LA-UR 96-2932

TITLE: **AN IMPLICIT METHOD FOR HYPERBOLIC CONSERVATION LAWS ON MESHES WITH SMALL CELLS IN ONE DIMENSION**

AUTHOR(S): Karen Pao and Jeff Saltzman

SUBMITTED TO: For External Distribution

Los Alamos

Los Alamos National Laboratory
Los Alamos New Mexico 87545

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

DISCLAIMER

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

An Implicit Method for Hyperbolic Conservation Laws on Meshes with Small Cells in One Dimension

Karen Pao Jeff Saltzman

August 14, 1996

1 Introduction

In this paper we study a new method for solving hyperbolic conservation laws on a Cartesian mesh with some small cells. Our main task here is to devise a stable algorithm in the small zones. An algorithm proposed by Berger and LeVeque [1] [2] combines the small zones with neighboring zones and solves rotated Riemann problems. This method is very geometrically oriented. It requires knowledge of the areas of the small cells, as well as the areas of cells contained in various "boxes" drawn from the edges of the small cells. Here we propose a more algebraic algorithm: we combine an implicit method with an explicit second-order conservative finite difference scheme. In section 2 the basic algorithm in one dimension is presented, as well as the slopes calculations and the iterative procedure. In section 3 we present some numerical results for the 1-D advection equation and the inviscid Burger's equation.

2 The Implicit Flux Method

2.1 The Basic Algorithm

We want to solve

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

on a one-dimensional grid with one small cell. If the grid size were uniform, our finite difference method would be

$$u_j^{n+1} = u_j^n - \frac{\Delta x}{\Delta t}(f_{j+1/2} - f_{j-1/2}) \tag{2}$$

where u_j^n is the average of u in cell j at time level n, and $f_{j+1/2}$ is the flux at the edge $j + 1/2$. $f_{j+1/2}$, is computed by solving the Riemann problem

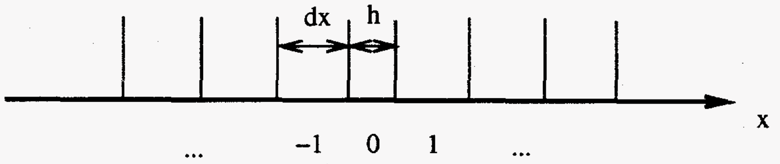

Figure 1: The finite difference mesh for $u_t + f(u)_x = 0$.

$R_l(u_{j+1/2,L}, u_{j+1/2,R})$, with $u_{j+1/2,L}$, $u_{j+1/2,R}$ obtained from Taylor expansions:

$$u^{n+1/2}_{j+1/2,L} = u^n_j + \tfrac{1}{2}(\Delta x - \Delta t \frac{\partial f(u_j)}{\partial u})\frac{\partial u}{\partial x} \tag{3}$$

$$u^{n+1/2}_{j+1/2,R} = u^n_{j+1} - \tfrac{1}{2}(\Delta x + \Delta t \frac{\partial f(u_{j+1})}{\partial u})\frac{\partial u}{\partial x} \tag{4}$$

The derivative $\frac{\partial u}{\partial x}$ would be computed by a limited slopes formula. For example, the limiter due to van Leer [3]:

$$\frac{\partial u}{\partial x} \approx \frac{\overline{\Delta u_i}}{\Delta x}, \tag{5}$$

$$\begin{aligned} \overline{\Delta u_i} \;=\; & min(2|u_i - u_{i-1}|, 2|u_{i+1} - u_i|, \\ & \tfrac{1}{2}|u_{i+1} - u_{i-1}|) \cdot sign(u_{i+1} - u_{i-1}), \\ & \text{if } (u_i - u_{i-1})(u_{i+1} - u_i) > 0, \\ 0 \qquad & \text{otherwise} \end{aligned} \tag{6}$$

We would like to develop a similar algorithm for the irregular mesh. The CFL condition should be independent of the size of small cells, even as they become arbitrarily small. This implies, however, that when the characteristics in a small cell are traced back from time level $n + 1/2$, they could reach into the neighboring zones, violating the CFL condition. In order to ensure stability everywhere, we compute the fluxes "implicitly" in the small cells, and explicitly elsewhere.

