JOURNAL OF COMPUTATIONAL PHYSICS 86, 1–17 (1990)

Calculation of Shocked One-Dimensional Flows on Abruptly Changing Grids by Mathematical Programming

JOHN E. LAVERY†

NASA Lewis Research Center,
Cleveland, Ohio 44135

Received February 26, 1988; revised January 24, 1989

The steady-state inviscid and nearly inviscid Burgers' equations and the Euler equations for steady-state shocked flow in a quasi-1-dimensional nozzle are discretized by cell-centered finite differences on the cells of an arbitrarily spaced grid. Inviscid and viscous terms are approximated to second order by 2-point schemes and 4-point schemes, respectively. There results an overdetermined system of nonlinear algebraic equations. This system is solved by a mathematical-programming procedure that minimizes a weighted sum of the absolute values of the residuals (the l_1 norm of the vector of residuals). In this algorithm, which is nonconservative, no upwinding, switches, arbitrary constants, or heuristic quantities are used. The artificial viscosity used to solve the inviscid Burgers' equation is small ($-10^{-15}u''$). No artificial viscosity of any kind is used for the Euler equations. The numerical solutions of both the viscous and the inviscid problems are accurate and nonoscillatory on grids with abrupt refinements in mesh length by factors as high as 10^4. Shocks are invariably captured in one cell and this cell is rarely more than three cells away from the cell in which the physical shock occurs.

1. INTRODUCTION

Solution of the steady-state Burgers' equation on equally spaced grids by mathematical programming has been investigated in [5]. In the present paper, mathematical programming is used to solve Burgers' equation and the Euler equations for quasi-1-dimensional flows on equally spaced grids and on grids with abrupt changes in mesh length by factors as high as 10^4. All of the computational results presented in this paper were obtained in double-precision arithmetic on an IBM 370.

A summary of the results of this paper was presented at the "11th International Conference on Numerical Methods in Fluid Dynamics, Williamsburg, Virginia, June 26–July 1, 1988" [6].

2. BURGERS' EQUATION

We consider first the steady-state inviscid Burgers' equation

$$(u^2)' = 0 \qquad \text{on} \quad (0, 1) \tag{2.1a}$$

† Present address: Mathematical Sciences Division, Office of Naval Research, Arlington, VA 22217-5000.

0021-9991/90 $3.00

with the boundary conditions

$$u(0)=g_0, \qquad u(1)=g_1 \qquad (g_0 \neq g_1). \tag{2.1b}$$

The physically relevant solution of (2.1) is the pointwise limit as $\varepsilon \to 0^+$ of the solution of the singularly perturbed problem

$$-\varepsilon u'' + (u^2)' = 0 \qquad \text{on } (0, 1) \tag{2.2a}$$

$$u(0)=g_0, \qquad u(1)=g_1. \tag{2.2b}$$

For the boundary conditions

$$g_0 = 1, \qquad g_1 = 0, \tag{2.3}$$

the solution of the viscous problem (2.2) is

$$u(x) = c \tanh(c(1-x)/\varepsilon) \tag{2.4}$$

for c such that $c \tanh(c/\varepsilon) = 1$ (boundary layer at $x = 1^-$ for small ε). For boundary conditions (2.3), the physically relevant solution of the inviscid problem (2.1) is

$$u(x) = \begin{cases} 1, & 0 \leqslant x < 1, \\ 0, & x = 1. \end{cases} \tag{2.5}$$

Let x_i, $i = 0, 1, \ldots, n$, be the node points, not necessarily equally spaced, of a grid on $[0, 1]$:

$$0 = x_0 < x_1 < x_2 < \cdots < x_{n-1} < x_n = 1. \tag{2.6}$$

Discretize Eq. (2.2a) on each cell (x_i, x_{i+1}) using a 4-point finite-difference scheme involving u_{i-1}, u_i, u_{i+1}, and u_{i+2} for the viscous term $-\varepsilon u''$ and a 2-point difference scheme involving only u_i and u_{i+1} for the inviscid term $(u^2)'$, both schemes being of second order with respect to maximum cell width at the midpoint $(x_i + x_{i+1})/2$ of the cell. The resulting finite-difference approximations of (2.2a) on the first cell (x_0, x_1), on the interior cells (x_i, x_{i+1}), $i = 1, 2, \ldots, n-2$, and on the last cell (x_{n-1}, x_n) are, respectively,

$$\varepsilon \left[\begin{aligned} &\frac{3x_0 + x_1 - 2x_2 - 2x_3}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)} g_0 \\ &+ \frac{x_0 + 3x_1 - 2x_2 - 2x_3}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3)} u_1 \\ &+ \frac{x_0 + x_1 - 2x_3}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)} u_2 \\ &+ \frac{x_0 + x_1 - 2x_2}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)} u_3 \end{aligned} \right] + \frac{u_1^2 - g_0^2}{x_1 - x_0} = 0, \tag{2.7a}$$

$$\varepsilon\left[\begin{array}{l}\dfrac{x_i+x_{i+1}-2x_{i+2}}{(x_{i-1}-x_i)(x_{i-1}-x_{i+1})(x_{i-1}-x_{i+2})}u_{i-1}\\ +\dfrac{-2x_{i-1}+3x_i+x_{i+1}-2x_{i+2}}{(x_i-x_{i-1})(x_i-x_{i+1})(x_i-x_{i+2})}u_i\\ +\dfrac{-2x_{i-1}+x_i+3x_{i+1}-2x_{i+2}}{(x_{i+1}-x_{i-1})(x_{i+1}-x_i)(x_{i+1}-x_{i+2})}u_{i+1}\\ +\dfrac{-2x_{i-1}+x_i+x_{i+1}}{(x_{i+2}-x_{i-1})(x_{i+2}-x_i)(x_{i+2}-x_{i+1})}u_{i+2}\end{array}\right]+\frac{u_{i+1}^2-u_i^2}{x_{i+1}-x_i}=0, \quad (2.7b)$$

