JOURNAL OF COMPUTATIONAL PHYSICS **124,** 383–395 (1996)
ARTICLE NO. 0067

A Second-Order Godunov Method on Arbitrary Grids

ESTEBAN G. TABAK

Courant Institute of Mathematical Sciences, New York University, New York 10012

Received June 18, 1993; revised August 28, 1995

A second-order Godunov method is proposed for the solution of general systems of conservation laws on arbitrary grids. Some applications are discussed: moving and deforming grids, local grid refinement, Lagrangian grids that make contact discontinuities perfectly sharp, and a new way to solve the time dependent small disturbance transonic flow equations of gas dynamics. As part of the algorithm, a way is presented to solve generalized Riemann problems with second-order accuracy.

INTRODUCTION

Systems of conservation laws arise in most branches of science and engineering; their numerical solution is often required to settle both practical and theoretical issues. Although many types of numerical methods have been applied to systems of conservation laws, it is generally established that the so-called "conservative" schemes are best suited for problems involving shock waves, since they treat weak solutions in a natural way. One of the most popular conservative methods is one created by Godunov [4]. Its popularity is due to its robustness and conceptual simplicity and to the later work of Van Leer [14, 15], who developed higher order versions of the scheme (Godunov's original was first-order accurate), and Colella and Woolworth [2], who applied it successfully to many problems in fluid dynamics.

The algorithm proposed in this paper is a generalization of these high order Godunov schemes that works in general grids. The original motivation for it came about in a study of the von Neumann paradox of oblique shock reflection [12]. We were led to study the equations of unsteady small disturbance transonic flow and found that these could be most easily solved in an "oblique" system of coordinates in space-time [13] which required a mild generalization of a standard second-order Godunov. We soon realized that the same ideas could be applied to a wide class of practical problems, including grid refinement, moving domains, and the accurate tracking of contact discontinuities. A brief description of these applications constitutes the third section of this paper.

In the first section we describe the new scheme. For the sake of clarity in the exposition, we deal directly with the generalized algorithm. The previous high order Godunov methods on which it is strongly based can be found in the original bibliography ([2, 4, 14, 15]) and in the book of LeVecque [8].

In the second section we introduce a rigorous second-order accurate way of solving the generalized Riemann problem for arbitrary systems of conservation laws. This is a tool required by all high order Godunov methods; the algorithm presented here is quite general and conceptually simple.

1. A SECOND–ORDER CONSERVATIVE SCHEME ON ARBITRARY GRIDS

We will develop an algorithm to solve one-dimensional systems of conservation laws of the form

$$u_t + f(u)_x = 0, \tag{1}$$

where $u(x, t)$ and $f(u)$ are n-dimensional vectors, and the Jacobian matrix $f'(u)$ has a complete set of real eigenvalues. Such systems describe how the integral of the density u over an interval changes due to the flux f across its boundaries. As Eqs. (1) are hyperbolic and generally nonlinear, they may develop discontinuities even from smooth initial data. After these discontinuities appear, we need to give a meaning to (1). This leads to the consideration of weak solutions, that may be defined in many equivalent ways. We will choose one which has a clear interpretation in terms of grids. Let us first rewrite (1) in the form

$$\nabla \cdot (f(u), u) = 0,$$

in which the divergence is to be computed in the (x, t) plane. Then apply Gauss' theorem to any closed curve S in (x, t), to get

$$\int_S (f(u), u)_n \, dS = 0. \tag{2}$$

Finally, define a weak solution to (1) as any piecewise smooth function $u(x, t)$ which satisfies (2) for all closed curves S. Notice that a grid provides a natural discrete

0021-9991/96 $18.00

"basis" for the space of all closed curves, namely the grid's cells.

All smooth solutions to (2) satisfy (1), but (2) also admits discontinuities, which must satisfy the "jump conditions"

$$[u]\,dx - [f]\,dt = 0,$$

where the brackets stand for jumps in the enclosed variables. These constraints, however, are not enough, since they allow too many discontinuities, making the solution to the initial value problem for (2) generally nonunique. There are many ways to get rid of this nonuniqueness: we may require the solution to be stable under small perturbations, to be the limit of a well-posed viscous modification of (1), or to satisfy an appropriate "entropy condition." In either characterization, what we have is a definite direction of time; some phenomena, such as the dissipation of energy at shocks, are irreversible. For our algorithm, we will see below that this determines how oblique some edges of the computational grid may be.

Equation (2) will be our basic building block. Applied to a cell, it yields a relation between the averages $\bar{u}$ and $\bar{f}$ of u and $f(u)$ at its edges. Notice that, in order to work with an arbitrary grid in (x, t), we need to stop considering u and f as two different entities, and, instead, to deal with the "generalized" flux or density $\bar{f}\Delta t - \bar{u}\Delta x$, where Δx and Δt, the projections of an edge on the x and t axis, are assigned a sign according to their orientation with respect to the cell's interior. The equation for a cell then becomes

$$\sum (\bar{f}_i \Delta t_i - \bar{u}_i \Delta x_i) = 0. \tag{3}$$

The resulting system of equations is not enough to determine the $\bar{u}$'s, as can be verified by simply counting the number of cells and edges of a grid. In order to remedy this, we need to distinguish between two kinds of edges, that we will call "spatial" and "temporal." For the time being, we should think of the spatial edges as those approximately oriented in the direction of the x-axis and of the temporal edges as those roughly in the t-direction. The idea will be to compute the solution at the spatial edges from (3) and, at the temporal edges, from some simple initial-value-like problems. In order to understand how to do this, let us begin with the description of a first-order method, a generalization of Godunov's original one.

In this first-order method, we replace the functions u and f along the edges by their averages $\bar{u}$ and $\bar{f}$. Suppose that, while solving the equations on a grid, we are at the stage illustrated on Fig. 1, in which the values of $\bar{u}$ are known at the spatial edges $S1$, $S2$, and $S3$ and are to be found at the temporal edges $T1$, $T2$, and $T3$ and the spatial ones $S4$ and $S5$.

The values at $S4$ and $S5$ will be computed using (3),

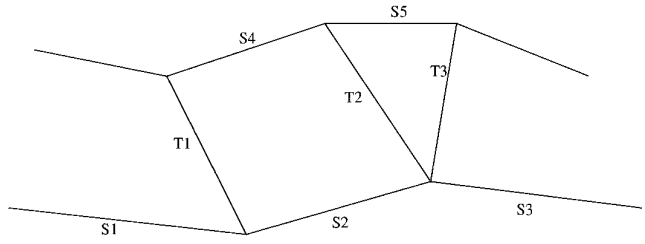

FIG. 1. Spatial and temporal edges.

once the ones at $T1$, $T2$, and $T3$ are known. To find these, we need one extra capability: that of advancing in time piecewise constant initial data, i.e., solving the Riemann problem for (1) with initial data provided on $S1$ and $S2$ or $S2$ and $S3$. Thus, the value of $\bar{u}$ on $T1$ will be computed by extending forward along $T1$ the data on $S1$ and $S2$, while the values at $T2$ and $T3$ will arise from those at $S2$ and $S3$. Let us now see what restrictions this procedure imposes on our grid (see Fig. 2).

We need the initial-value problems to be well posed. This restricts the slope of the spatial edges, which must be smaller than those of the characteristics. In other terms, the direction of these edges must be "space-like."

The temporal edges must lie entirely within the domain of influence of two consecutive spatial edges alone. This restriction is of the Courant–Friederichs type, as can be seen by applying it to a case with horizontal spatial edges and vertical temporal ones, where it reduces to the condition $\Delta x/\Delta t > c$, where c is the maximum absolute value of the characteristic speed.

When applying Eq. (3) to a closed contour of our grid, we need to have one and only one spatial edge where the solution is to be found. Therefore, although any number (including zero) of temporal edges may leave a grid's node, the number reaching one from below must always be one.