Let cell 0 be the small cell of size h, and all other cells are of size Δx, as shown in figure 1. According to the idea above, we compute the Taylor expansion implicitly in cell 0. At edge 1/2, we only need to compute $u_{1/2,L}$ implicitly, because it is the only state that comes from the small cell:

$$u^{n+1/2}_{1/2,L} = u^{n+1}_0 + \tfrac{1}{2}(h + \Delta t \frac{\partial f(u_0)}{\partial u})\frac{\partial u}{\partial x} \tag{7}$$

Instead of expanding $u_{1/2}^{n+1/2}$ about the cell averages at time level n, we expand about the cell averages at time level $n+1$. Similarly, at edge $-1/2$, we only need to compute $u_{-1/2,R}$ implicitly:

$$u_{-1/2,R}^{n+1/2} = u_0^{n+1} - \tfrac{1}{2}(h - \Delta t \frac{\partial f(u_0)}{\partial u})\frac{\partial u}{\partial x} \tag{8}$$

Assuming we know how to computed the limited slopes in cells near the small cell, we can compute these two states. Again, we can solve a Riemann problem at each edge on the whole grid. To update cells away from the small one, use (2); to update cells -1, 0, 1, the only cells affected by the implicit flux calculation, we take a mixture of the implicit and the explicit fluxes:

$$\begin{aligned}
u_{-1}^{n+1} &= u_{-1}^{n} - \frac{\Delta t}{\Delta x}((1-\tfrac{h}{\Delta x})(f_{-1/2}^{n+1} - f_{-3/2}^{n+1}) + \tfrac{h}{\Delta x}(f_{-1/2}^{n} - f_{-3/2}^{n}) && (9)\\
u_{0}^{n+1} &= u_{0}^{n} - \frac{\Delta t}{h}((1-\tfrac{h}{\Delta x})(f_{1/2}^{n+1} - f_{-1/2}^{n+1}) + \tfrac{h}{\Delta x}(f_{1/2}^{n} - f_{-1/2}^{n}) && (10)\\
u_{1}^{n+1} &= u_{1}^{n} - \frac{\Delta t}{\Delta x}((1-\tfrac{h}{\Delta x})(f_{3/2}^{n+1} - f_{1/2}^{n+1}) + \tfrac{h}{\Delta x}(f_{3/2}^{n} - f_{1/2}^{n}) && (11)
\end{aligned}$$

where f^{n+1} is the flux calculated implicitly, and f^n is the flux calculated explicitly. Where the implicit flux is not calculated by (7) and (8), use the explicit flux. Note that if $h = \Delta x$,(10) – (11) are reduced to (2). This is an implicit scheme, and so we need to update cells -1, 0, 1 by some iterative process. The iterative process will be discussed in section 2.2.

We now describe the slope limiting process. The slopes in cells -1,0, and 1 may involve cells of different sizes. According to the van Leer limiter, twice the forward or backward differences should not exceed the centered difference. Thus we compute the forward and backward differences in each of the three cells, and scale them so that the differences divided by Δx approximates the derivative to the first order. The centered differences in cell j are obtained by taking the derivative of the interpolated polynomial through the values in cells $j-1$, j, $j+1$. Again, scale the centered difference so that when divided by Δx it approximates the derivative to the second order. The slopes in the three cells are:

$$\frac{\partial u}{\partial x} \approx \frac{\overline{\Delta u_i}}{\Delta x}, i = -1, 0, 1 \tag{12}$$

$$\begin{aligned}
\overline{\Delta u_i} &= min(2|\Delta_+ u_i|, 2|\Delta_- u_i|, |\Delta_0 u_i|)\cdot sign(\Delta_0 u_i), && (13)\\
&\qquad \text{if } (\Delta_+ u_i)(\Delta_- u_i) > 0 \\
&0 \qquad \text{otherwise}
\end{aligned}$$

where for $i = -1$,

$$\Delta_+ u_i = 2(u_{i+1} - u_i)/(1 + \tfrac{h}{\Delta x}); \tag{14}$$

$$
\begin{aligned}
\Delta_- u_i &= (u_i - u_{i-1}); && (15)\\
\Delta_0 u_i &= \Delta_+ u_i - (\Delta_+ u_i - \Delta_- u_i)\frac{1+\frac{h}{\Delta x}}{3+\frac{h}{\Delta x}}; && (16)
\end{aligned}
$$

for $i = 0$,

$$
\begin{aligned}
\Delta_+ u_i &= 2(u_{i+1} - u_i)/(1+\tfrac{h}{\Delta x}); && (17)\\
\Delta_- u_i &= 2(u_i - u_{i-1})/(1+\tfrac{h}{\Delta x}); && (18)\\
\Delta_0 u_i &= \tfrac{1}{2}(\Delta_+ u_i + \Delta_- u_i); && (19)
\end{aligned}
$$