$$\varepsilon\left[\begin{array}{l}\dfrac{-2x_{n-2}+x_{n-1}+x_n}{(x_{n-3}-x_{n-2})(x_{n-3}-x_{n-1})(x_{n-3}-x_n)}u_{n-3}\\ +\dfrac{-2x_{n-3}+x_{n-1}+x_n}{(x_{n-2}-x_{n-3})(x_{n-2}-x_{n-1})(x_{n-3}-x_n)}u_{n-2}\\ +\dfrac{-2x_{n-3}-2x_{n-2}+3x_{n-1}+x_n}{(x_{n-1}-x_{n-3})(x_{n-1}-x_{n-2})(x_{n-1}-x_n)}u_{n-1}\\ +\dfrac{-2x_{n-3}-2x_{n-2}+x_{n-1}+3x_n}{(x_n-x_{n-3})(x_n-x_{n-2})(x_n-x_{n-1})}g_1\end{array}\right]+\frac{g_1^2-u_{n-1}^2}{x_n-x_{n-1}}=0. \quad (2.7c)$$

For equally spaced x_i, Eqs. (2.7) reduce to Eqs. (2.1) of [5]. In particular, Eq. (2.7b) reduces to the more familiar form

$$\varepsilon\frac{-u_{i-1}+u_i+u_{i+1}-u_{i+2}}{2h^2}+\frac{u_{i+1}^2-u_i^2}{h}=0 \quad (2.8)$$

$(h=x_{i+1}-x_i)$.

Equations (2.7) are an overdetermined system of n equations for the $n-1$ unknowns u_i, $i=1, 2, ..., n-1$. Why this system should be solved by a l_1 procedure rather than by a l_2 (least-square) procedure is discussed in [5]. The l_1 strategy for solving system (2.7) consists in finding the u_i, $i=1, 2, ..., n-1$ that minimize the weighted sum

$$\sum_{i=0}^{n-1}(x_{i+1}-x_i)|r_i|, \quad (2.9)$$

of the absolute values of the residuals r_i (left sides) of Eqs. (2.7). In [5], the weights $(x_{i+1}-x_i)$ were all equal and were omitted. Sum (2.9) was minimized by the procedure used in [5]: system (2.7) (with the equations reweighted by multiplication by $(x_{i+1}-x_i)$) was linearized by Newton's method, the resulting overdetermined linear system was solved by the Barrodale–Roberts l_1 algorithm [1, 2, 3] as

implemented in the IMSL (Ed. 9) subroutine RLLAV and these two steps were repeated. Convergence was deemed to have occured when the relative l_1 error

$$\sum_{i=1}^{n-1} \frac{|u_i^{\text{cur}} - u_i^{\text{pre}}|}{|u_i^{\text{cur}}|} \tag{2.10}$$

between the current solution values u_i^{cur} and the values u_i^{pre} on the previous step was less than $0.5 * 10^{-10}$.

All of the computational results for Burgers' equation were obtained using the l_1 procedure with a homotopy (continuation procedure) in ε. For $\varepsilon = 10^{-1}$, the initial guesses for the u_i were set equal to $1 - x_i$. Once convergence for $\varepsilon = 10^{-1}$ was achieved (as determined by the convergence criterion involving (2.10) discussed above), the final solution for $\varepsilon = 10^{-1}$ was used as the initial guess for the solution of the problem with $\varepsilon = 10^{-2}$. In general, the solution for $\varepsilon = 10^{-k+1}$ was used as the initial guess for the solution of the problem with $\varepsilon = 10^{-k}$, $k = 2, 3, \ldots, 15$. The results presented in [5] show how well the l_1 procedure performs on equally spaced grids. Additional numerical experiments on equally spaced grids for system (2.7), (2.3) with $n = 4, 8, 16, 32$, and 64 cells were performed. These results suggest that the estimate $O((\varepsilon/h)^m)$ of [5] for the difference between the numerical solution u_i and the inviscid solution $u \equiv 1$ can be sharpened slightly to $O((\varepsilon/4h)^m)$. Here, the integer m represents the number of mesh units between the node in question and the position of the boundary or interior layer, rounded upwards to the nearest integer. The layer here is the boundary layer located at $x = 1^-$. The results for $n = 4$ confirm that the l_1 procedure produces the correct inviscid solution even on extremely coarse grids. The numerical results of [5] and the new numerical results mentioned here show that the l_1 procedure not only produces accurate non-oscillatory approximations of the solution of the inviscid problem (2.1), (2.3) as $\varepsilon \to 0$ but also produces good approximations of the solutions of the viscous problem (2.2), (2.3). Indeed, for all of the results mentioned above, the numerical solution u_i was within 3% of the theoretical viscous solution $u(x_i)$ in the boundary layer and was much closer than that outside the boundary layer.

While the l_1 procedure is known to perform well on equally spaced grids, its performance on grids with variable spacing remains to be investigated. A particularly interesting case is that of achieving boundary-layer resolution by concentrating node points near the boundary. With standard methods, node points can be concentrated in the boundary layer only if the grid is gradually stretched. This results in having many extra nodes in the inviscid region outside the boundary layer, since the stretching factor is usually much less than a factor of 2. If performance of the standard methods were not so degraded by abrupt changes in mesh length, one would often choose to have a very fine grid in the boundary layer and a coarse grid outside the boundary layer with the transition between the two being quite abrupt. This would result in savings in storage as well as in the amount of programming and CPU time necessary to generate the grid and compute the solution. It is on grids of this type that the l_1 procedure was put to the test. Each grid consisted of

two sets of nodes: nodes with coarse equal spacing outside the boundary layer at $x = 1^-$ and nodes with fine equal spacing in the boundary layer, the coarse cells being larger than the fine cells by factors as high as 10^4. The following grids with factors of 10^r, $r = 1, 2, 3, 4$, were used:

$$\begin{aligned} &7 \text{ large cells each of width } \frac{10^r}{7 * 10^r + 10} \text{ on } \left(0, \frac{7 * 10^r}{7 * 10^r + 10}\right), \\ &10 \text{ small cells each of width } \frac{1}{7 * 10^r + 10} \text{ on } \left(\frac{7 * 10^r}{7 * 10^r + 10}, 1\right). \end{aligned} \tag{2.11}$$