Finally, notice that, when applying (3) to find $\bar{u}$ at a spatial edge, the value we are really computing is $\bar{f}\Delta t - \bar{u}\Delta x$. In order to determine $\bar{u}$, we need to make the approximation that $\bar{f} \approx f(\bar{u})$, which is fully consistent with the algorithm's spirit and second-order accurate in the variation of u along the edge. Then the value of $\bar{f}\Delta t - \bar{u}\Delta x$ will determine $\bar{u}$ uniquely, through the solution of a nonlinear system of equations, provided $f(u_1)\Delta t - u_1\Delta x \neq f(u_0)\Delta t - u_0\Delta x$ whenever $u_1 \neq u_0$. But this is true if $|\Delta x/\Delta t| > \max_{i,u} |\lambda_i(u)|$, where the λ_i's are the eigenvalues of $f'(u)$, and the maximization is carried out along any curve joining u_0 and u_1. This follows from considering the difference

$$\begin{aligned}
&\|(f(u_1) - f(u_0))\Delta t - (u_1 - u_0)\Delta x\| \\
&\qquad \geq (|\Delta x/\Delta t|\,\|u_1 - u_0\| - \|f(u_1) - f(u_0)\|)\,|\Delta t| \\
&\qquad \geq (|\Delta x/\Delta t| - \max_{i,u} |\lambda_i(u)|)\,\|u_1 - u_0\|\,|\Delta t| > 0.
\end{aligned}$$

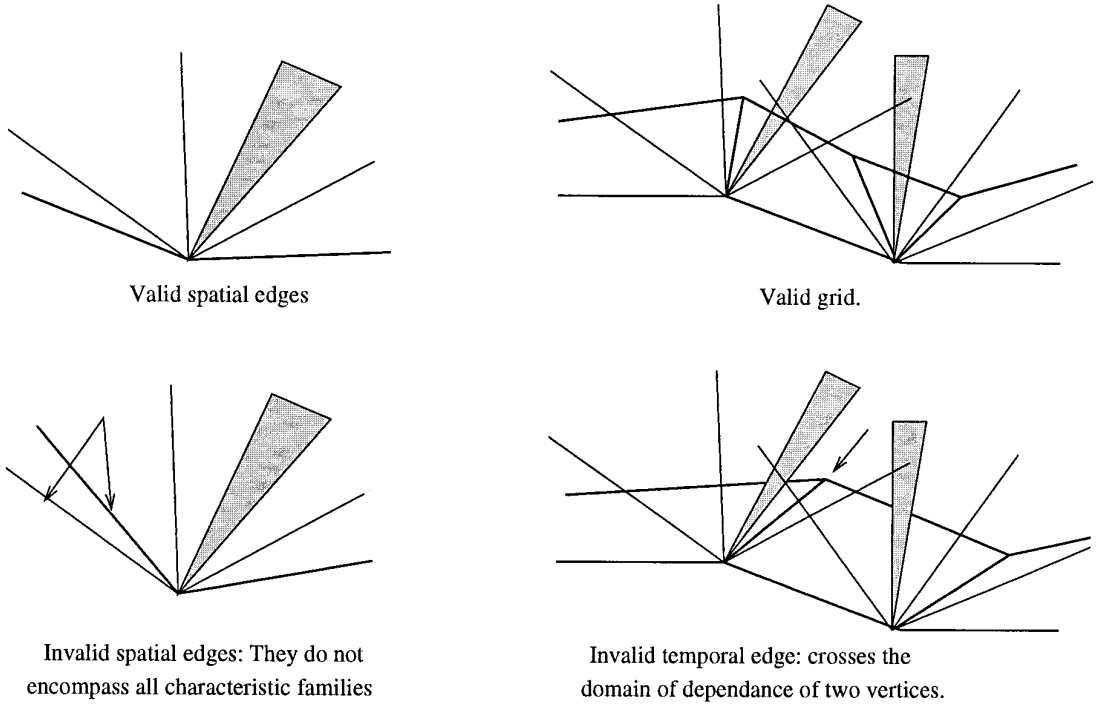

FIG. 2. Edges and characteristics. Edges are drawn in thick lines and characteristics in thin lines. Shadowed regions represent expansion fans.

This does not impose a new restriction on the spatial edges, since the well-posedness of the initial-value problems required the slope of a spatial edge to be bounded by the inverse of the maximum eigenvalue of $f'(u)$. Otherwise, we would march backwards in time at least along one characteristic family and, therefore, violate the entropy condition across this family's shocks.

With this restriction on the spatial edges, advancing the solution in time in a neighborhood of a grid point becomes a Riemann problem, even though the initial data are not really given on a line of constant time. We can see this by inverse reasoning: the solution to a real Riemann problem is always constant below the lines $x/t = \pm\max_{i,u} |\lambda_i(u)|$, so, in particular, it has the same values on the spatial edges as on the line $t = 0$.

Let us now preceed to a second-order algorithm. Its structure is essentially the same as the one described above. The grid is divided into spatial and temporal edges, with the restrictions already discussed. Then we apply a simple extension of Van Leer algorithm [15] to build higher order Godunov methods. For a second-order method, suppose that we have already found the values of $\overline{u}_i$ on a row of spatial edges, as in Fig. 3. We can compute a second-order accurate estimate for the average slope $\partial u/\partial q$ at every edge i (here q stands for a variable in the direction of the edge), by comparing the values of $\overline{u}_{i-1}$, $\overline{u}_i$, and $\overline{u}_{i+1}$ with the ones predicted by a Taylor expansion at x_i (this is actually true only if the points x_{i-1}, x_i, and x_{i+1} either are not aligned or lie all on the q axis; this adds a minor restriction to the design of a grid). The formula for $\partial u/\partial q$ is

$$\frac{\partial u}{\partial q} \approx \frac{k_2 u_{i-1} - k_1 u_{i+1} - (k_2 - k_1) u_i}{k_2 h_1 - k_1 h_2}$$

if the three points are not aligned and

$$\frac{\partial u}{\partial q} \approx \frac{h_1^2 (u_{i+1} - u_i) - h_2^2 (u_{i-1} - u_i)}{h_1^2 h_2 - h_1 h_2^2}$$

if they all lie on the q axis. Here $h_{1,2}$ and $k_{1,2}$ stand for the coordinates, in the directions of q and its normal (with origin at x_i), of x_{i-1} and x_{i+1}.

We further constrain these slopes with the monotonicity conditions due to Van Leer: If $\overline{u}_i$ lies between $\overline{u}_{i-1}$ and $\overline{u}_{i+1}$, we require the same from the linear interpolant $\overline{u}_i + u_q Q$ inside the ith interval; otherwise, we adopt $u_q = 0$. These restrictions, or their equivalent in the ENO schemes [6], are necessary to avoid spurious oscillations in the vicinity of shocks.