and for $i = 1$,

$$
\begin{aligned}
\Delta_+ u_i &= (u_{i+1} - u_i)/(1+\tfrac{h}{\Delta x}); && (20)\\
\Delta_- u_i &= 2(u_i - u_{i-1})/(1+\tfrac{h}{\Delta x}); && (21)\\
\Delta_0 u_i &= \Delta_- u_i + (\Delta_+ u_i - \Delta_- u_i)\frac{1+\frac{h}{\Delta x}}{3+\frac{h}{\Delta x}}. && (22)
\end{aligned}
$$

2.2 The Iterative Method

As mentioned above, an iterative procedure is needed for (10) – (11). One could simply iterate on u^{n+1}:

$$
\begin{aligned}
u_{-1}^{n+1,i+1} &= u_{-1}^n - \frac{\Delta t}{\Delta x}((1-\tfrac{h}{\Delta x})(f_{-1/2}^{n+1,i} - f_{-3/2}^{n+1,i}) + \tfrac{h}{\Delta x}(f_{-1/2}^n - f_{-3/2}^n)\\
u_0^{n+1,i+1} &= u_0^n - \frac{\Delta t}{h}((1-\tfrac{h}{\Delta x})(f_{1/2}^{n+1,i} - f_{-1/2}^{n+1,i}) + \tfrac{h}{\Delta x}(f_{1/2}^n - f_{-1/2}^n)\\
u_1^{n+1,i+1} &= u_1^n - \frac{\Delta t}{\Delta x}((1-\tfrac{h}{\Delta x})(f_{3/2}^{n+1,i} - f_{1/2}^{n+1,i}) + \tfrac{h}{\Delta x}(f_{3/2}^n - f_{1/2}^n)
\end{aligned}
$$

When $\frac{h}{\Delta x}$is small, the iterations may converge very slowly. A relaxation technique would speed up the iterative processes. A relaxation technique being used at this moment is

$$
\begin{aligned}
u^{*,i} &= (1-\alpha)u^{n+1,i} + \alpha u^{*,i-1} && (23)\\
u^{n+1,i+1} &= F(u^{*,i}) && (24)
\end{aligned}
$$

for $u = u_{-1}$, u_0, u_1. At this moment the radius of convergence of this technique cannot be calculated analytically. The iterative process stops when the difference of u_0 from two consecutive iterations reaches below a tolerance level, or when the number of iterations becomes too large.

3 Numerical Examples

We have implemented this implicit flux method for the advection equation and the inviscid Burger's equation. For the advection equation, we have also analyzed the convergence rate of the relaxation technique in the absence of slopes.

3.1 The Advection Equation

We solve

$$u_t + au_x = 0, \tag{25}$$

for any $a \neq 0$. If $a > 0$, the implicit method would be to trace characteristics exactly, since the solution u_j at any time would only depend on u in the cells left of the jth cell. Thus we are interested in the case $a < 0$.

In our example, we take $a = -1$, and the initial condition

$$u(x,0) = \begin{cases} 1.0 & \text{if } 0.625 \leq x \leq 0.875 \\ 0.1 & \text{otherwise} \end{cases}$$

We place a small cell near the center of the interval $[0,1]$. We have made runs for $\frac{h}{\Delta x} = 0.05, 0.25, 0.375, 0.5, 0.625, 0.75, 1.0$, with 64 points, at a CFL condition of 0.8. While the solutions with different aspect ratios do not differ much, the numbers of iterations taken to convergence are significantly different. In particular, the smaller the $\frac{h}{\Delta x}$, the more iterations it takes. For each $\frac{h}{\Delta x}$, we have made runs using different relaxation parameters α. For these runs, we have also simplified the underlying algorithm to a first-order one. In figure 2, we have plotted the (average) ratios of the differences between the convergent solution and the iterates. The approximated average ratios are plotted because the iteration scheme seems to be sensitive to the initial guesses. The ratios for each α are about the same on the average, indicating that the rate of convergence is approximately first-order, that is, $e^{n+1} = Ce^n$. For each $\frac{h}{\Delta x}$, there is an "optimal" α. At this value of α, the number of iterations is a minimum. For α less than this optimal value the ratios oscillate about some average value then diverge. This oscillating behavior is not unlike that of the iterates in Newton's method for finding roots to nonlinear equations, if one of the iterates comes close an extremum. For α greater than the optimal value, the ratios decrease to a minimum; we have used this minimum in our plot. We suspect that the first guess in the iterations plays a significant role for the oscillatory behavior; however, the location of the optimal relaxation parameter is independent of the guesses. At $\alpha = 1$, the iterations converge for $\frac{h}{\Delta x} \geq 0.5$; the iterations converge to the wrong solution for $\frac{h}{\Delta x} < 0.5$. For all $\frac{h}{\Delta x}$, the iterative procedure does not converge within 40 iterations when $\alpha > 1$. We have found that the accuracy is independent of the relaxation parameters.

