, $\delta = 12.0$, CPU time = 1.4; (b) The reference solution and Lax–Friedrichs method, $\Delta x = 0.2$, CPU time = 17.0.

TABLE IIIa

Lax–Friedrichs Method

Δx	CPU	Error
0.5	2.98	0.171
0.4	4.34	0.145
0.3	7.64	0.118
0.2	17.02	0.087
0.1	66.62	0.051

Note. Based on a standard regression analysis we compute: $\varepsilon = O(\Delta x^{0.75})$, standard error estimate for the exponent 0.01; CPU time $= O(\varepsilon^{-2.58})$, standard error estimate for the exponent 0.02.

TABLE IIIb

Front Tracking

δ	CPU	Error
12.0	1.42	0.056
10.0	1.86	0.049
8.0	3.42	0.042
6.0	6.02	0.034
4.0	13.50	0.024
3.0	24.26	0.022
2.0	62.86	0.012

Note. Based on a standard regression analysis we compute: $\varepsilon = O(\delta^{0.8})$, standard error estimate for the exponent 0.1; CPU time $= O(\varepsilon^{-2.5})$, standard error estimate for the exponent 0.1.

this time the solution resembles the solution to the Riemann problem with $p_l = 210.0$ and $p_r = 10.0$, except that the contact discontinuity has not yet developed fully. For this problem the front tracking method was compared with the Lax–Friedrichs method. The reference solution for this problem used $\delta = 0.5$. Figure 4.4 shows the solution computed by Lax–Friedrichs method (a) and the solution computed by the front tracker (b), as well as the reference solution, all at $t = 2.0$. Although the solution obtained by the front tracker used less than one tenth of the CPU time of that obtained by the Lax–Friedrichs method, the two solutions have roughly the same accuracy. In Table III we give the CPU times and errors for the two methods. The results here are all in the same vein as the results in Table II.

5. CONCLUSION

We have described a general front tracking algorithm for conservation laws in one space variable. This algorithm can be used for any system for which the solution to the Riemann problem is computable.

The front tracker was tested on the system of compressible non-stationary gas dynamics. In our test examples the front tracking method gives more accurate solutions in less CPU time than either the Lax–Friedrichs method or Glimm's method. The front tracker is well suited to study the asymptotic behaviour of an initial value problem. The method is nonconservative, but the errors in the conserved variables are small.

We aim to continue our study of this front tracking algorithm by applying it to a system of equations where the solution of the Riemann problem is more complicated.

REFERENCES

1. A. J. Chorin, *J. Comput. Phys.* **22**, 517 (1976).
2. C. M. Dafermos, *J. Math. Anal. Appl.* **38**, 33 (1972).
3. J. Glimm, *Commun. Pure Appl. Math.* **18**, 697 (1965).
4. J. Glimm, J. Grove, B. Lindquist, O. McBryan, and G. Trygvason, *SIAM J. Sci. Stat. Comput.* **9**, 61 (1988).
5. J. Glimm, C. Klingenberg, O. McBryan, B. Plohr, D. Sharp, and S. Yaniv, *Adv. Appl. Math.* **6**, 259 (1985).
6. J. J. Gottlieb, *J. Comput. Phys.*, to appear.
7. J. J. Gottlieb and C. P. T. Groth, *J. Comput. Phys.* **78**, 437 (1988).
8. G. W. Hedstrom, "Some Numerical Experiments with Dafermos' Method for Nonlinear Hyperbolic Equations," in *Lecture Notes in Mathematics* (Springer-Verlag, Berlin, 1972), p. 117.
9. H. Holden, *Commun. Pure Appl. Math.* **40**, 229 (1987).
10. H. Holden, L. Holden, and R. Høegh-Krohn, *Comput. Math. Appl.* **15**, 595 (1988).
11. T. Johansen, A. Tveito, and R. Winther, *SIAM J. Sci. Stat. Comput.* **10**, 846 (1989).
12. R. J. LeVeque, *SIAM J. Numer. Anal.* **19**, 1051 (1982).
13. B. J. Lucier, *Math. Comput.* **46**, 59 (1986).
14. N. H. Risebro, *Proc. AMS*, to appear.
15. J. Smoller, *Shock Waves and Reaction–Diffusion Equations* (Springer-Verlag, New York, 1983).
16. B. K. Swartz and B. Wendroff, *Appl. Numer. Math.* **2**, 385 (1986).
17. P. Woodward, "Trade-offs in Designing Explicit Hydrodynamical Schemes for Vector Computers," in *Parallel Computations*, edited by G. Rodrigue (Academic Press, New York, 1982), p. 153.
18. P. Woodward and P. Colella, *J. Comput. Phys.* **54**, 115 (1984).