Once the linear interpolant for the $\overline{u}_i$'s has been built, we proceed to compute the average fluxes $\overline{u}$ and $\overline{f}(u)$ on the temporal edges connecting to the next row of spatial ones. This involves, however, solving with second-order accuracy a generalized version of the Riemann problem, in which the states at both sides of the discontinuity are

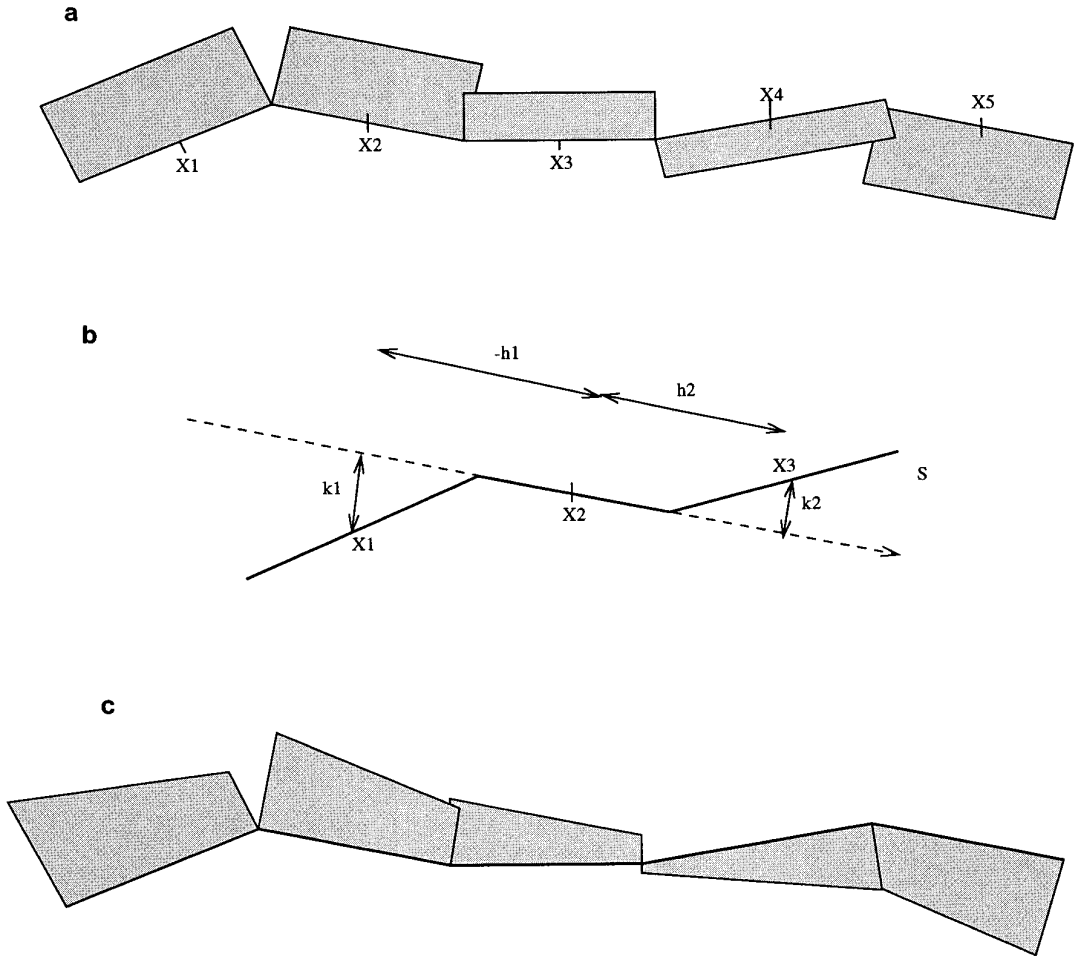

FIG. 3. Slope computation: (a) Data at the beginning of a step; (b) slope computation; (c) piecewise linear interpolant.

linear instead of constant. We show how to solve these generalized Riemann problems in the following section. With the fluxes computed, we can use (3) to find the $\overline{u}_i$'s on the new spatial edges.

2. THE GENERALIZED RIEMANN PROBLEM

A basic ingredient of the method just described is the computation to second order of the interaction between two contiguous cells. This section introduces a general algorithm that performs this computation if a standard Riemann solver is provided. The idea behind the algorithm is to find a first-order solution, henceforth denoted as basic state, and let the second-order perturbations propagate along its characteristics.

This algorithm should be viewed as an alternative to other ones proposed to solve the generalized Riemann problem. Flux-limiter methods, as the one discussed in [9], implicitly incorporate a generalized Riemann problem solver into a finite difference scheme, i.e., Lax–Wendroff. The method proposed here, instead, solves the generalized Riemann problem in a separate step. Closer in spirit are the algorithms proposed in [1, 3]; these, however, concentrate on a specific problem, i.e., gas dynamics. We will consider general systems instead. From a practical point of view, the algorithm proposed here and the ones in [3, 9] are in some sense equivalent, since they are all second-order accurate for weak discontinuities and they involve roughly the same amount of work. (Although the algorithm presented here has a more laborious outlook, it reduces, in its final implementation, to hardly more than the solution of a system of linear equations.) An advantage of the present algorithm is that it is conceptually simple, providing a clear understanding of the information flow along the characteristics. In addition, it is particularly appropriate for use with the general conservative method of this paper, since it computes the solution to second order at any point (x, t), not necessarily at $x = 0$.

The generalized Riemann problem may be posed as follows: Given an initial condition consisting of two smooth

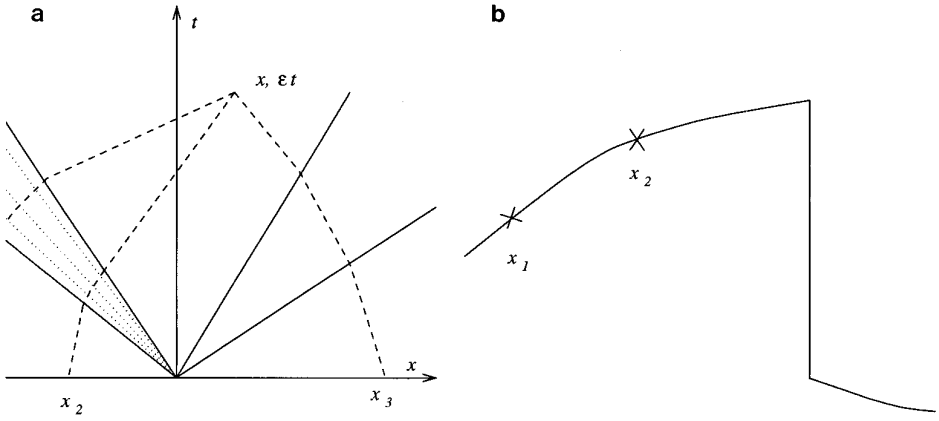

FIG. 4. (a) Tracing backwards the characteristics; (b) looking for the corresponding values in the initial configuration.

states separated by a discontinuity at $x = 0$, we are required to find the solution to (1) with second-order accuracy in time. Formally, we have the initial condition

$$u(x,0) = \begin{cases} u_{\mathrm{l}}(x) & \text{for } x < 0 \\ u_{\mathrm{r}}(x) & \text{for } x > 0 \end{cases} \tag{4}$$

with u_{l} and u_{r} smooth, and we want to compute $u(x, \varepsilon t)$ to $O(\varepsilon^2)$. For a second-order Godunov, the states on both sides of the discontinuity will be linear in the conserved quantities, but we will allow more general initial conditions. The method that we will describe remains basically unchanged with data given not on the x-axis but on any pair of space-like edges, as required by the algorithm of the previous section.

The plan of this section is the following: We will start by describing an algorithm which solves the generalized Riemann problem (4) with second-order accuracy. The algorithm consists of the solution of a set of standard Riemann problems, designed so that the information carried along every characteristic to $(x, \varepsilon t)$ is taken into account. This first algorithm is conceptually simple, but computationally inefficient, since it requires a relatively large number of operations. We proceed therefore to simplify it and give a second, much simpler version, which requires very little computational effort.

We will denote by (u_1, u_2) the solution to a Riemann problem with $u = u_1$ for $x < 0$ and $u = u_2$ for $x > 0$. The first algorithm starts computing the basic state, which is the exact solution to the Riemann problem $(u_{\mathrm{l}}(0), u_{\mathrm{r}}(0))$. A typical basic state consists of $n + 1$ constants (n being the number of components of the vector u) separated by n simple waves (shocks, rarefactions, or slip-lines).

Then we "trace back" the characteristics. By this we mean drawing the characteristic lines of the basic state which contribute to $(x, \varepsilon t)$ (see Fig. 4); we may replace the exact characteristics by straight lines without losing second-order accuracy. Denote by x_j the point where the characteristic j of the basic state hits the $t = 0$ axis (or, in the general case, the line in space-time where the data are provided). From the initial condition, we read off $u(x_j, 0)$. This leaves us with n states, one for each characteristic, that we will make interact to produce an estimate for the solution at $(x, \varepsilon t)$.