Convergence occured for all four grids for ε up to and including 10^{-15} in seven or fewer Newton iterations for each ε. In the boundary layer, the difference between the numerical solution u_i and the inviscid solution 1 was $O((\varepsilon/4h)^m)$. Here, h denotes the width of each of the fine cells in the boundary layer and m denotes the distance in units of h from $x = 1^-$ to the node under consideration, rounded upwards to the nearest integer ($m \leqslant 8$). Selected u_i are compared in Table I to the values of the viscous solution (2.4) for the most extreme of the cases, $r = 10^4$ (abrupt refinement in mesh length by a factor of 10^4 at $x = 7000/7001$). The l_1 procedure not only produces numerical solutions that converge to the inviscid solution

TABLE I

Comparison of the l_1 Solution u_i of System (2.7), (2.3) for Grid (2.11) with $r = 4$ (Abrupt Refinement in Mesh Length by a Factor of 10^4 in the Boundary Layer) with the Solution $u(x_i)$ of the Viscous Problem (2.2), (2.3)

		$x_6 = \frac{6000}{7001}$	$x_7 = \frac{7000}{7001}$	$x_8 = \frac{70001}{70010}$	$x_9 = \frac{70002}{70010}$	$x_{16} = \frac{70009}{70010}$
$\varepsilon = 10^{-1}$	u_i	0.747244	0.001404	0.001263	0.001123	0.000140
	$u(x_i)$	0.891625	0.001428	0.001286	0.001143	0.000143
$\varepsilon = 10^{-2}$	u_i	0.976638	0.010163	0.009147	0.008131	0.001016
	$u(x_i)$	1.000000	0.014283	0.012855	0.011426	0.001428
$\varepsilon = 10^{-3}$	u_i	0.997888	0.095725	0.086228	0.076706	0.009616
	$u(x_i)$	1.000000	0.141873	0.127850	0.113775	0.014283
$\varepsilon = 10^{-4}$	u_i	0.999954	0.801083	0.766947	0.725164	0.121880
	$u(x_i)$	1.000000	0.891332	0.857952	0.815319	0.141873
$\varepsilon = 10^{-5}$	u_i	1.000000	1.000000	1.000000	0.999999	0.813168
	$u(x_i)$	1.000000	1.000000	1.000000	1.000000	0.891332
$\varepsilon = 10^{-6}$	u_i	1.000000	1.000000	1.000000	1.000000	0.982347
	$u(x_i)$	1.000000	1.000000	1.000000	1.000000	1.000000
$\varepsilon = 10^{-7}$	u_i	1.000000	1.000000	1.000000	1.000000	0.998248
	$u(x_i)$	1.000000	1.000000	1.000000	1.000000	1.000000

Note. The entries in the table are the values of the u_i and $u(x_i)$.

1 but also produces good approximations of the viscous solutions $u(x_i)$: the maximum relative error $|u_i - u(x_i)|/|u(x_i)|$ in the data of Table I is 16%, 29%, 33%, 14%, 9%, 2%, and 0.2% for $\varepsilon = 10^{-k}$, $k = 1, 2, 3, 4, 5, 6, 7$, respectively. Given the extreme abruptness of the change from the coarse cells to the fine cells, this is remarkable agreement. Results for grids (2.11) with less abrupt changes in mesh length ($r = 3, 2, 1$ for factors of 1000, 100, and 10, respectively) showed even better approximation of the viscous solution than that mentioned for $r = 4$ just above and showed analogous convergence to the inviscid solution.

Abrupt changes in mesh length can arise in practice not only when a refined grid is used near a boundary or interior layer but also when grids generated for separate parts of a domain meet. The l_1 procedure was tested on four grids that simulate this situation. Each grid had eight cells with one small cell just to the left of $x = 0.5$ sandwiched between three large cells on the left and four large cells on the right. The four grids used were

$$\begin{gathered}\text{3 large cells each of width } \frac{0.5 - 0.125 * 10^{-r}}{3} \text{ in } (0,\ 0.5 - 0.125 * 10^{-r}),\\ \text{1 small cell of width } 0.125 * 10^{-r} \text{ in } (0.5 - 0.125 * 10^{-r},\ 0.5)\\ \text{4 large cells each of width } 0.125 \text{ in } (0.5,\ 1),\end{gathered} \tag{2.12}$$

$r = 1, 2, 3, 4$. The widths of the three types of cells in (2.12) are in the ratios

$$\tfrac{4}{3} * 10^{r} - 1 : 1 : 10^{r}. \tag{2.13}$$

Selected results for $r = 2$, for which the mesh length changes by factors of 132 and 100 to the left and right, respectively, of the small cell, are presented in Table II. These results show that the l_1 procedure does not yield a particularly good numeri-

TABLE II

Comparison of the l_1 Solution u_i of System (2.7), (2.3) for Grid (2.12) (Cell Widths in the Ratio 132: 1: 100) with the Solution $u(x_i)$ of the Viscous Problem (2.2), (2.3)

		$x_2 = 0.33250$	$x_3 = 0.49875$	$x_4 = 0.5$	$x_5 = 0.625$	$x_6 = 0.75$
$\varepsilon = 10^{-1}$	u_i	1.018740	1.180288	1.184322	0.648835	0.745356
	$u(x_i)$	0.999997	0.999911	0.999909	0.998894	0.986614
$\varepsilon = 10^{-2}$	u_i	1.000000	1.000000	1.000000	0.999996	0.999600
	$u(x_i)$	1.000000	1.000000	1.000000	1.000000	1.000000
$\varepsilon = 10^{-3}$	u_i	1.000000	1.000000	1.000000	1.000000	0.999996
	$u(x_i)$	1.000000	1.000000	1.000000	1.000000	1.000000
$\varepsilon = 10^{-4}$	u_i	1.000000	1.000000	1.000000	1.000000	1.000000
	$u(x_i)$	1.000000	1.000000	1.000000	1.000000	1.000000