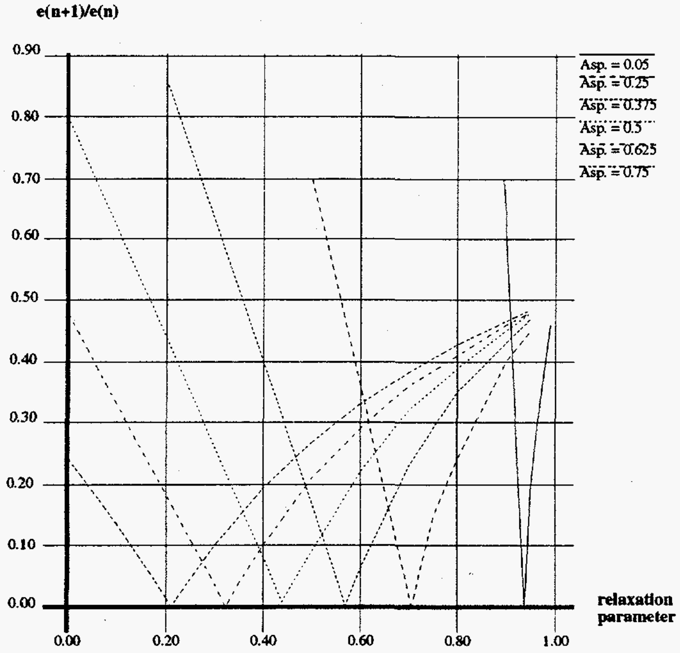

Figure 2: α vs. C

3.2 Burgers Equation

We solve the inviscid Burgers equation

$$u_t + uu_x = 0. \tag{26}$$

Here $f(u) = \frac{1}{2}u^2$ is the flux function. Again, we put the small cell j_{small} near the center of the interval $[0, 1]$. and we study a rarefaction wave with initial conditions

$$u(x_j, 0) = \begin{cases} -1.0 & \text{if } j < j_{small} + 1 \\ 1.0 & \text{otherwise} \end{cases}$$

We have made runs using 64 points, with $\frac{h}{\Delta x}$= 1.0, 0.05, 0.25, 0.5, 0.75. For $\frac{h}{\Delta x} \neq 1$, we the solutions are shifted (see figures 3 – 6). Convergence studies show that the method is actually first-order (see tables 1 – 5), even with small cells.

No. of Pts	$\|\|e\|\|_2$	$\|\|e\|\|_\infty$
64	3.2×10^{-3}	7.4×10^{-3}
128	1.6×10^{-3}	3.7×10^{-3}
256	$.80 \times 10^{-3}$	1.9×10^{-3}

Table 1: Errors at $t = 0.625$ for $\frac{h}{\Delta x} = 0.05$.

No. of Pts	$\|\|e\|\|_2$	$\|\|e\|\|_\infty$
64	2.7×10^{-3}	7.4×10^{-3}
128	1.4×10^{-3}	3.7×10^{-3}
256	$.68 \times 10^{-3}$	1.9×10^{-3}

Table 2: Errors at $t = 0.625$ for $\frac{h}{\Delta x} = 0.25$.

No. of Pts	$\|\|e\|\|_2$	$\|\|e\|\|_\infty$
64	2.9×10^{-3}	7.6×10^{-3}
128	1.5×10^{-3}	3.8×10^{-3}
256	$.74 \times 10^{-3}$	1.9×10^{-3}

Table 3: Errors at $t = 0.625$ for $\frac{h}{\Delta x} = 0.5$.