In our first procedure, we compute pairwise interactions between the states using a standard Riemann solver for (1). The algorithm is best described in the language of trees. We begin with the tree of traced-back characteristics of the basic state and transform it into a binary tree by adding intermediate nodes and edges, with the following restrictions: different edges should not cross (i.e., the order of the nodes must be respected), and branches coming from different sides of the initial discontinuity should not be combined until the uppermost node. Figure 5a shows two admissible trees; those on Fig. 5b, on the other hand, are not admissible. They all correspond to a system with six characteristic families.

Next we label the edges of this new binary tree with two numbers, denoting the leftmost and rightmost characteristic that they represent. For the lowest edges, the two numbers are equal; they are those of the corresponding characteristic family in the original tree. For a parent edge, the left number is the same as that of the left child, while the right number repeats that of the right child.

We now proceed to compute states associated with each node. The ones on the lowest row have already been assigned from the initial data. To find the state of a parent node given those of its two children, we solve the Riemann problem (ul, ur), where ul and ur are the states of the left and right children. Then we assign to the parent the constant state in this solution where the characteristics that label the left edge come from the left, and those that label the right edge come from the right (see Fig. 6). By construc-

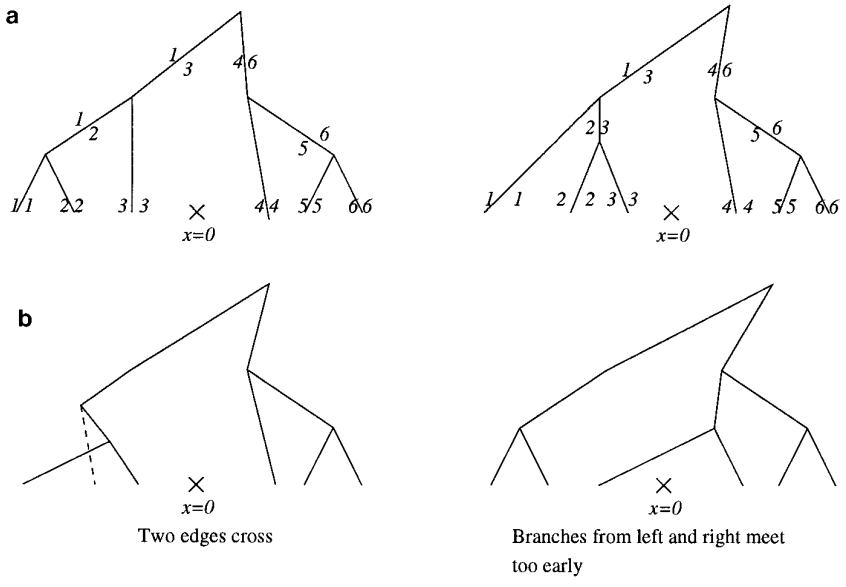

FIG. 5. (a) Valid trees; (b) invalid trees.

tion, the rightmost characteristic of the left child and the leftmost one of the right child are consecutive; therefore, the state to choose is the one located between the waves of these two families.

In summary, we trace the tree of characteristics, transform it into a binary tree, assign values to its lowermost nodes from the initial data, and compute a root node from its descendents. The value at the uppermost node is our estimate for the solution at $(x, \varepsilon t)$.

There is only one point left, namely what to do if $(x, \varepsilon t)$ lies inside a rarefaction wave of the basic state. In this case, the corresponding characteristic will trace back to $x = 0$ and no state can be assigned to it. Instead, we erase that edge from the tree altogether, and only in the last step, when computing the interaction between representatives from left and right, do we assign to the final node the state arising at $(x, \varepsilon t)$.

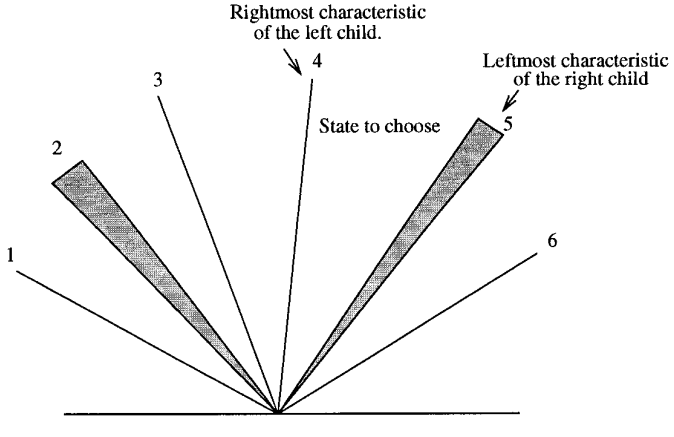

FIG. 6. Assignment of a state to a parent node in the Riemann problem between its two children.

Let us now examine the algorithm more closely to gain insight into why it works and how we can achieve the same results with less computational effort. Clearly, the whole algorithm is based on assuming that the domain of dependence of $(x, \varepsilon t)$ at $t = 0$ is the finite set of points x_j. This is only true if the system has Riemann invariants and, moreover, the solution we are dealing with is smooth. The reason why the validity of the method goes far beyond these restrictive hypotheses is that all the interacting states are close to each other. Thus there are indeed "pseudo Riemann invariants," which are the invariants of the system linearized at any of these states. These "invariants" are conserved to second order in ε in both the exact solution and our numerical procedure. Formally, if $u(x, t) = u_0 + O(\varepsilon)$, we can approximate (1) with $u_t + Au_x = 0$, where A is the Jacobian of $f(u)$ evaluated at u_0. If l_j^i is the jth component of the ith left eigenvector of A, then the quantities $R^i = \sum_{j=1}^n l_j^i u_j$ are pseudo Riemann invariants, meaning that, along the ith characteristic, R_i is conserved up to $O(\varepsilon^2)$. As both the real evolution and our pairwise interactions are close to the same state u_0 and they both start with the same values of these pseudo Riemann invariants, the difference between the exact value and the numerical estimate for $u(x, \varepsilon t)$ is $O(\varepsilon^2)$, proving that the method is second-order accurate.

This proof gives us a clue on how to obtain the same accuracy without having to solve n Riemann problems: If we know some linear expressions that are "almost" conserved along the characteristics, why not find the final state directly from these? The whole procedure would thus

reduce to solving a linear system of equations. Indeed it does so, but for a little detail. The states to the right of the initial discontinuity are close to each other, since we are assuming that the initial data is piecewise smooth. So are the states to the left. The states from left and right are also close in regions where the solution is smooth, which completes the proof of the order of the algorithm. But we would like the method to work also for large discontinuities, since shocks are a common occurrence in hyperbolic waves. This is the reason why, when forming the binary tree, the information from the left and the right was kept separate until the last step. This makes the estimate at least first-order accurate, even if the states on the right and the left are not close at all, because the last Riemann problem mimics the real one.

Thus the following algorithm suggests itself: Solve the basic Riemann problem and trace back the characteristics. Linearize the equations both to the left and to the right of the initial discontinuity, and calculate the expressions for the linear Riemann invariants on both sides. Compute a representative from the left and the right by imposing the conservation of these Riemann invariants. Then solve the final Riemann problem between these two states to calculate the solution at $(x, \varepsilon t)$. Of course, this last step can also be reduced to solving a system of equations, if we linearize this last Riemann problem in the spirit of Roe [7]. If we choose to do so, we can use this linearization globally and solve only one system of equations. But this is just one possible implementation of the standard Riemann solver, that we are in this section considering as a black box.