Note. The entries in the table are the values of the u_i and $u(x_i)$.

cal solution of the viscous problem with $\varepsilon = 10^{-1}$ but does solve the inviscid problem well as $\varepsilon \to 0$. Convergence occurred for $\varepsilon = 10^{-k}$, $k = 1, 2, ..., 15$ (homotopy procedure). At nodes nearest the boundary layer, the $O(\varepsilon^m)$ convergence seen in previous results is observed. Analogous results were obtained for $r = 1$ (less abrupt cell-width ratios of 123 : 1 : 10). For $r = 3, 4$ (more abrupt ratios of 1332 : 1 : 1000 and 13332 : 1 : 10000, respectively), most of the u_i converged to 1 while one or more of the u_i converged to the nonphysical value -1. All of these results for grids (2.12) were obtained with the homotopy procedure starting from $u_i = 1 - x_i$ and $\varepsilon = 10^{-1}$ as described at the beginning of this section. When the initial guess for $r = 3, 4$ was $u_i = 1$ and the homotopy went from $\varepsilon = 10^{-3}$ (instead of $\varepsilon = 10^{-1}$) to $\varepsilon = 10^{-15}$ (stepping through powers of 10), the u_i all remained near 1 and converged back to 1 as $\varepsilon \to 0$.

The results for inviscid and nearly inviscid Burgers' equations suggest that the l_1 procedure is a good candidate for solving steady-state conservation laws. This topic is investigated further in the next section.

3. Euler Equations for Quasi-One-Dimensional Flow

Steady quasi-1-dimensional flow of air in a nozzle of length 10 with cross-sectional area $A(x)$ can be described by

$$\frac{d}{dx}(A\rho u) = 0, \tag{3.1a}$$

$$\frac{d}{dx}(A\rho u^2) + A\frac{dp}{dx} = 0, \tag{3.1b}$$

$$\frac{d}{dx}[Au(\rho E + p)] = 0, \tag{3.1c}$$

on (0, 10), where ρ is the density, u is the velocity, $E = e + (u^2/2)$ is the total energy, e is the internal energy, $p = (\gamma - 1)\rho e$ is the pressure for a perfect gas and $\gamma = 1.4$ (all quantities normalized). The area function

$$A(x) = 1.398 + 0.347 \tanh(0.8x - 4) \tag{3.2}$$

and the boundary conditions

$$\begin{aligned} &\rho(0) = 0.502, \qquad u(0) = 1.299, \qquad e(0) = 1.897, \\ &\rho(10) = 0.776 \end{aligned} \tag{3.3}$$

of [10] were used. The four boundary conditions (3.3) yield a supersonic-subsonic flow with shock at $x = 4.816$.

Let x_i, $i=0, 1, ..., n$, be the node points of a not necessarily equally spaced grid on $[0, 10]$ that satisfies relations (2.6) with $x_n=1$ replaced by $x_n=10$. Equations (3.1) are discretized on this grid using 2-point finite-difference schemes of $O((x_{i+1}-x_i)^2)$ centered at the midpoint of the cell (x_i, x_{i+1}):

$$\frac{A_{i+1}\rho_{i+1}u_{i+1}-A_i\rho_i u_i}{x_{i+1}-x_i}=0, \tag{3.4a}$$

$$\frac{A_{i+1}\rho_{i+1}u_{i+1}^2-A_i\rho_i u_i^2}{x_{i+1}-x_i}+\frac{A_{i+1}+A_i}{2}\frac{p_{i+1}-p_i}{x_{i+1}-x_i}=0, \tag{3.4b}$$

$$\frac{A_{i+1}u_{i+1}(\rho_{i+1}E_{i+1}+p_{i+1})-A_i u_i(\rho_i E_i+p_i)}{x_{i+1}-x_i}=0 \tag{3.4c}$$

(cf. [4, 8, 11]). System (3.4) with the boundary conditions (3.3) is an overdetermined system of $3n$ equations for the $3n-1$ unknowns ρ_i, $(\rho u)_i$, e_i, $i=1, 2, ..., n-1$, and $(\rho u)_n$, e_n. The l_1 strategy for solving this problem is to minimize the sum

$$\sum_{i=0}^{n-1}\sum_{j=1}^{3}(x_{i+1}-x_i)|r_{ij}|, \tag{3.5}$$

where r_{ij} is the residual (left side) of the jth of Eqs. (3.4) for a given i (cf. (2.9)). Sum (3.5) was minimized by the l_1 procedure used in Section 2: system (3.4) with the equations reweighted by multiplication by $(x_{i+1}-x_i)$ was linearized by Newton's method and the resulting overdetermined linear system was solved by the Barrodale–Roberts l_1 algorithm (IMSL Ed. 9 subroutine RLLAV). Convergence was deemed to have occurred when the relative l_1 error

$$\sum_{i=1}^{n-1}\frac{|\rho_i^{\text{cur}}-\rho_i^{\text{pre}}|}{|\rho_i^{\text{cur}}|}+\sum_{i=1}^{n}\frac{|(\rho u)_i^{\text{cur}}-|\rho u)_i^{\text{pre}}|}{|(\rho u)_i^{\text{cur}}|}+\sum_{i=1}^{n}\frac{|e_i^{\text{cur}}-e_i^{\text{pre}}|}{|e_i^{\text{cur}}|} \tag{3.6}$$

between the solution values on the current Newton step (superscript "cur") and the solution values on the previous Newton step (superscript "pre") was less than $0.5 * 10^{-10}$. No homotopy was used since no artificial viscosity was added to Eqs. (3.4). The initial guesses were

$$\begin{gathered}\rho_i=\rho(0) \quad \text{for} \quad x_i<4.816, \qquad \rho_i=\rho(10) \quad \text{for} \quad x_i>4.816,\\ (\rho u)_i=\frac{A(0)\,\rho(0)\,u(0)}{A(x_i)}\ \forall x_i, \qquad e_i=e(0)\ \forall x_i.\end{gathered} \tag{3.7}$$

The choice of initial ρ_i that have a "discontinuity" at the position of the shock in the physically relevant solution of (3.1), (3.3), that is, at $x=4.816$, helps out the convergence of the l_1 procedure. What happens when the initial ρ_i are on a linear function connecting the boundary values $\rho(0)$ and $\rho(10)$ (with the initial $(\rho u)_i$ and e_i as in (3.7)) will be discussed later in this section.