No. of Pts	$\|\|e\|\|_2$	$\|\|e\|\|_\infty$
64	2.8×10^{-3}	7.2×10^{-3}
128	1.5×10^{-3}	3.7×10^{-3}
256	$.74 \times 10^{-3}$	1.9×10^{-3}

Table 4: Errors at $t = 0.625$ for $\frac{h}{\Delta x} = 0.75$.

No. of Pts	$\|\|e\|\|_2$	$\|\|e\|\|_\infty$
64	$1.9 \times 10_{-3}$	$4.4 \times 10_{-3}$
128	$.99 \times 10_{-3}$	$2.2 \times 10_{-3}$
256	$.50 \times 10_{-3}$	$1.1 \times 10_{-3}$

Table 5: Errors at $t = 0.625$ for $\frac{h}{\Delta x} = 1$.

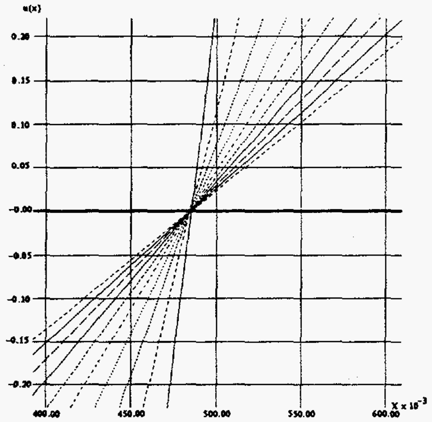

Figure 3: A magnification of the solution for $\frac{h}{\Delta x} = 0.05$

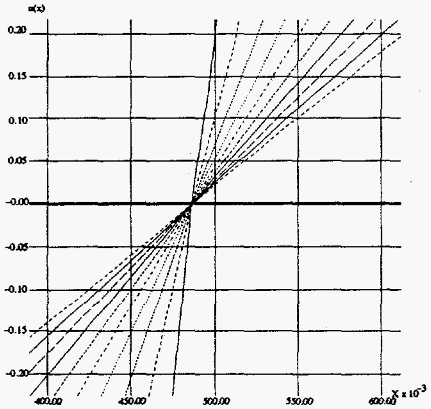

Figure 4: A magnification of the solution for $\frac{h}{\Delta x} = 0.25$

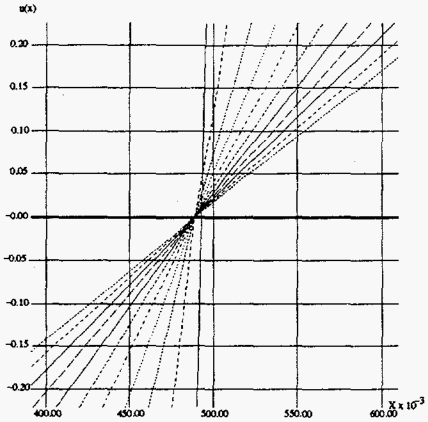

Figure 5: A magnification of the solution for $\frac{h}{\Delta x} = 0.50$

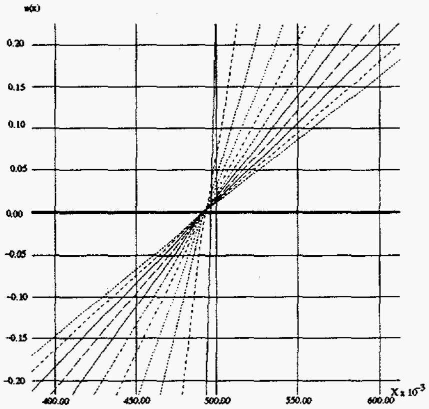

Figure 6: A magnification of the solution for $\frac{h}{\Delta x} = 0.75$

4 Future Work

We would like to implement this algorithm for systems of conservation laws in one space dimension. We would also like to implement it for two or three space dimensions. Two problems we may have in higher dimensions are the calculation of the transverse flux (if we use the unsplit method in [4]) and the complex geometries. The latter could complicate the iterative procedure.

References

[1] M. Berger and R. LeVeque, *Cartesian Meshes and Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations* Proc. 3rd Intl. Conf. Hyperbolic Problems, Uppsala, Sweden, June 1990

[2] M. Berger and R. LeVeque, *A Rotated Difference Scheme for Cartesian Grids in Complex Geometries*, AIAA paper CP-91-1602

[3] B. van Leer, *J. Comp. Phys.*, **32**, 101 (1979)

[4] P. Colella, *J. Comp. Phys.*, **87**, 171 (1990)