Let us write explicitly the system of equations to solve in order to find u^{left}, a representative from all the states coming from $x < 0$. Assume that, after tracing back the characteristics, we find that the first n_1 of them originated to the left of the initial discontinuity. Denote by u_j^i the jth component of the state at x_i, and by u^{lin} the state at which we have chosen to linearize the equations. This can be any of the u^i's with $i \le n_l$ or, more consistently, just $u_1(0)$. Using the corresponding pseudo Riemann invariants, we can find u^{left} from the system

$$\sum_{j=1}^{n} l_j^i u_j^{\text{left}} = \begin{cases} \sum_{j=1}^{n} l_j^i u_j^i & \text{for } i \le n_1 \\ \sum_{j=1}^{n} l_j^i u_j^{\text{lin}} & \text{otherwise.} \end{cases}$$

The choice of the right-hand side of the last $n - n_1$ equations was quite arbitrary, since any state close to the ones on the left of the initial discontinuity would have worked. We need second-order accuracy only in the first n_1 pseudo Riemann invariants of u^{left}; the others will not count (up to $O(\varepsilon^2)$) in the final determination of $u(x, \varepsilon t)$.

The description of the algorithm is now complete; let us see how it works in a simple example. We solved the generalized Riemann problem for a perfect gas with density ρ, velocity u, specific energy e, and pressure P given by $P = (\gamma - 1)\rho e$, with $\gamma = 1.4$. As initial data we took

$$\rho = \begin{cases} 4 - 0.1x, & x < 0 \\ 1 - 0.05x, & x > 0, \end{cases} \quad u = \begin{cases} -0.5 + 0.1x, & x < 0 \\ -1.0 + 0.1x, & x > 0, \end{cases}$$

$$e = \begin{cases} 3 - 0.1x, & x < 0 \\ 1 - 0.05x, & x > 0. \end{cases}$$

We computed the solution at $t = 1$ at 30 equidistant points between $x = -3$ and $x = 3$ with the algorithm just described, and plotted it with the stars in Fig. 7. The dotted lines correspond to the "exact" solution computed with the second-order Godunov of Section 1 with 300 points and $\Delta t = 0.005$. We used a locally Lagrangian grid (see Section 3 below) to avoid smearing the contact discontinuity.

We can see the nearly perfect agreement of the two solutions, even though neither the jump in the data nor the time interval are small, as required by the algorithm. The only perceptible consequence of this is an error in the location of the shock that our generalized Riemann solver moves at a constant velocity equal to its exact initial speed. It follows that, if we want to use the generalized Riemann solver of this section for long times and big discontinuities for some practical purpose, as for a fast, "manual" estimate of the consequences of a dam's break due to flooding, only the shock's location has to be further corrected, averaging, for instance, its initial and (predicted) final speeds. This is, of course, not necessary for the use of the Riemann solver as part of a second-order Godunov. There the time intervals have to be small, the jumps are of the order of a cell's size except at the shocks, and these move at the right speed due to the conservative nature of the algorithm.

3. APPLICATIONS

In this section, we describe some applications of the algorithm of Section 1. We show how it helps implementing grid refinement, designing locally Lagrangian grids for the computation of sharp contact-discontinuities, solving systems of conservation laws in moving domains, and dealing with changes of coordinates, as those occurring in the solution to the equations of unsteady transonic flow.

An algorithm that works on general grids is clearly well suited for local grid refinement. In particular, the flexibility provided by the inclusion of space-like edges (as opposed to purely spatial ones) enables us to treat the fluxes at the boundaries between fine and coarse sections of a grid in a natural way. At places where the grid is finer, the CFL condition requires the time intervals to be smaller as well.

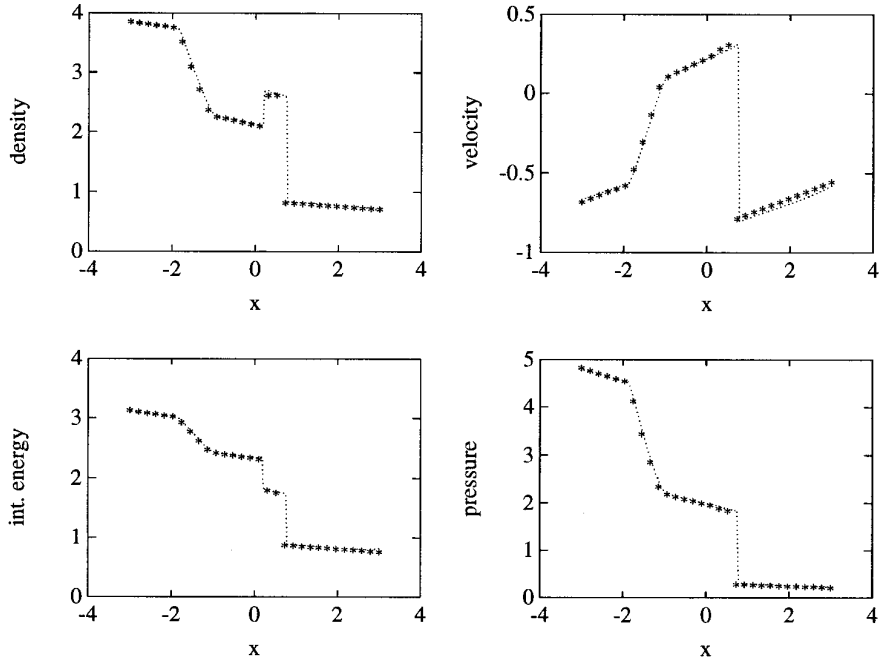

FIG. 7. Generalized Riemann problem for a perfect gas. Stars, one step solution; dots, generalized 2nd-order Godunov.

If one does not want to take the smallest Δt throughout the grid (a very expensive solution), one needs to provide internal boundary conditions for the finer grid at the intermediate times not computed in the coarser sections. This usually requires some ad hoc interpolation of the outer solution, which at best sheds some doubts on the accuracy of the solution at the first few cells of the finer grid. The grid plotted in Fig. 8, instead, shows a natural way to implement these intermediate boundary conditions. The oblique edges at the interface are all space-like, as follows from the CFL condition for the coarser grid.

Next we discuss an application to the tracking of contact-discontinuities, following an idea that, generalized to track any simple wave, was proposed by Harten and Hyman in [5]. There are regions in the solution to some problems in fluid dynamics—close, for instance, to the interface between two fluids—where a precise computation of passively transported quantities becomes important. A standard Godunov performs poorly on this, since the diffusion caused by the continuous averaging of the solution is not balanced, at linearly degenerate waves, by the nonlinear compression that keeps shock waves sharp. Instead, we can define locally a Lagrangian grid, moving at approximately the velocity of the fluid. It is easily seen that the

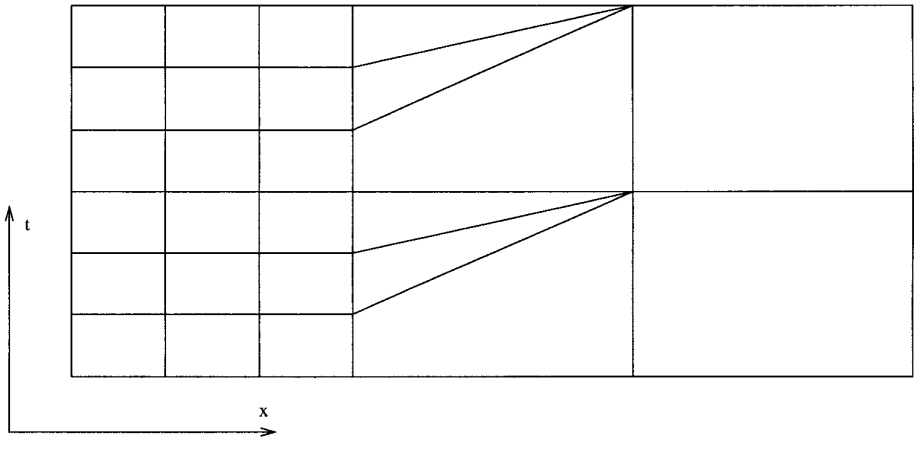

FIG. 8. Interface between fine and coarse grids.