We first present selected results from numerical experiments done on equally spaced grids with $n = 4$, 8, 16, and 32 cells. Results for $n = 64$ were also obtained but are not presented here. The numerical densities ρ_i are compared to the exact densities $\rho(x_i)$ in Table III. (An exact solution of system (3.1), (3.3) can be calculated in implicit form. Newton's method was used on this implicit algebraic equation to obtain the $\rho(x_i)$.) As expected, the largest errors occur near the shock at $x = 4.816$. However, the largest error on the coarsest grid, namely, the error at $x = 5$ for the grid with $n = 4$, is a mere 1.2% (numerical $\rho_i = 0.72487622$ vs exact $\rho(x_i) =$

TABLE III

Comparison of the l_1 Solution ρ_i of System (3.4), (3.3) on Equally Spaced Grids with the Exact Solution $\rho(x_i)$ of System (3.1), (3.3)

x_i	$n=4$ ρ_i	$n=8$ ρ_i	$n=16$ ρ_i	$n=32$ ρ_i	Exact $\rho(x_i)$
0	0.50200000	0.50200000	0.50200000	0.50200000	0.50200000
0.3125				0.50180564	0.50180564
0.6250			0.50148612	0.50148608	0.50148606
0.9375				0.50096156	0.50096151
1.2500		0.50010440	0.50010343	0.50010309	0.50010297
1.5625				0.49870454	0.49870422
1.8750			0.49644520	0.49644289	0.49644204
2.1875				0.49282759	0.49282547
2.5	0.48727452	0.48720550	0.48716393	0.48715000	0.48714495
2.8125				0.47846351	0.47845229
3.1250			0.46571636	0.46565257	0.46562992
3.4375				0.44766209	0.44762167
3.7500		0.42473478	0.42410261	0.42392164	0.42385978
4.0625				0.39488327	0.39480465
4.3750			0.36264995	0.36240318	0.36232317
4.6875				0.32955429	0.32949226
5	0.72487622	0.71953064	0.71722244	0.71660180	0.71639225
5.3125				0.73735974	0.73724317
5.6250			0.75160317	0.75143709	0.75137902
5.9375				0.76064976	0.76062347
6.2500		0.76664319	0.76654967	0.76651889	0.76650781
6.5625				0.77019025	0.77018581
6.8750			0.77246467	0.77245994	0.77245821
7.1875				0.77385278	0.77385212
7.5	0.77470941	0.77470626	0.77470431	0.77470363	0.77470339
7.8125				0.77522195	0.77522186
8.1250			0.77553725	0.77553716	0.77553713
8.4375				0.77572865	0.77572864
8.7500		0.77584496	0.77584492	0.77584491	0.77584490
9.0625				0.77591547	0.77591546
9.3750			0.77595828	0.77595827	0.77595827
9.6875				0.77598425	0.77598425
10	0.77600000	0.77600000	0.77600000	0.77600000	0.77600000

0.71639225). The rate of decrease of the error as $h \to 0$ at a given x is observed to be $O(h^2)$, where $h = x_{i+1} - x_i$.

The results of Table III indicate that the l_1 procedure performs significantly better on equally spaced grids than both the tridiagonal shock-fitting algorithm of [10] and the bidiagonal shock-capturing algorithm of [11]. In Fig. 4 of [10], the exact density function $\rho(x)$ and the densities ρ_i computed by the tridiagonal shock-fitting algorithm on equally spaced grids with $n = 16$ and 32 are plotted. The ρ_i of the l_1 procedure for $n = 16$, which are nonoscillatory and equal to the exact densities $\rho(x_i)$ to within 0.12%, are much better than the oscillatory results of [10] for $n = 16$. The l_1 densities for $n = 32$ are equal to the exact densities to within 0.03% but cannot be compared to the densities of the tridiagonal shock-fitting algorithm for $n = 32$ since no tables of computed values are given in [10]. In Fig. 8a of [11], the mach numbers m_i (= the u_i of the present paper) computed by the bidiagonal shock-capturing algorithm on an equally spaced grid with $n = 64$ are plotted. In this figure, three cells are required to make the transition from one side of the shock to the other. The l_1 procedure performs much better: the transition over the shock is completed in the single cell (4.53125, 4.6875), which is the left neighbor of the cell containing the physical shock. Whether the l_1 mach numbers, which except for the u_i at $x_i = 4.6875$ are equal to the theoretical values to within 0.013%, or the mach numbers of [11] are more accurate on the smooth portions of the solutions is not clear since the magnitude of the errors is not stated in [11].

The nozzle equations were investigated also on the following grids with abrupt refinements near the shock:

$$\begin{gathered} \text{3 large cells each of width 1.25 in (0, 3.75),} \\ \text{10 small cells each of width 0.125 in (3.75, 5),} \\ \text{4 large cells each of width 125 in (5, 10).} \end{gathered} \tag{3.8}$$

This grid includes all of the node points of the equally spaced grid with $n = 8$ as well as the node points that are generated by dividing the cell (3.75, 5) into 10 equal small cells. The l_1 procedure produced an accurate nonoscillatory solution with a crisp shock. The position of the shock was, however, in error by one cell—the numerical solution had a shock in (4.625, 4.75) rather than in (4.75, 4.875). Not counting the error at the node point $x = 4.75$, which was large because of the error in the shock position, the largest error of the numerical solution vs the exact theoretical solution was less than 0.5%. At the node points of the equally spaced grid with $n = 8$, the values of the numerical solution for grid (3.8) coincided to 15 decimal digits with the values of the numerical solution for the equally spaced grid with $n = 8$ (arithmetic carried out with 14 hexadecimal digits, equivalent to 16.8 decimal digits). A table of numerical results will not be presented for this case since results for a grid with even greater local refinement are presented below.

TABLE IVa

Comparison of the l_1 Solution ρ_i of System (3.4), (3.3) on Grid (3.9) with the Exact Solution $\rho(x_i)$ of System (3.1), (3.3)

x_i	ρ_i	$\rho(x_i)$	Error
0	0.50200000	0.50200000	0
1.6	0.49849216	0.49848625	0.00000591
3.2	0.46229173	0.46181430	0.00047743
4.8	0.32024257	0.31824788	0.00199469
4.81	0.31925053	0.31727225	0.00197828
4.82	0.70456939	0.70054954	0.00401985
4.83	0.70551894	0.70151339	0.00400555
4.84	0.70645849	0.70246693	0.00399156
4.85	0.70738815	0.70341025	0.00397790
4.86	0.70830798	0.70434344	0.00396454
4.87	0.70921808	0.70526659	0.00395149
4.88	0.71011852	0.70617980	0.00393872
4.89	0.71100937	0.70708315	0.00392622
4.9	0.71189072	0.70797672	0.00391400
6.175	0.76550749	0.76533395	0.00017354
7.45	0.77459730	0.77459385	0.00000345
8.725	0.77583765	0.77583759	0.00000006
10	0.77600000	0.77600000	0

TABLE IVb

Comparison of the Numerical Pressures p_i for the l_1 Solution of System (3.4), (3.3) on Grid (3.9) with the Exact Pressures $p(x_i)$ for System (3.1), (3.3)

x_i	p_i	$p(x_i)$	Error
0	0.38091760	0.38091760	0
1.6	0.37719746	0.37719011	0.00000735
3.2	0.33952653	0.33892231	0.00060422
4.8	0.20396442	0.20124151	0.00272291
4.81	0.20308040	0.20037833	0.00270207
4.82	0.65206289	0.64774290	0.00431999
4.83	0.65329352	0.64899092	0.00430260
4.84	0.65451185	0.65022626	0.00428559
4.85	0.65571798	0.65144902	0.00426896
4.86	0.65691200	0.65265930	0.00425270
4.87	0.65809399	0.65385719	0.00423680
4.88	0.65926403	0.65504280	0.00422123
4.89	0.66042220	0.65621620	0.00420600
4.9	0.66156859	0.65737749	0.00419110
6.175	0.73330850	0.73312771	0.00018079
7.45	0.74557960	0.74557602	0.00000358
8.725	0.74725262	0.74725256	0.00000006
10	0.74747157	0.74747157	0

The l_1 procedure was put to a more stringent test by solving problem (3.4), (3.3) on the grid consisting of

$$\begin{gathered} \text{3 large cells each of with 1.6 in (0, 4.8),} \\ \text{10 small cells each of width 0.01 in (4.8, 4.9),} \\ \text{4 large cells each of width 1.275 in (4.9, 10),} \end{gathered} \tag{3.9}$$

for which the mesh length changes abruptly by a factor of 160 at $x=4.8$ and by a factor of 127.5 at $x=4.9$. The numerical densities ρ_i are compared to the exact densities $\rho(x_i)$ in Table IVa. The numerical pressures p_i are compared to the exact pressures $p(x_i)$ in Table IVb. The shock is located in the correct cell (4.81, 4.82), the numerical solution is nonoscillatory, the maximum density error is less than 0.6% and the maximum pressure error is less than 1.4%.

The accuracy of the numerical solutions mentioned above in this section suggests that the numerical solution in the vicinity of the shock might be efficiently calculated by first solving the Euler equations globally on a coarse grid and then solving the Euler equations locally on a fine grid imposed on the coarse-grid cell that contains the shock. This was done for several cases with the boundary conditions

$$\begin{aligned} \rho_0 &= \rho_{\text{left}}, & \rho_n &= \rho_{\text{right}}, \\ u_0 &= u_{\text{left}}, & e_0 &= e_{\text{left}} \end{aligned} \tag{3.10}$$

and the initial guesses

$$\begin{aligned} &\rho_i = \rho_{\text{left}}, \quad e_i = e_{\text{left}} \quad \text{for} \quad x_i < 4.816, \\ &\rho_i = \rho_{\text{right}}, \quad e_i = e_{\text{right}} \quad \text{for} \quad x_i > 4.816, \\ &(\rho u)_i = \frac{(A\rho u)_{\text{left}}}{A(x_i)} \quad \forall x_i, \end{aligned} \tag{3.11}$$

where the subscript "left" ("right") refers to the numerical value of a quantity at the left (right) boundary of the local grid. When the l_1 procedure was applied locally on the cell (3.75, 5) divided into 10 equal subintervals with boundary conditions obtained from the solution for the equally spaced grid with $n=8$ on (0, 10), the shock in the final numerical solution was correctly positioned in the cell (4.75, 4.875). The solution values obtained by applying the l_1 procedure locally in (3.75, 5) coincided to 15 decimal digits with the results obtained globally on the grid (3.8) with the exception of the ρ_i and e_i at 4.75, which the "global" procedure had identified as being on the wrong branch of the solution.