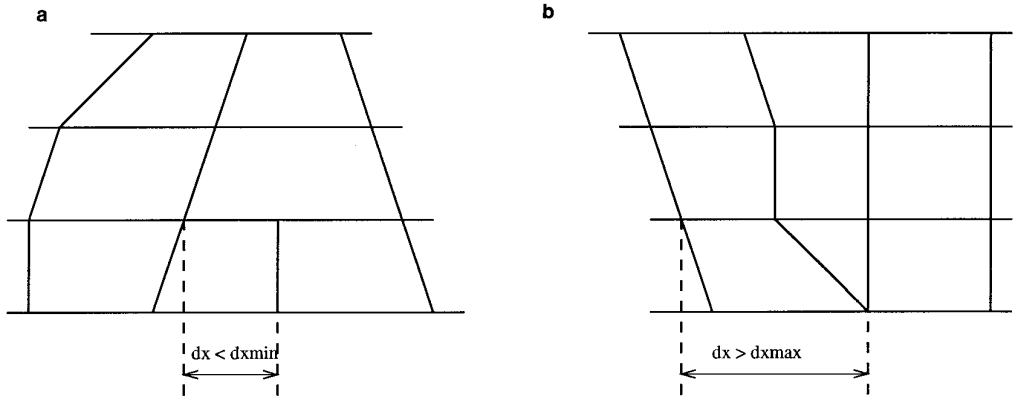

FIG. 9. (a) Deletion of a grid point; (b) addition of a grid point.

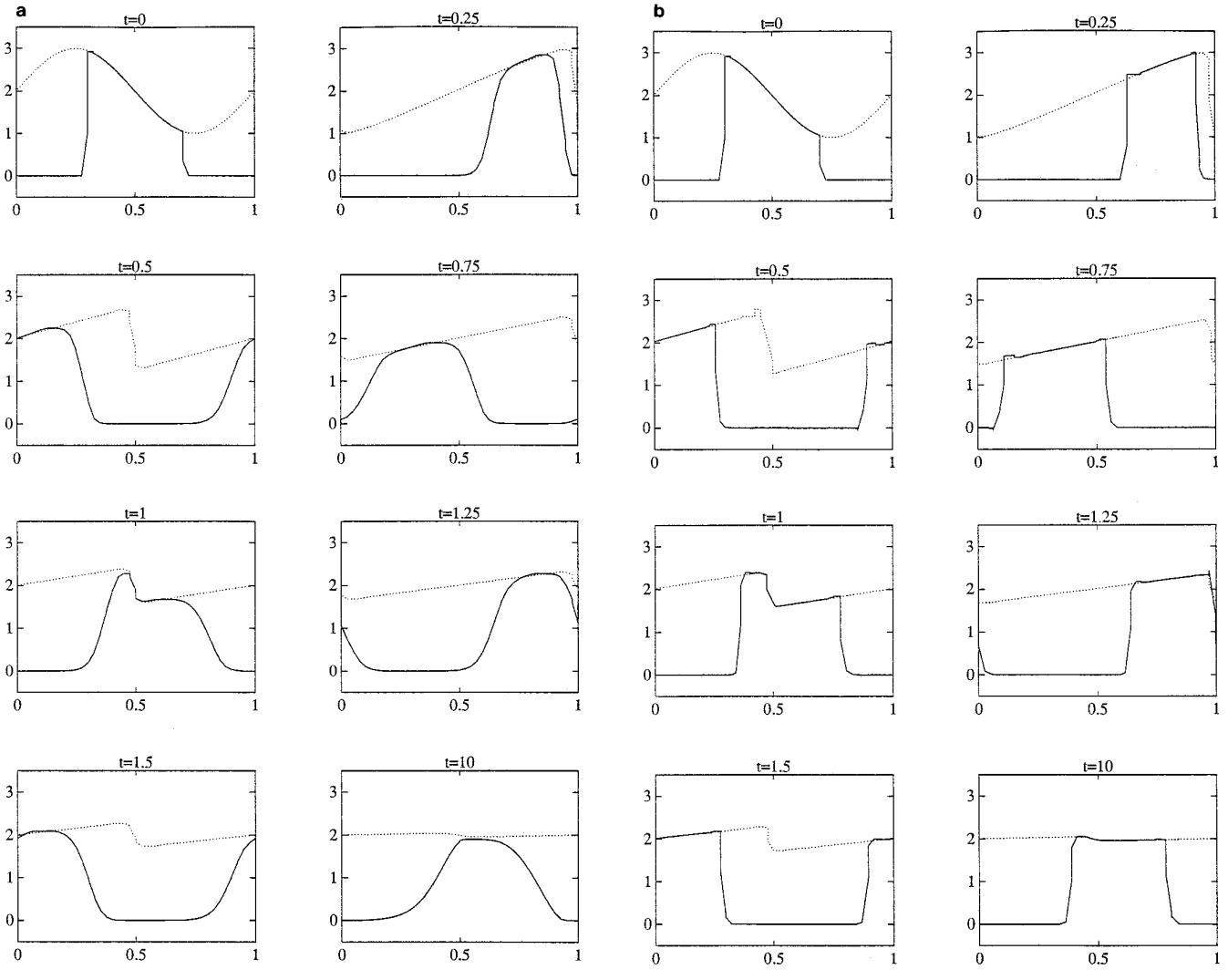

FIG. 10. Evolution of S and SC: (a) with a fixed grid; (b) with a Lagrangian grid.

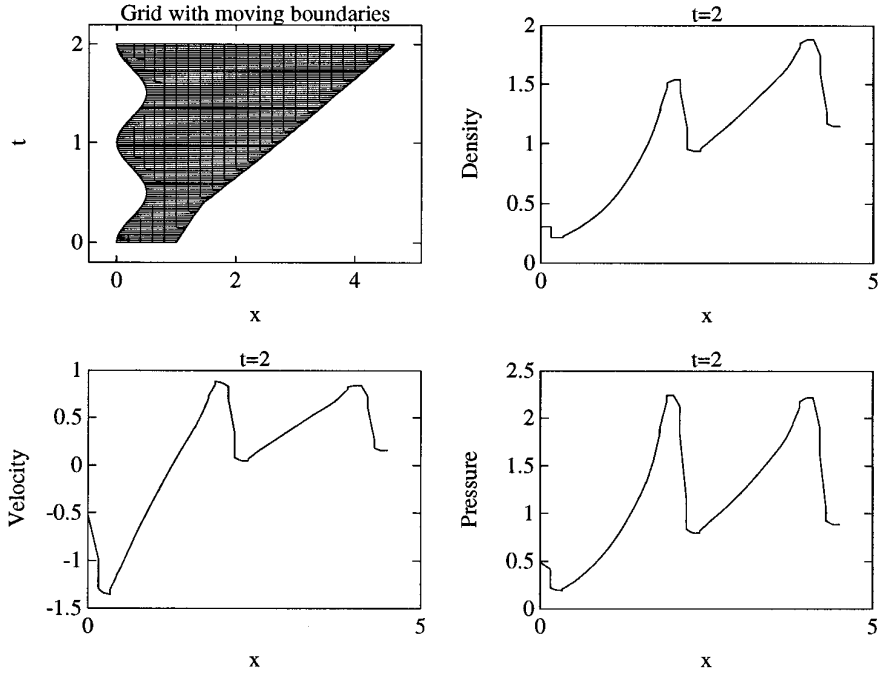

FIG. 11. Piston problem for a perfect gas.

flux of mass through the interfaces of such a grid will be negligible—null for a perfect Lagrangian grid—so the distribution of passive quantities will not be diffused due to the averaging that is intrinsic to Godunov's method.

As an example, consider the concentration of salt in a river. For simplicity, we will consider a prismatic channel with constant cross section and model it with the equation of conservation of mass,

$$S_t + Q_x = 0, \tag{5}$$

together with a hydrological law $Q = Q(S)$. Here x is the longitudinal coordinate along the reach, S is the area with water, and Q is the flux of water through it. The mean velocity of the flow is $U = Q/S$, and the equation for the convection of salt reads

$$(SC)_t + (QC)_x = 0, \tag{6}$$

where C is the concentration of salt. Notice that no diffusion was incorporated into this model, so an initially sharp distribution of salt should remain sharp forever. We would like the numerics to mimic this, not only because the diffusion may be really negligible, as is the case in most phenomena that involve small time scales, but also because we may be interested in modeling the diffusion based on physical considerations, and not on an uncontrollable numerical error. One solution is to use a Lagrangian grid close to discontinuities and high gradients of C.

We solved numerically (5) and (6) with $Q(S) = S^2/2$, periodic boundary conditions, and initial data

$$S(x,0) = 2 + \sin(2\pi x), \quad C(x,0) = \begin{cases} 1 & \text{for } 0.3 < x < 0.7 \\ 0 & \text{elsewhere,} \end{cases}$$

with a mixed grid, Lagrangian over a domain slightly larger than the support of C and Eulerian elsewhere. We have adopted the following simple rules for handling the grid: If two grid-points get closer to each other than a given distance *dxmin*, we erase one of them from the grid. Instead, if two contiguous points get further appart than a given *dxmax*, we create a new point in between. Notice that, with the algorithm of this paper, there is no need to arbitrarily redistribute averages when points are added or removed from the grid; the conservation laws applied to the grid take care of that. In Fig. 9, addition and removal of grid-points is exemplified.

The results with a fixed grid and with the one described above are plotted on Figs. 10a and b. The dotted lines correspond to S, while the continuous lines represent the product SC, the other conserved quantity. For both runs, we took only 40 grid points, about 20 of them Lagrangian for the second run, to underline the efficiency of the method.

For Eqs. (5) and (6), the nonlinear characteristic velocity is S and the linearly degenerate one is $Q/S = S/2$. We see

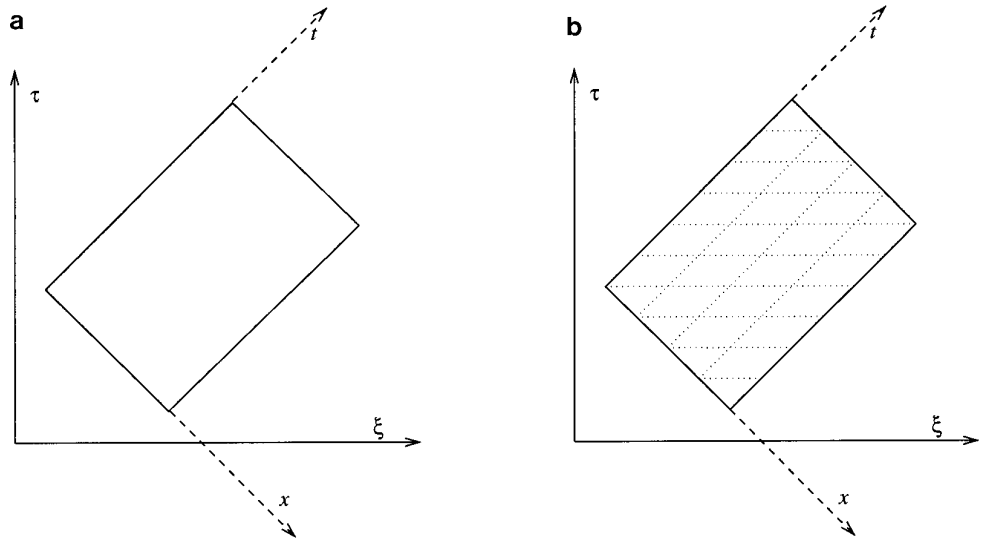

FIG. 12. (a) Domain of integration; (b) optimal grid.

how the Lagrangian grid handles difficult situations, like the repeated interaction of the contact discontinuity with a shock, without smearing the sharp initial profile of C. A fixed grid, on the other hand, does a poor job on this, smearing the discontinuities continuously, due to averaging. On the other hand, the absence of numerical viscosity is responsible for the appearance of slight overshoots near the discontinuities. Once such overshoots are created, due to the imperfect character of the monotonicity constraints, there is no viscous mechanism that will damp them away.

The same idea can be applied to gas dynamics. Here, a grid that locally moves with the fluid will account for very sharp slip-lines, something that would otherwise require sophisticated techniques (see, for instance, [17]). We have used a locally Lagrangian grid in the solution to the generalized problem for gas dynamics of Section 2; a fixed grid would have smeared the slip-line, requiring many more points to match the accuracy of the one-step generalized Riemann solver. One important point should be made though: this tracking technique for contact discontinuities works well (almost perfectly indeed) for one-dimensional problems; it does not seem to generalize in any simple way to the multidimensional case.

Another reason we may have to adopt a moving grid is that the region we are interested in may change in time. This is the case, for instance, of a gas initially at rest pushed from the left by a moving piston. In this case, we would like to have the left boundary of the grid coincident with the moving piston, while, on the right, we would like to have our domain growing so as to keep the first wave coming into the unperturbed state always inside. In Fig. 11, we see the grid in x-t space corresponding to a periodic movement of the piston, and the computed values of the density, velocity, and pressure at $t = 2$ for a perfect gas with $\gamma = 1.4$. The gas is initially at rest, with density and internal energy normalized to 1 and the movement of the piston given by

$$x = 0.25 * (1 - \cos(2\pi t)).$$

To make the adjustment of the grid automatic, we adopted for the velocity of the right boundary the maximum value of the rightgoing acoustic characteristic velocity of the fluid over the last few gridpoints. The rules for adding or deleting grid points were the same as for the problem of salt concentration in a river. The grid shown is very coarse (it starts with only five cells) for clarity in the plots; the numerical results can be made much more accurate by refining the grid.

When changes of coordinates are required, a problem with simple geometry in physical space may get moving boundaries in the new coordinate system, leading naturally to the application of the algorithm described in this paper. As an example, let us take the one that motivated this work: the equations of time dependent small disturbance transonic flow. These equations, which describe many diffraction patterns of weakly nonlinear geometrical acoustics (see, for example, [7, 11]), can be written in the form

$$\sigma_t + (\sigma^2/2)_x + \eta_y = 0$$
$$\eta_x - \sigma_y = 0.$$

Here σ is proportional to the first term in the perturbation expansions for the density, pressure, temperature, and longitudinal velocity of the gas, while η relates to the