The cell (4.75, 4.875) was divided into 10 finer subintervals each of length 0.0125 and the Euler equations were solved on this grid on (4.75, 4.875) with initial guesses (3.11). The l_1 procedure produced a solution with the shock correctly located in the interval (4.8125, 4.825). This interval was in turn divided into 10 subintervals each

of length 0.00125 and the Euler equations were solved on this grid on (4.8125, 4.825) with initial guesses (3.10). The l_1 procedure again produced a solution with the shock located in the correct cell, namely (4.815, 4.81625). The ρ_i for this last case are compared to the values of $\rho(x_i)$ in Table V. The shock is sharp and the relative error is less than 0.5%. The l_1 procedure has refined the position of the shock by a factor of 1000—the shock, originally known to be only in an interval of length 1.25, is now known to be in an interval of length 0.00125. While the l_1 procedure seems to have an ability to refine the position of the shock indefinitely, there is no point in carrying out further computations on a finer grid on (4.815, 4.81625) since the 0.5% error in the boundary conditions passed down from the coarsest grid will eventually make the calculations produce inaccurate results.

Finally, let us investigate the behavior of the l_1 procedure on grids with abrupt changes in mesh length that are not near the shock, simulating the patching of two grids. We will consider the following four grids:

$$\begin{gathered} 5 \text{ large cells each of width } 1.5 - 0.25 * 10^{-r} \text{ on } (0, 7.5 - 1.25 * 10^{-r}), \\ 1 \text{ small cell of width } 1.25 * 10^{-r} \text{ on } (7.5 - 1.25 * 10^{-r}, 7.5), \\ 2 \text{ large cells each of width } 1.25 \text{ on } (7.5, 10) \end{gathered} \tag{3.12}$$

for $r = 1, 2, 3, 4$. The widths of the three types of cells in (3.12) are in the ratios $1.2 * 10^r - 0.2 : 1 : 10^r$. When the l_1 procedure was started from the initial guesses (3.7), convergence to nonphysical solutions occurred. When the initial guesses for ρ_i were changed to the values of a linear function connecting the two boundary values $\rho(0) = 0.502$ and $\rho(10) = 0.776$, namely,

$$\rho_i = 0.502 + 0.0274 x_i, \tag{3.13}$$

TABLE V

Comparison of the Local l_1 Solution ρ_i of System (3.4), (3.10) with the Exact Solution $\rho(x_i)$ of System (3.1), (3.3)

x_i	ρ_i	$\rho(x_i)$	Error
4.8125	0.31739039	0.31702901	0.00036138
[illegible]	[illegible]	[illegible]	[illegible]
4.81625	0.70351214	0.70018542	0.00332672
4.8175	0.70363216	0.70030696	0.00332520
4.81875	0.70375203	0.70042833	0.00332370
4.82	0.70387174	0.70054954	0.00332220
4.82125	0.70399129	0.70067059	0.00332070
4.8225	0.70411069	0.70079148	0.00331921
4.82375	0.70422992	0.70091220	0.00331772
4.825	0.70434900	0.70103276	0.00331624

TABLE VI

Comparison of the l_1 Solution ρ_i of System (3.4), (3.3) on Grid (3.12) with $r=4$ with the Exact Solution $\rho(x_i)$ of System (3.1), (3.3)

x_i	ρ_i	$\rho(x_i)$	Error
0	0.50200000	0.50200000	0
1.499975	0.49904476	0.49904076	0.00000400
2.99995	0.47160924	0.47132962	0.00027962
4.499925	0.67383402	0.34904594	0.32478808
5.9999	0.76238443	0.76202453	0.00035990
7.499875	0.77470599	0.77470313	0.00000286
7.5	0.77470626	0.77470339	0.00000287
8.75	0.77584496	0.77584490	0.00000006
10	0.77600000	0.77600000	0

the initial values of $(\rho u)_i$ and e_i remaining as in (3.7), convergence to physical solutions did occur for $r=1, 2, 3, 4$. In all of these solutions, however, the shock in the solution produced by the l_1 procedure was off by one cell. The ρ_i for $r=4$ (cell-width ratios of 12000:1:10000) are given in Table VI. The errors (except for the error at 4.499925) are less than 0.05% in spite of the extreme changes in mesh length. The ρ_i, $(\rho u)_i$, and e_i at $x=7.5$ and 8.75 for grid (3.12) coincide to 15 decimal digits with the ρ_i, $(\rho u)_i$, and e_i of the 8-cell equally spaced grid.

The numerical experiments mentioned in the above paragraph confirm that the l_1 procedure in its current form is sensitive to the initial guesses (cf. [5]). All of the other numerical experiments mentioned in this section were rerun with the linear initial ρ_i of (3.13) replacing the discontinuous initial ρ_i of (3.7) (for global problems on (0, 10)) and linear initial ρ_i and e_i that connect the boundary conditions replacing the discontinuous initial ρ_i and e_i of (3.11) (for local problems). In many cases, the l_1 procedure with these initial guesses produced numerical solutions in which the shock was correctly positioned; in the majority of cases, the shock was incorrectly located by a few cells (three or fewer except in one case when it was five); finally, in some cases, convergence did not occur. In all cases of convergence, the numerical solution was close to the physical solution and the numerical accuracy of the solution values was the same as that seen in the solutions in Tables III–VI.

4. Discussion

Bidiagonal schemes related to (3.4) have been proposed by MacCormack [8] for compressible viscous flow and by Casier *et al.* [4] and Wornom [11] for Euler equations. In these papers, the steady-state solution is obtained by integration through natural time (time marching) rather than by solving the steady-state

system directly by Newton's method. Once steady state is reached and the time derivatives vanish, the conservative bidiagonal scheme of [11] and scheme (3.4) of the present paper are identical for non-shock cases. The good performance of bidiagonal schemes for purely subsonic, purely supersonic, unshocked transonic, and shocked transonic Euler flows has been documented in [4, 11]. These cases are not treated in the present paper, since they have three or fewer, rather than four, physical boundary conditions and therefore result in nonoverdetermined systems (3.4). While the l_1 procedure can be used to solve nonoverdetermined systems, it is more expensive than Newton's method or time marching and is not to be recommended.