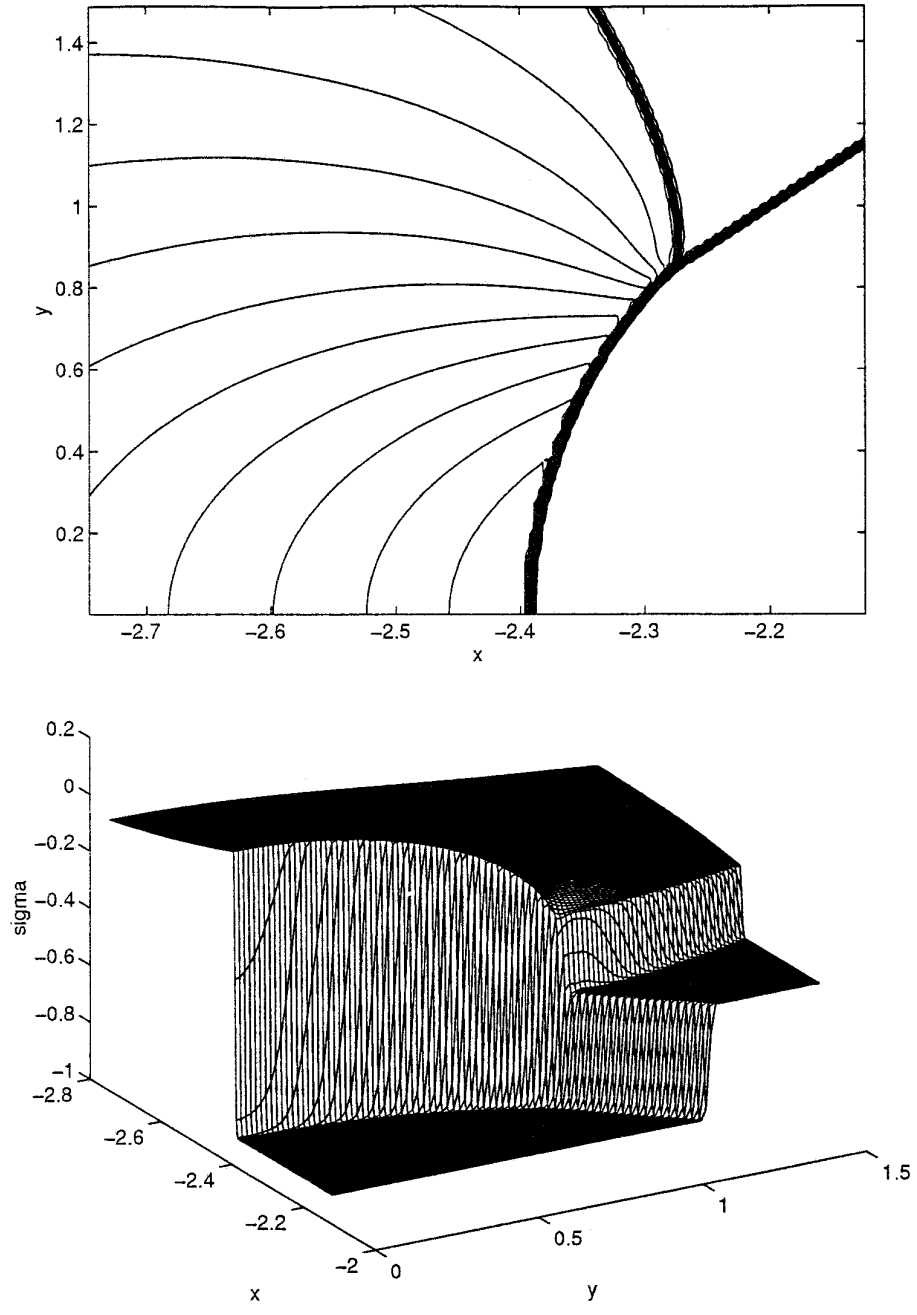

FIG. 13. Pseudo Mach-stem. Contour lines and spatial view of σ.

expansion of the transversal velocity. Normally, one would like to solve these equations in some rectangular domain, given initial and boundary conditions which depend on the problem. However, the planes with constant t turn out to be characteristic surfaces of these equations, making the initial value problem for them ill-posed. Although the mixed initial-boundary value problem one would like to solve is believed to be well posed, the characteristic nature of time makes the numerical solution of the system very difficult (for a full report on this and on the numerical scheme that will be briefly sketched here, see [13]). A way to avoid these difficulties, is to switch to a new coordinate system ξ, y, τ, where $\xi = t + x$ and $\tau = t - x$. In these coordinates, the equations read

$$(\sigma - \sigma^2/2)_\tau + (\sigma + \sigma^2/2)_\xi + \eta_y = 0$$

$$\eta_\tau - \eta_\xi + \sigma_y = 0.$$

Introducing $\omega = \sigma - \sigma^2/2$, with inverse $\sigma = 1 - \sqrt{1-2\omega}$ (assuming $\sigma < 1$ and $\omega < 1/2$), we get

$$\omega_\tau - (\omega + 2\sqrt{1-\omega})_\xi + \eta_y = 0$$
$$\eta_\tau - \eta_\xi - (\sqrt{1-2\omega})_y = 0.$$

Here τ is not a characteristic of the equations; indeed, it is a valid time-like variable. Thus we can think of advancing in τ instead of t, using a fractional-step alternate-direction procedure to decouple the ξ and y derivatives. The two systems to solve are

$$\omega_\tau - (\omega + 2\sqrt{1-\omega})_\xi = 0$$
$$\eta_\tau - \eta_\xi = 0$$

and

$$\omega_\tau + \eta_y = 0$$
$$\eta_\tau - (\sqrt{1-2\omega})_y = 0.$$

The first system decouples into two scalar equations, while the second may be viewed as describing a polytropic evolution of gas dynamics in Lagrangian coordinates, with ω acting as specific volume, $(-\eta)$ as velocity, and with pressure given by the convex $P(\omega) = \sqrt{1-2\omega}$. Thus both systems can be solved with high order Godunov methods. However, in the original variables, we had a mixed initial-boundary value problem, so the domain of integration, for fixed y, has the shape sketched in Fig. 12a. This region clearly cannot be covered by a fixed grid, for its boundaries are moving. Thus the algorithm of this paper is called upon. Among the various ways to design a grid apt for this problem, a simple analysis based on the characteristic velocities led us to choose the one drawn on Fig. 12b.

As an example of the use of this algorithm, we have plotted in Fig. 13 its computation of the ''pseudo Mach-stem'' arising in the context of the von Neumann paradox of oblique shock reflection. Here the initial data consists of a single shock wave separating two constant states, impinging obliquely upon a wall. The boundary data are the no-flux condition through the wall $\eta = 0$ at $y = 0$ and the absorbing conditions at the other three numerical boundaries.

The paradox that von Neumann observed in [16] and that since has been the subject of many research efforts (see [12] and references therein) is that, for small angles of incidence and weak shocks, the solutions observed both numerically and experimentally appear to be inconsistent with the equations of gas dynamics.

In Fig. 13, we have plotted contour lines and a perspective of σ. The point where three shocks appear to meet constitutes the heart of the paradox, since the equations do not admit such triple shocks. For a full account on this, as well as on similar applications of this algorithm to the numerical elucidation of the self-focusing of waves and the structure of nonlinear singular rays, we refer the interested reader to [12].

ACKNOWLEDGMENTS

This work arose as an independent numerical development within a joint project with Rubén R. Rosales [12, 13]. I thank Rubén for his many suggestions and helpful discussions. Partial support for this work was provided by an Alfred Sloan Doctoral Dissertation Award, by NSF under Grants DMS-9008520 and DMS-9001805, and by DARPA under Grant N0014-92-J-1796.

REFERENCES

1. M. Ben-Artzi and J. Falcovitz, *SIAM J. Sci. Stat. Comput.* **7,** 744 (1986).
2. P. Colella and P. R. Woodward, *J. Comput. Phys.* **54,** 174 (1984).
3. P. Colella, *SIAM J. Sci. Stat. Comput.* **6,** 104 (1985).
4. S. K. Godunov, *Mat. Sb.* **47**(3), 271 (1959).
5. A. Harten and J. M. Hyman, *J. Comput. Phys.* **50,** 235 (1983).
6. A. Harten and S. Osher, *SIAM J. Numer. Anal.* **24,** 279 (1987).
7. J. K. Hunter, *SIAM J. Appl. Math.* **48,** 1 (1988).
8. R. J. Leveque, *Numerical Methods for Conservation Laws* (Birkhäuser, Basel, 1990).
9. R. J. Leveque, University of Washington Technical Report No. 92-11, 1992 (unpublished).
10. P. L. Roe, *J. Comput. Phys.* **43,** 357 (1981).
11. R. R. Rosales, ''Diffraction Effects in Weakly Nonlinear Detonation Waves,'' in *Proceedings, Seminaire International sur les Problèmes Hyperboliques de Bordeaux 1988,* Lecture Notes in Phys. (Springer Verlag, New York/Berlin, 1988).
12. E. G. Tabak and R. R. Rosales, *Phys. Fluids* **6**(5), 1874 (1994).
13. E. G. Tabak, R. R. Rosales, in preparation.
14. B. Van Leer, *J. Comput. Phys.* **23,** 276 (1976).
15. B. Van Leer, *J. Comput. Phys.* **32,** 101 (1979).
16. J. Von Neumann, ''Oblique Reflection of Shocks,'' in *Collected Works* (Pergamon, New York, 1963).
17. H. Yang, *J. Comput. Phys.* **89,** 125 (1990).