The goal of the present paper is to investigate the numerics of the supersonic–subsonic case, in which the system is overdetermined by the presence of the four boundary conditions (3.3). In [4] this case was not treated. In [11], the supersonic–subsonic problem was discretized as an overdetermined algebraic system and the overdetermination was eliminated by arbitrarily omitting the continuity equation on the shocked cell. Conservation of mass was enforced by modifying the cells before and after the shock. In the present paper, no such arbitrary omissions or modifications are permitted. Here, it is the algorithm itself, not the user, that selects which equation to omit (always the momentum equation over the shocked cell, not the continuity equation). The principle of ending up with the same number of equations and unknowns is valid for physical systems that themselves are not overdetermined. In the inherently overdetermined supersonic-subsonic case, however, it is not only acceptable to end up with an overdetermined discrete system, it is preferred. The question to ask is then not why the system should be overdetermined but rather how the solution of the overdetermined system should be defined. The usual tactic is to define the solution to be the set of dependent variables that minimizes one of the l_p norms, $1 \leqslant p \leqslant \infty$, of the vector of (weighted) residuals. The choice $p = 1$ is based on the fact that $p = 1$ concentrates all of the error in one and only one residual, the the residual of the momentum equation on the shocked cell (all other residuals are zero). The example of Section 3 of [5] illustrates the dangers of taking the well-trodden path of using the least-squares principle ($p = 2$), which spreads the error over many cells.

The fact that the l_1 procedure omits one equation from the system (3.4), (3.3) indicates that it is not conservative. For the Euler system (3.4), the values of the unweighted absolute residuals of the momentum equation on the shocked cell are given in Table VII for the data of Table III. It is seen from the data in this table that the residual tends to zero as h decreases (n increases). This result is true for the data of Tables IV–VI also. The interpretation for this behavior is as follows. The l_1 procedure is designed to to minimize the residual of the momentum equation on the shocked cell, all other residuals being zero. The Rankine–Hugoniot jump condition on the shocked cell is that the residual of the momentum equation on the shocked cell vanish. Thus, the l_1 procedure seeks to minimize the departure from the Rankine–Hugoniot condition on the shocked cell. For certain cases such as the results for the Burgers' equation with an interior shock presented in [5], the

TABLE VII

Unweighted Absolute Residuals of the Momentum Equation on the Shocked Cell, $|r_{i2}|$

n	Shocked cell	Absolute residual $\|r_{i2}\|$
4	(2.5, 5.0)	1.11669816
8	(3.75, 5.0)	0.32297255
16	(4.375, 5.0)	0.23960115
32	(4.6875, 5.0)	0.09078541

residual on the shocked cell actually vanishes. For the Euler results mentioned above, the residual is not zero but is close to zero and decreases in magnitude with h. The nonconservative l_1 procedure produces the most nearly conservative solution that can be obtained on the given grid without artificially altering terms in the discrete equations.

The current algorithm for implementing the l_1 procedure, which is based on Newton's method and the Barrodale–Roberts algorithm [1, 2, 3] is much more expensive than standard algorithms (TVD, ENO, etc.), which require $O(n)$ operations. The Barrodale–Roberts algorithm, which requires storing and operating on a full matrix and ignores the quadridiagonal or block-bidiagonal structure of the systems representing Eqs. (2.7) and (3.4), is estimated to require not less than $O(n^2)$ and perhaps as many as $O(n^3)$ operations [3, pp. 231–236]. Moreover, this number of operations must be done on each step of the Newton outer loop. The present l_1 procedure is not only expensive, it may also be the cause of the changes in the position of the numerical shock for different initial conditions mentioned at the end of Section 3. Development of a more refined l_1 procedure is a key to both the computational and the theoretical success of the l_1 strategy. One candidate is the l_1 procedure of [7], which is based on the l_1 algorithm of Seneta and Steiger [9; 3, pp. 237–258] and unifies the theoretical and numerical processes in one framework. The development of new l_1 procedures for the banded matrices that occur in fluid-flow problems will involve a redirection of effort in operations research, which has historically been oriented towards the full-matrix situation. To carry out an analysis for the singularly perturbed Burgers' equation and the Euler equations similar to

matrices will have to be developed. The development of an analytical framework will allow one to answer fundamental questions such as whether the discrete solution is unique, what the domain of convergence is, and what the rate of convergence to the physically relevant solutions of the original problems (2.1) or (3.1) are.

5. Conclusion

The results presented here show that the l_1 procedure produces remarkably accurate numerical solutions of the 1-dimensional Burgers' equation and Euler equations on coarse grids, fine grids, and grids with abrupt changes in mesh length.

REFERENCES

1. I. BARRODALE AND F. D. K. ROBERTS, *SIAM J. Num. Anal.* **10**, 839 (1973).
2. I. BARRODALE AND F. D. K. ROBERTS, *Commun. ACM* **17**, 319 (1974).
3. P. BLOOMFIELD AND W. L. STEIGER, *Least Absolute Deviations* (Birkhäuser, Boston, 1983).
4. F. CASIER, H. DECONINCK, AND C. HIRSCH, *AIAA J.* **22**, 1556 (1984).
5. J. E. LAVERY, *J. Comput. Phys.* **79**, 436 (1988).
6. J. E. LAVERY, in *Proceedings, 11th Int. Conf. on Numerical Methods in Fluid Dynamics, Williamsburg, Virginia, June 26–July 1, 1988*, edited by D. L. Dwoyer *et al.* (Springer-Verlag, New York/Berlin, 1989), Lecture Notes in Physics, Vol. 323, p. 360.
7. J. E. LAVERY, *SIAM J. Num. Anal.* **26**, 1081 (1989).
8. R. W. MACCORMACK, *AIAA J.* **20**, 1275 (1982).
9. E. SENETA AND W. L. STEIGER, *Discrete Appl. Math.* **7**, 79 (1984).
10. G. R. SHUBIN, A. B. STEPHENS, AND H. M. GLAZ, *J. Comput. Phys.* **39**, 364 (1981).
11. S. F. WORNOM AND M. M. HAFEZ, *AIAA J.* **24**, 215 (1986).